Volume 49 Number 2 May 2025 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Fuzzy Clustering and Kernel PCA-Based High-Dimensional Imbalanced Data Integration with Octree Encoding

Qin Wang
Financial Teaching and Research Office, Zhongyuan University of Science and Technology, Zhengzhou 461100, China
E-mail: cgwac_wq@163.com

Keywords: fuzzy clustering, web crawler, high dimensional imbalance, national accounts, principal component analysis, data coding

Received: February 12, 2025

Due to the high-dimensional and unbalanced characteristics of national economic accounting data, there is a large amount of redundant information in the data, which will lead to problems such as boundary shift and integration overfitting shift when integrating the data, and will increase the difficulty of subsequent data integration. For this reason, a fuzzy clustering-based method for integrating high-dimensional unbalanced data of national accounts is proposed. Using the kernel principal component analysis method to reduce the dimensionality of high-dimensional imbalanced national economic accounting data, in order to reduce the complexity and sparsity of the data while preserving the main information of the original data as much as possible. Use fuzzy clustering algorithm for data clustering. Fuzzy clustering allows data points to belong to multiple clusters simultaneously, with each cluster having a membership measure that represents the strength of the relationship between data points and each cluster. Introducing deviation maximization for optimizing fuzzy clustering methods to ensure that the distance between each data point and its cluster center is as large as possible, while ensuring that the distance between data points within the same cluster is as small as possible. Based on text free grammar rules and conversion functions, convert national economic accounting data into hesitant fuzzy language data and obtain the optimal data attribute weight vector. Calculate the distance between different categories and the minimum distance, and determine the repulsion phenomenon between unknown and known classes through the objective function. Using Lagrange multipliers to solve the objective function and obtain the optimal clustering center. According to the optimal clustering center, complete the clustering of national economic accounting data and obtain different categories of national economic accounting data. According to the experimental results, the data integration imbalance of the proposed method ranges from 1.68% to 32.85%, and the total number of samples fluctuates between 139 and 5136. The three indicators of the integrated data are all greater than 0.88. Through actual coding cases, the coding ability of our method for high-dimensional imbalanced data in national economic accounting has been verified.

Povzetek: Predstavljena je vizualno-tekstualna klasifikacija sentimentov z uporabo računalniških metod. Prispevek izboljšuje analizo z integracijo večmodalnih podatkov in predlaga nov model.

1 Introduction

Integration of high-dimensional unbalanced data in national economic accounting refers to the process of systematically organizing, summarizing and integrating the high-dimensional and unbalanced data involved in the process of national economic accounting [1]. National economic accounting data mainly come from statistics, administrative data, accounting accounts, census and sample survey data and other aspects, covering agriculture, industry, services and other major industrial sectors, and covering the government, enterprises (including state-owned enterprises, private enterprises, foreign-funded enterprises, etc.), residents and other economic subjects, including gross domestic product (GDP), consumer price index (CPI), investment in fixed assets, total imports and exports and other economic indicators, so it has a significant high-dimensional characteristics; at the same time, there are differences in the proportion and growth rate of different industrial sectors in the national economy, the government and enterprises, as well as residents, and other economic agents in the distribution of income, consumption, investment and other aspects of the performance of the different regions of the level of economic development, industrial structure, resource endowments, etc., resulting in uneven economic performance between the regions. This leads to the imbalance of economic performance between regions, and there may be differences in economic performance in different time periods, which also leads to the significant imbalance of national economic accounting data [2-3]. The integrated high-dimensional unbalanced data provide richer materials for data mining and analysis, and researchers can use the integrated economic data to explore the deep economic laws and discover the potential economic problems and

trends, which helps to deepen the understanding of the national economy and provide strong support for economic development.

The research design and objectives of this paper focus on the integration of high-dimensional imbalanced data in national economic accounting. The research aims to address how to effectively manage and utilize these complex data to provide more accurate reflection of the national economic situation and policy formulation basis. The specific assumption is that by using web crawling technology to obtain comprehensive data and applying kernel principal component analysis for dimensionality reduction, combined with fuzzy clustering algorithms and data encoding methods, a more optimized data integration effect can be achieved than traditional methods. The measures of success include the degree of information retention after data dimensionality reduction, the accuracy of clustering results, and the convenience of data management. The evaluation indicators may involve the proportion of dimensionality reduction after data dimensionality reduction, the stability and interpretability of clustering results, as well as the storage efficiency and retrieval speed of encoded data. Through these measures, the research aims to provide new ideas and methods for the processing and analysis of national economic accounting data.

Fuzzy clustering is a clustering algorithm based on fuzzy set theory and fuzzy logic, which can take into account the situation that each data point belongs to multiple clustering centers, and can deal with uncertain or noisy data and improve the stability, reliability and flexibility of clustering analysis [4]. Applied to the integration of high-dimensional unbalanced data of national accounting, it can consider the ambiguity that each data point may belong to multiple categories, and realize the clustering division of national accounting data by calculating the affiliation degree of the data points to each category, so as to deal with high-dimensional and unbalanced data in a more flexible way. Therefore, we propose the integration of high-dimensional unbalanced data of national economic accounting based on fuzzy clustering to reflect the real structure of high-dimensional unbalanced data of national economic accounting more accurately, so as to improve the accuracy and flexibility of the high-dimensional unbalanced data of national economic accounting, solve the problem of unbalanced data of national economic accounting and improve the efficiency and automation degree of the integration of data, so that the overall operation status of the national economy can be reflected more accurately for the policy making. At the same time, through the integration and sharing of data resources, we can strengthen the international economic exchanges and cooperation, and jointly deal with the global economic problems and challenges [5].

The main contributions and limitations of different methods is shown in Table 1.

Table 1: Main contributions and limitations of different methods

Different methods	Main contributions	Limitation	Computing efficiency	Accuracy	Robustness
Ikoma et al [6]	Using web crawling technology to obtain Earth environmental data related to national life; Develop data integration layer and application layer	The diversity and format of data make API processing difficult; Insufficient data fusion	Medium	Medium	Low (data missing, incorrect, duplicate)
Yang et al [7]	Provide strategies for integrating big data analysis and neural network optimization design; Conduct predictive research and error analysis	Large error	High (big data processing and neural network training)	High (optimized through error analysis)	Medium (depending on data quality and neural network architecture)
Han et al [8]	Propose a data integration scheme based on k-anonymity and data privacy protection protocol; Introducing secure multi-party computation and ciphertext classification methods	There are significant differences in data format, structure, and quality; Encryption classification increases complexity	Low (ciphertext processing and secure multi-party computation)	Medium (depending on the effectiveness of privacy protection protocols)	High (Protecting Privacy)
Dong et al [9]	Propose an algorithm for clustering incomplete	The calculation process is quite complex	High (combining clustering and	High (time series prediction method)	Medium (depending on the performance of

	data; Combining clustering algorithms and information granulation methods; Propose a time series prediction method		information granulation to improve efficiency)		clustering algorithms)

2 Integration of high-dimensional imbalance data in national accounts

2.1 High-dimensional imbalance data acquisition for national accounts based on web crawler technology

Based on the characteristics of high-dimensional and unbalanced national economic accounting data, in order to effectively complete the integration of this data and realize the unified management of high-dimensional unbalanced data, web crawler technology is used to automate the data collection of national economic accounting data from different data sources, which mainly involves the selection of the Uniform Resource Locator System (URL) strategy and the extraction of page content [10]. First, the data collection specifies one or several starting URLs, downloads and interprets their corresponding web page source code. In order to ensure the accuracy of national accounts data extraction, regular expressions are utilized as the main extraction means to start data extraction, and after extraction, the extracted national accounts information is stored in the database for subsequent integration of high-dimensional imbalance data in national accounts. In addition, in the process of national economic accounting data collection, new URL addresses are continuously recognized and added to the pending list to ensure the comprehensiveness of national economic accounting data collection. The data collection based on web crawler will continue until it meets the preset termination criteria to stop crawling, and then the corpus will be constructed on this basis [11]. The specific process of crawling the high-dimensional imbalance data of national economic accounting is shown in Figure 1.

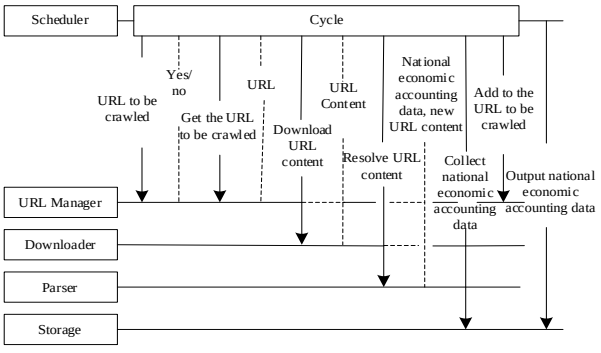


Figure 1: Crawling process of high-dimensional imbalanced data in national economic accounting

As shown in Figure 1, when crawling national accounting data, it mainly consists of five parts: scheduler, URL manager, downloader, parser and memory. Using the scheduler URL, query whether there are national accounting data URL resources to be crawled, and get the national accounting data URL resources to be crawled to locate the URL resources, and then transfer them into the URL manager; using the downloader, save the national accounting data resources URL resources; transfer the national accounting data resources URL resources to the parser, and use the parser to parse them, and get all the national accounting high-dimensional imbalance resources therein. data resource URL resource; transfer the national economic accounting data resource URL resource to the parser, and use the parser to parse it to obtain all the national economic accounting high-dimensional imbalance data and the new national economic accounting data resource URL locator; store the national economic accounting data in the memory and iteratively parse the new national economic accounting data resource URL locator to obtain it [12], after obtaining all the national economic accounting high-dimensional imbalance data, end the iteration, store all the national economic accounting high-dimensional imbalance data in the memory, so as to complete the acquisition of the national economic accounting high-dimensional imbalance data and obtain the national economic accounting data set X , this dataset contains imbalance data of different dimensions and attributes.

The reliability of using web crawlers to collect national economic data mainly depends on correct URL selection, accurate regular expression extraction, and continuous URL iteration parsing. By setting clear starting URLs and termination criteria, ensure the comprehensiveness and accuracy of data collection. To verify the accuracy of the data, data comparison and verification steps were implemented, comparing the crawled data with the officially released national economic accounting data. In addition, it also includes a data cleaning process, such as removing duplicate items, correcting erroneous data, etc., to ensure the quality of the final national economic accounting dataset obtained.

2.2 Kernel principal component analysis-based dimensionality reduction processing for high-dimensional unbalanced data

In high-dimensional unbalanced data for national economic accounting, "high-dimensional data" refers to data sets with multiple attributes or characteristics that

may be involved in many aspects of national economic accounting, such as time, region, industry, type of enterprise, economic indicators, etc. Unbalanced data refers to data sets where there are significant differences in sample sizes between categories. The unbalanced data refers to the data set in which there is a significant difference in the sample size between the categories. By integrating high-dimensional unbalanced data, redundancy and duplication in the data can be eliminated, and the accuracy and consistency of the data can be improved, which can help to reflect the real situation of the national economy more accurately, and provide a reliable basis for policy formulation and decision-making. In order to ensure the effect of subsequent data integration, the kernel principal component analysis is used to undersample the acquired high-dimensional imbalance data of national accounts, mapping the original data into a new low-dimensional space, while retaining the main information of the original data as much as possible. In this process, redundant information and noise will be effectively removed or weakened [13], which can realize the high-dimensional data dimensionality reduction, reduce the workload of the subsequent classifier training and testing, and synchronously improve the integration efficiency; and through the data dimensionality reduction can, to a certain extent, reduce the sparsity and complexity of the data distribution in the high-dimensional space, deal with the overfitting bias of the data. The preprocessed data can be maximized to achieve a relatively balanced state between the majority class and the minority class [14]. Kernel Principal Component Analysis (PCA) is a nonlinear dimensionality reduction technique that effectively addresses the problem of dimensionality reduction in high-dimensional imbalanced data by mapping raw data to a high-dimensional feature space and performing principal component analysis in that space. In the processing of high-dimensional imbalanced data in national economic accounting, kernel PCA first selects appropriate kernel functions and parameters, maps the original data to the kernel space, and forms a kernel matrix. By centralizing and decomposing the kernel matrix, the main components of the data, namely eigenvectors and eigenvalues, are extracted. Based on these main components, project the raw data into a low dimensional space to achieve dimensionality reduction of the data. This process not only preserves the main information of the data, but also helps balance the sample size between different categories, providing strong support for subsequent data integration and analysis. The steps to reduce and unbalance the high-dimensional unbalanced data of national accounts based on kernel principal component analysis are as follows.

(1) The initial sample matrix is established. Assuming that in the national economic accounting data set X, the value of the i th national economic accounting high-dimensional imbalance data under the j th attributes is x_{ij}, the initial matrix Z is established, and the matrix is normalized, and the processed matrix Z' is:

$$Z=\begin{pmatrix} x'_{11} & x'_{12} & \cdots & x'_{1n} \\ x'_{21} & x'_{22} & \cdots & x'_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x'_{i1} & \cdots & x'_{ij} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ x'_{m1} & x'_{m2} & \cdots & x'_{mn} \end{pmatrix} \tag{1}$$

Among them, x'_{ij} represents initialized, arbitrary national accounts high-dimensional imbalance data; the m represents the number of high-dimensional imbalances in national accounts to be evaluated. n represents the number of high-dimensional imbalance data attribute features of the national accounts.

(2) Selection of the nuclear function and nuclear parameters based on the high and unbalanced characteristics of the national economic accounting data, selection of the nuclear function that is appropriate to the characteristics of the sample data and the setting of the relatively optimal parameter values, mapping the R space data into the H space gives the kernel matrix U as:

$$U_{ij}=\varphi\left(Z'_i\right)\cdot\varphi\left(Z'_j\right) \tag{2}$$

Among them, φ represents a mapping operation. Z'_i and Z'_j represent the number corresponding to row i and column j of the high-dimensional unbalanced data S' of the matrix national economic accounts.

(3) Centered on the kernel matrix, centered on the high-dimensional disequilibrium data of the national accounts, which is formulated as follows:

$$U'_{ij}=U_{ij}-\bar{U}_i-\bar{U}_j+\bar{U} \tag{3}$$

Among them, U'_{ij} represents the kernel matrix after centering the high-dimensional disequilibrium data from the national accounts. $\bar{U}$ represents all data averages in the kernel matrix.

(4) U'_{ij} is decomposed to obtain the eigenvalue λ_u and its eigenvector V_u of the matrix, and the decomposition formula is:

$$U'_{ij}V_u=\lambda_u V_u \tag{4}$$

(5) Determine the dimension of the feature space t, conditions for spatial dimension t should be fulfilled are is:

$$\frac{\sum_{u=1}^{t}\lambda_u}{\sum_{u=1}^{n}\lambda_u}\geq 0.8 \tag{5}$$

(6) Calculate the principal components of X, the national accounts were obtained by projecting the high-

dimensional data as follows:

$$\left(V\cdot\varphi\left(X\right)\right)=\sum_{j=1}^{n}\alpha_j^u U\left(X_j,X\right) \tag{6}$$

Where, α_j^u represents the j th data of the characteristic vector V_u of high-dimensional data of national economic accounting, $\left(V\cdot\varphi\left(X\right)\right)$ represents the u th principal component of the high-dimensional data point X in national economic accounting.

(7) The factor score $Y_{i\tilde{t}}$ of the i th national economic accounting high-dimensional data point in the $\tilde{t}$ th spatial principal component is calculated as:

$$Y_{i\tilde{t}}=X\cdot V_{i\tilde{t}} \tag{7}$$

(8) Calculate the score F_i for each sample of national accounts data as:

$$F_i=w_1Y_{i1}+w_2Y_{i2}+\cdots+w_iY_{i\tilde{t}} \tag{8}$$

Among them, w represents the weight coefficients of each principal component.

(9) According to the results of F_i, the national economic accounting data X is arranged in descending order, and the first r copies of multiple data are deleted, so as to realize the national economic accounting data and realize the undersampling [15], complete the under-sampling of high-dimensional data of national accounts, realize the balanced processing of the data downgrading and unbalanced data, and obtain the downgraded data set $\hat{X}$ of national accounts.

2.3 Integration of national accounts data

2.3.1 Clustering of national accounts data based on fuzzy clustering

According to the above subsection to complete the national economic accounting data dimensionality reduction processing, although through the data dimensionality reduction can be reduced to a certain extent the imbalance of the data [16], but the national economic accounting data is still characterized by imbalance, therefore, in order to ensure that the effective integration of the data, fuzzy clustering algorithm is used for the data clustering, and the introduction of the data coding method on the basis of the clustering, which is combined with the above two steps to realize the effective integration of the national economic accounting data. Fuzzy clustering method has significant advantages in dealing with high-dimensional unbalanced data, the method can establish the uncertain description of the sample to the category, can be well adapted to the complex structure of high-dimensional unbalanced data [17], at the time when it clustering, more objectively reflecting the ambiguities and uncertainties in the real world, allowing the data points to belong to multiple clusters at the same time, and each sub-cluster has a degree of subordination measure, which expresses the strength of the relationship between the data points and each cluster. Each sub-cluster has an affiliation measure, which indicates the strength of the relationship between the data point and each cluster. This flexibility allows fuzzy clustering to better capture subtle differences and overlaps in the data, thus providing more accurate clustering results [18].

When the fuzzy clustering method is used to cluster the national economic accounting data $\hat{X}$, in order to ensure that the method can be better adapted to the unbalanced characteristics of the data, the optimization of the fuzzy clustering method is introduced by deviation maximization, which ensures that the distance between each data point and the cluster center of the cluster to which it belongs is as large as possible, and at the same time ensures that the distance between the data points in the same cluster is as small as possible. In this way, not only can the clustering results be clearer and more accurate, but also can reduce the impact of noise and outliers on the clustering results; and through the maximization of the deviation, the optimal number of clusters and clustering centers can be automatically selected, making the clustering results more stable and reliable. Assuming the set of attributes of the national economic accounting data $\hat{X}$ is $N=\{n_1,n_2,\ldots,n_n\}$, the attribute weight vector is $w=\left(w_1,w_2,\cdots,w_m\right)^T$, the national economic accounting data $\hat{X}_i$ under each attribute n_j is represented using language expressions or statements, the steps for clustering national accounts data based on fuzzy clustering are as follows:

(1) According to the text free grammar rules, use the conversion function to transform the national economic accounting data into hesitant fuzzy language data, and obtain the optimal national economic accounting data attribute weight vector $w'=\left(w_1',w_2',\cdots,w_m'\right)^T$ based on the maximization of the deviation.

(2) Take each of the national accounts data $\hat{X}_i$ as a category, calculating the spacing $d\left(\hat{X}_i,\hat{X}_j\right)$ between the different categories as:

$$d\left(\hat{X}_i,\hat{X}_j\right)=\left[\frac{1}{L}\sum_{l=1}^{L}\left(\frac{\left|\hat{X}_i,\hat{X}_j\right|}{2D+1}\right)^{\lambda}\right]^{\frac{1}{\lambda}},\left(\hat{X}_i,\hat{X}_j\in A\right) \tag{9}$$

Among them, D represents the maximum distance, λ represents the degree of affiliation calculated with the category spacing.

(3) Calculate the minimum distance $d'\left(\hat{X}_i,\hat{X}_j\right)$ for national accounts data as:

$$d'\left(\hat{X}_i,\hat{X}_j\right)=\arg\min_{X_i,X_j\in\hat{X}} d\left(\hat{X}_i,\hat{X}_j\right),\left(0\le i,j\le n,i\ne j\right) \quad (10)$$

Among them, i and j represent the i th and j th attribute for national accounts data.

(4) Assuming that the center of clustering of the national accounts data is known to be B_κ, the unknown class clustering centers to be B_χ, calculating the objective function $\Im$ between the two, to determine the phenomenon of exclusion, the objective function between the unknown and known categories of national accounts data is formulated as follows:

$$\Im=\sum_{i=1}^{a}\sum_{k=1}^{n} w_k' u_{ik}^m\left(S_k-\hat{X}_i\right)+\sum_{j=1}^{b}\sum_{k=1}^{n} w_k' v_{jk}^m\left(S_k-\hat{X}_i\right) \quad (11)$$

Among them, u_{ik}^m and u_{jk}^m indicate that under dimension m, the degree for the k th national accounts data of the i th and the j th attributes, the relationship between the two is as follows:

$$\sum_{i=1}^{a} u_{ik}^m+\sum_{j=1}^{b} v_{jk}^m=1 \quad (12)$$

Among them, a,b represents the set of national economic accounting data, the range for values for u_{ik}^m and v_{jk}^m are both $[0,1]$.

(5) Using Lagrange multipliers, solving Eq. (11) yields:

$$\Im'=\Im-\Diamond\left[\sum_{i=1}^{a} u_{ik}^m+\sum_{j=1}^{b} v_{jk}^m-1\right] \quad (13)$$

Among them, $\Diamond$ represents the Lagrange multiplication operator.

(6) Calculate to obtain an optimal national accounting data center K_i as:

$$\begin{cases}\dfrac{\partial\Im'}{\partial K_i}=-2\sum_{i=1}^{a} w_k' u_{ik}^m\left(S_k-K_i\right)\\ K_i=\sum_{k=1}^{n}\omega_k u_{ik}^m S_k\end{cases} \quad (14)$$

The clustering of national accounts data can be accomplished by obtaining different categories of national accounts data $Y=\left(Y_1,Y_2,...,Y_k\right)$ based on the optimal data centers calculated in equation (14).

2.3.2 Integration of national accounts data

In order to improve the effect of national economic accounting data integration, make a large number of complex high-dimensional national economic accounting data become easy to manage, based on the above clustering, the clustered national economic accounting data are coded in the paper, and data coding is an important part of data integration. In national economic accounting, the data involved are huge and complex, and it is not only inefficient but also easy to make mistakes when dealing with these raw data directly. Through the coding process, each data point can be given a concise, unique identification, thus facilitating the rapid identification, retrieval and management of data, rapid identification of the clusters to which each data point belongs, so that the data has a clear identification, easy to understand and interpret [19]; at the same time, the coded data is more concise than the original data, which can save storage space, and realize the integration of high-dimensional unbalanced data.

In the paper, an octree model is used to code and store $Y=\left(Y_1,Y_2,...,Y_k\right)$ to complete the overall integration of high-dimensional unbalanced data of national economic accounting, we encode and store the data; fork tree is a special kind of tree data structure with order and hierarchy, which makes it has great advantages in the balance of display accuracy and speed, the elimination of hidden lines and hidden surfaces, etc., and it can efficiently deal with the sparse and dense data to complete the optimization of storage and integration of the data. Therefore, the octree model is constructed in the paper, and the clustering results are converted into binary codes and stored in the octree nodes to accomplish efficient data compression coding and storage [20]. The structure of the octree model is shown in Figure 2.

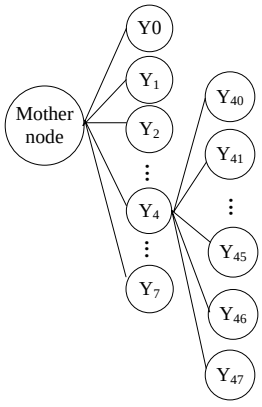


Figure 2: Octree model structure

The specific steps for octree-based coding of national accounts data are as follows.

(1) Using the national accounts data Y obtained in 2.3.1, the octree retrieval model is constructed to extend the quadtree in the two-dimensional space to the three-dimensional subdivision space.

$$\begin{cases}\tau=\tau\left(Y_k\right)+\hbar/4\\ \zeta=\zeta\left(Y_k\right)+\hbar/4\\ \xi=\xi\left(Y_k\right)+\hbar/4\end{cases} \quad (15)$$

Among them, τ, ζ and ξ represent the 3D coordinates of the node, $\hbar$ represents the cube length of the node, such that the acquired high-dimensional national accounting data Y_k as the octree parent node.

(2) Taking the parent node Y_k as the datum node, the positions of edge neighbor nodes and face neighbor nodes of the datum node are computed, respectively, using Y_5 and Y_6 to express as:

$$\begin{cases} \tau(Y_6) = \tau(Y_5) + 4 \\ \zeta(Y_6) = \zeta(Y_5) - 4 \\ \xi(Y_6) = \xi(Y_6) \end{cases} \tag{16}$$

(3) The expression for the octree retrieval model child nodes Y_4 are computed from the coordinates of the parent node:

$$\begin{cases} \tau(Y_4) \in \{\tau(Y), \tau(Y) + \hbar, \tau(Y) - \hbar\} \\ \zeta(Y_4) \in \{\zeta(Y), \zeta(Y) + \hbar, \zeta(Y) - \hbar\} \\ \xi(Y_4) \in \{\xi(Y), \xi(Y) + \hbar, \xi(Y) - \hbar\} \end{cases} \tag{17}$$

(4) Introducing judgmental encoding, judging the child nodes Y_{45} and child node Y_{46} whether the corresponding national economic accounting data is located in the code number is consistent. If it is consistent, the two child nodes are judged to be neighboring nodes to each other; if it is not consistent, the corresponding data Y_{46} is put into the neighborhood data sheet Y_{45}, calculated as follows:

$$\begin{cases} \tau(\Delta) = \tau(\nabla) \pm \hbar_{\nabla} \\ \zeta(\Delta) = \zeta(\nabla) \\ \xi(\Delta) = \xi(\nabla) \end{cases} \tag{18}$$

Among them, ∇ and Δ represent different judgment code.

(5) When the judgment code is the code of two nodes and only one bit is different, these two nodes are edge neighbor nodes to each other, this process is formulated as:

$$\begin{cases} \tau(\nabla) \pm \hbar_{\nabla} = \tau(\Delta) \\ \zeta(\nabla) \pm \hbar_{\nabla} = y(\Delta) \\ \xi(\Delta) = \xi(\nabla) \neq 0 \end{cases} \tag{19}$$

(6) When the judgment code is the code of two nodes, and only one bit is different but connected, these two nodes are point neighbor nodes to each other, this process is formulated as:

$$\begin{cases} \tau(\nabla) \pm \hbar_{\nabla} = \tau(\Delta) \\ \zeta(\nabla) \pm \hbar_{\nabla} = y(\Delta) \\ \xi(\nabla) \pm \hbar = \xi(\Delta) \end{cases} \tag{20}$$

To summarize the above steps, determine the relationship between all the child nodes, according to the node relationship can be expressed in the hierarchy and spatial relationship of the data block, according to the occupancy information and hierarchical relationship of the node, each node is assigned a unique address code, and finally, all the nodes of the address code and attribute bits are stored in order to form the linear octree coded data set, complete the national economic accounting data coding, so as to achieve the final integration of national economic accounting high-dimensional imbalance data. The final integration of the high-dimensional unbalanced data of national accounts is realized.

3 Test analysis

To verify this method can integrate high dimensional unbalanced data of national economic accounting, from labor dynamic survey in 2022, family financial survey, national health and nutrition survey, rural urban migration survey and family income five data sets, randomly selected the national economy unbalanced data, divided into regional unbalanced data, industry imbalance data, urban and rural imbalance data, income imbalance data and other unbalanced data (education, social welfare, etc.) five types of unbalanced data set, as an experimental data set. At the same time, the experiment utilizes the generalized web crawler technique to crawl the national economic accounting data, and the multi-threaded crawling object is expanded from the seed URL to the whole Web, so as to provide reliable data for the experiment. The specific coverage of the national economic accounting data set is shown in Table 2.

3.1 Analysis of data crawling results

In order to verify whether the method of this paper can effectively crawl the high-dimensional imbalance data of national economic accounting, use the method of this paper to crawl the high-dimensional imbalance data of national economic accounting from the experimental data set, in addition, part of the data is more than the categorized data set, this experiment uses a widely used transformation method to merge some of the categories of the multiclassified data set into a dichotomous data set, and use the method of this paper to crawl the data of national economic accounting to be shown in Table 3.

Table 2: Experimental dataset

Name of National Economic Accounting Dataset	Coverage
Labor force dynamic survey	Employment, unemployment, labor loss, etc
Family Financial Survey	Different provinces, cities, and rural areas
National Health and Nutrition Survey	Community, Family, Individual, Health
Survey on Rural Urban Migration	Rural residents, urban households, and migrant workers
Household Income Survey	Individual income distribution in urban and rural areas, as well as in rural areas

Table 3: High dimensional Imbalance data in national economic accounting

Name of National Economic Accounting Dataset	Total number of samples (pieces)	Attribute	Number of majority class samples (pcs)	Number of minority class samples (pcs)	Imbalance (°)
Labor force dynamic survey	2361	35	1298	210	8.5
Family sinancial survey	598	42	232	21	2.6
National Health and nutrition survey	1500	29	895	98	4.0
Survey on rural urban migration	139	28	59	9	1.5
Household income survey	5136	16	2150	413	32.5

Analysis of Table 2 shows that: using this paper's method to crawl to the national economic accounting high-dimensional imbalance data, the imbalance degree ranges from 1.68% to 32.85%, and the total number of samples fluctuates between 139 and 5136, the sample size and imbalance degree of the national economic accounting data is widely distributed, and the difference is as high as 4997bit; In addition, the high-dimensional imbalance data of national economic accounting crawled by the method of this paper contains five types of information in the experimental dataset, which indicates that the imbalance data of national economic accounting can be obtained by using the crawler method of this paper.

3.2 Analysis of the effect of data downscaling

In order to verify the effect of the method of this paper on the dimensionality reduction of high dimensional imbalance data of the national economy, the indicator pressure function o and descending masses ϕ are introduced, pressure function o denotes the loss value of the data before and after dimensionality reduction, the range of values is $[0,1]$, the smaller its value, the smaller the degradation loss, and vice versa; the degradation quality ϕ denotes the quality of the data after dimensionality reduction, and the range of values is $[0,1]$, the larger the value, the better the effect of data dimensionality reduction, and vice versa; both are calculated as follows:

$$o=\sqrt{\frac{\sum_{i=1,j=1,i\neq j}^{\vartheta}\left(d_{ij}-\bar{d}_{ij}\right)^2}{\sum_{i=1,j=1,i\neq j}^{\vartheta}d_{ij}^2}} \tag{21}$$

$$\phi=\frac{1}{\varpi\vartheta}\sum_{i,j\in 1,\cdots,\varpi}\Theta_{i,j}-\frac{\varpi}{\vartheta-1} \tag{22}$$

Where, d_{ij} represents the spatial distance between the i th data and the j th data after dimension reduction of high dimensional data, $\bar{d}_{ij}$ represents the mean spatial distance of d_{ij}, $\Theta_{i,j}$ represents the number of overlaps, ϑ represents the number of samples taken, ϖ represents the neighborhood size.

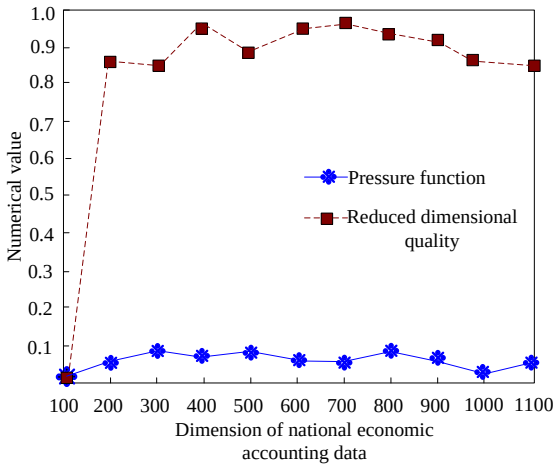


Figure 3: Calculation of dimensionality reduction results indicators for high dimensional imbalance data in national economic accounting

From the experimental data set, 10 national economic accounting high-dimensional imbalance data are randomly selected and numbered from 1 to 10, using the method of this paper for its dimensionality reduction processing, using the above formula to calculate the pressure function and quality of dimensionality reduction of the national economic accounting data after dimensionality reduction processing, and the results of the two calculations are shown in Figure 3.

The analysis of Figure 3 shows that: after using this paper's method to downsize the experimental national economic accounting high-dimensional imbalance data, the maximum value of the pressure function is 0.017, indicating that the loss value of the national economic accounting data after using this paper's method to downsize is smaller; and the minimum value of the quality of the downsizing is 0.86, indicating that the quality of the national accounting data after using this paper's method to downsize is retained to a higher degree, further proving that this paper's method is good for the downsizing of high-dimensional imbalance data of national economic accounting.

3.3 Analysis of the validity of the integration of high-dimensional imbalance data in national accounts

In order to verify whether the fuzzy clustering method in this paper can integrate the high-dimensional imbalance data of the experimental national accounts, and to introduce the adjust mutual information Q, adjust Rand factor ℓ indicators and guidelines for variance ratios σ, adjust mutual information Q is used to evaluate the correlation between the category to which the data belong and the experimental national accounts high-dimensional imbalance data; and the adjust Rand coefficients ℓ is used to evaluate the fit of distributions between different experimental national accounts high-dimensional imbalance data under the same category; the cubic difference ratio criterion σ represents the clustering effect, and all three take values in the range of $[0,1]$, the larger the value, the better the clustering effect of the high-dimensional imbalance data of the experimental national accounts, and the three formulas are:

$$\ell = \frac{\upsilon - E[\upsilon]}{\max(\upsilon) - E[\upsilon]} \tag{23}$$

$$Q(\iota_1, \iota_2) = \frac{M - E[M]}{\frac{1}{2}\left(H(\iota_1) + H(\iota_2) - E[M]\right)} \tag{24}$$

$$\sigma = \frac{tr(\Phi)}{tr(\Gamma)} \cdot \frac{\Xi - \mathrm{I}}{\mathrm{I} - 1} \tag{25}$$

Among them, M represents the mutual information between high-dimensional imbalance data in national accounts ι_1 and ι_2. E represents the expected value of both, υ represents the Rand coefficient, H represents the information entropy, φ denotes the number of pairs of elements belonging to the same category in the clustered and integrated high-dimensional imbalance data of the national accounts and the real category. ι represents the logarithm of the elements of the different classes of national accounts high-dimensional imbalance data after clustering, C represents the clustering completeness, Φ represents the covariance matrix between classes of high-dimensional unbalanced data from national accounts, Γ represents the covariance matrix within the class, Ξ represents the total number of high-dimensional imbalances in the national accounts. I represents the number of data categories.

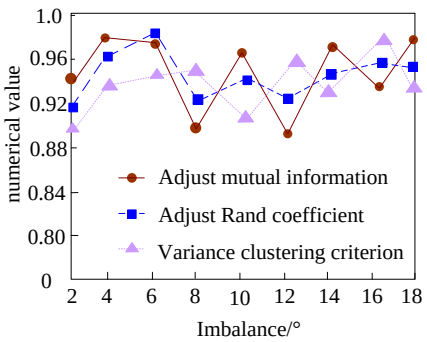


Figure 4: Calculation table of data clustering indicators

From the experimental data set, one data set was randomly selected, with the increase of the high-dimensional imbalance data of national economic accounting, calculate the adjust mutual information Q and the adjust Rand coefficient ℓ and the variance ratio criterion σ for the high-dimensional imbalance data of different national economic accounts, in order to analyze the data processing effect of the method in this paper, the test results are shown in Figure 4.

The analysis of Figure 4 shows that: after using the fuzzy clustering method of this paper to integrate the high-dimensional imbalance data of the experimental national economic accounting, with the continuous change of data imbalance, the adjusted mutual information, adjust Rand coefficient and variance ratio criterion of the integrated data are all greater than 0.88, which are in the larger value, indicating that the use of this paper's fuzzy clustering method can be used to integrate the high-dimensional imbalance data of the experimental national economic accounting effectively. integration.

3.4 Analysis of the effects of integrating high-dimensional imbalance data in national accounts

In order to verify whether the method of this paper can code the high-dimensional imbalance data of the experimental national accounts, the high-dimensional imbalance data of a city's national accounts are randomly selected from the experimental data set and integrated and coded using the method of this paper, and the coding results of the excerpted parts are shown in Table 4.

Table 4: Coding of high dimensional unbalanced data in national economic accounting (section table)

Data encoding	First level coding	Content	Secondary encoding	Content
2110205	2	National economy industry	5	Producer index
2110308	3	National economy retail enterprises	8	Operational index
2110401	4	Fixed assets in national economic accounting	1	Investment distribution data
2110502	5	Current assets in national economic accounting	2	Overall distribution data
2110602	6	Intangible assets in national economic accounting	2	Overall distribution data
2110701	7	Provincial National Economic Operation Status	1	Municipal level economic operation situation
21118Y1	18	2008 National Economy	Y1	Economic performance in January

The data encoding in Table 4 shows the implementation scheme of our method for encoding high-dimensional imbalanced data in national economic accounting. Specifically, the encoding scheme is divided into two levels: first level encoding and second level encoding. First level coding usually represents the major category or main classification of data, such as "2" representing "national economic industry", "3" representing "national economic retail enterprise", etc. These numbers are short and representative, making it easy to quickly identify and classify data.

The second level encoding further refines the specific content or attributes of the data, such as "5" representing "producer index" under the first level encoding "2", "8" representing "operation index" under the first level encoding "3", etc. This type of secondary encoding not only increases the accuracy of data description, but also helps to more accurately locate the required information during data analysis.

Analysis of Table 3 shows that: after using the method of this paper to code the high-dimensional imbalance data of national economic accounting, each expression of the data can be expressed in numbers or letters, for example, in Table 3, "2110205" stands for "national economic accounting industrial producer index", "21118Y1" stands for "January 2018 economic operation of the national economy", etc., where "211" stands for "January 2018 economic operation of the

national economy", and the last four digits represent a class code and a secondary code, respectively, so as to comprehensively describe the national economic accounting data, which further proves that the method of this paper can effectively encode the high-dimensional imbalance data of the national economic accounting, so as to improve the effect of the integration of the national economic accounting data.

In order to verify the superiority of the fuzzy clustering method proposed in this article in integrating high-dimensional imbalanced data of national economic accounting, fuzzy clustering was compared with K-means clustering method and hierarchical clustering method. The experimental dataset will still use the dataset described earlier, and one of the datasets will be randomly selected for the experiment. The following are the comparison results of three clustering methods.

Table 5: Comparison of clustering methods

Method	First level coding	Content	Secondary encoding	Content
Fuzzy Clustering	0.92	0.90	0.93	120
K-means Clustering	0.85	0.82	0.87	80
Hierarchical Clustering	0.88	0.86	0.89	180

Table 6: Comparison of dimensionality reduction methods

Method	Stress Function	Dimensionality Reduction Quality	Runtime (seconds)
Kernel PCA	0.017	0.86	150
PCA	0.035	0.78	100
t-SNE	0.022	0.80	240

According to Table 5, fuzzy clustering outperforms K-means clustering and hierarchical clustering in adjusting mutual information, adjusting Rand coefficient, and variance ratio criteria, indicating that the fuzzy clustering method has higher accuracy and effectiveness in integrating high-dimensional imbalanced data in national economic accounting.

To verify the superiority of the proposed kernel PCA method in dimensionality reduction of high-dimensional imbalanced data in national economic accounting, kernel PCA was compared with PCA and t-SNE dimensionality reduction methods. The experimental dataset will still use the dataset described earlier, and one of the datasets will be randomly selected for the experiment. Here are the comparison results of three dimensionality reduction methods.

Analysis of Table 6 shows that kernel PCA outperforms PCA and t-SNE in both pressure function and dimensionality reduction quality indicators, indicating that the kernel PCA method has higher accuracy and effectiveness in dimensionality reduction of high-dimensional imbalanced data in national economic accounting.

4 Conclusion

In order to accurately reflect the overall operation of the national economy and provide a scientific and reasonable basis for policy making, the integration of high-dimensional unbalanced data of national economic accounting based on fuzzy clustering is proposed. Firstly, we use the web crawler technology to obtain the high-dimensional unbalanced data of national economic accounting to make the basis for data integration; based on the principal component analysis, we balance the high-dimensional unbalanced data of national economy and complete the data under-sampling treatment, so that the pre-processed data can maximize the possibility of achieving the state of relative balance between the majority class and the minority class; based on the principal component analysis method, we downsize the high-dimensional balanced data of national economy, this completes the pre-processing of the high-dimensional imbalance data of national economy; using fuzzy clustering, the obtained national economic accounting data are combined into a cluster to realize the integration of the high-dimensional imbalance data of national economic accounting, so as to provide strong support for economic development.

This article uses clustering performance indicators such as Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) to evaluate the integration effect. Compared with traditional clustering methods such as k-means, hierarchical clustering, or DBSCAN, fuzzy clustering exhibits higher performance in handling high-dimensional imbalanced data. This is mainly because fuzzy clustering can handle the ambiguity of data points belonging to multiple categories, thereby more accurately reflecting the inherent structure of the data. Specifically, fuzzy clustering is more effective in handling categories with overlapping or fuzzy boundaries, while traditional hard clustering methods such as k-means may perform poorly in these situations. Although hierarchical clustering can generate hierarchical clustering structures, it may face challenges in computational complexity and result interpretability when processing high-dimensional data. DBSCAN relies on density thresholds to identify clusters, which is not flexible enough for handling imbalanced data.

The experimental results show that when using the method proposed in this paper to reduce the dimensionality of these data, the maximum value of the pressure function is only 0.017, and the minimum value of the dimensionality reduction quality reaches 0.86, indicating that this method effectively reduces the dimensionality of the data while preserving its quality. The fuzzy clustering method was used to perform cluster analysis on the integrated high-dimensional imbalanced data. The results showed that the three indicators of adjusted mutual information, adjusted Rand coefficient, and variance ratio criterion all exceeded 0.90, and showed superiority compared to other clustering methods such as K-means clustering and hierarchical clustering. Meanwhile, the comparison results between the kernel PCA dimensionality reduction method and PCA and t-SNE dimensionality reduction methods show that kernel PCA outperforms the other two methods in terms of pressure function and dimensionality reduction quality, verifying the high accuracy and effectiveness of kernel PCA in dimensionality reduction of high-dimensional imbalanced data in national economic accounting.

In summary, the method proposed in this article for integrating high-dimensional imbalanced data in national economic accounting based on fuzzy clustering demonstrates higher performance in handling high-dimensional imbalanced data. By comparing with SOTA technology, we can find that fuzzy clustering is more effective in processing data with fuzzy boundaries and overlapping categories, while kernel PCA can more effectively extract nonlinear features from the data. These advantages make the method proposed in this article more accurate and reliable in handling complex national economic accounting data, providing a more scientific and reasonable basis for policy-making.

By using PCA dimensionality reduction to process high-dimensional imbalanced data in national economic accounting, key information can be effectively preserved, complexity and sparsity can be reduced, which helps to improve the accuracy and efficiency of policy decision-making and economic modeling. Fuzzy clustering methods can accurately reflect the economic characteristics of high-dimensional data, such as differences in labor market and economic conditions, and provide targeted recommendations for policy formulation. The accuracy of web crawling technology is crucial for the integrity of economic datasets. The method proposed in this article can accurately capture comprehensive economic data, providing reliable support for policy-making and economic research. Overall, the method proposed in this article has achieved significant results in data integration, improving data processing efficiency and accuracy, and providing strong data support for national economic decision-making.

Although kernel PCA has its advantages in processing high-dimensional data, it still faces challenges when dealing with high-dimensional spatial complexity data in the economic field. Economic datasets contain a large number of variables (such as labor dynamics, household finances, health and nutrition status, etc.) that may be interrelated and highly nonlinear. Kernel PCA transforms these variables into a new feature space through nonlinear mapping, which may reveal complex structures hidden in the original data. However, due to the limitations of time and space complexity, kernel PCA may not be directly applicable to very large economic datasets.

To overcome these limitations, researchers may adopt strategies such as using approximation algorithms to accelerate the computation and feature decomposition of kernel matrices, or using distributed computing frameworks to process large-scale datasets. In addition, feature selection or pre dimensionality reduction can be used to reduce the dimensionality of input data, thereby reducing the computational burden of kernel PCA.

References

[1] Rashid, A., Nakib, T. H., & Shahriar T. abib M.A. Hasanuzzaman M. (2024). Energy and economic analysis of an ocean thermal energy conversion plant for Bangladesh: A case study. Ocean engineering, 293(Feb.1):1.1-1.17. https://doi.org/10.1016/j.oceaneng.2023.116625.

[2] Dev, K., Chih-Lin I, & Khowaja, S. A. (2023). Guest editorial dense - data integrity, integration and security issues for consumer data in industry 5.0. IEEE Transactions on Consumer Electronics, 69(4):809-812. https://doi.org/10.1109/TVT.2024.3399470.

[3] Gallo-Bernal, S., Pea-Trujillo, V., Briggs, D., Machado-Rivas, F., Pianykh, O. S., & Flores, E. J., et al. (2024). A data science-based analysis of socioeconomic determinants impacting pediatric diagnostic radiology utilization during the COVID-19 pandemic. Pediatric radiology, 54(11):1831-

1841. https://doi.org/10.1007/s00247-024-06039-8.

[4] Liu R. Yang F., Wang. (2022). Incremental clustering algorithm for high dimensional data based on improved spark technology. Computer Simulation, 39(12), 383-386, 444. https://doi.org/10.3969/j.issn.1006-9348.2022.12.070.

[5] Cheng, Y., & Su, J. (2024). Economic data forecasting through interval data analysis. International Journal on Artificial Intelligence Tools, 33(07), 2440002. https://doi.org/10.1142/S0218213024400025.

[6] Ikoma, E., & Kitsuregawa, M. (2023). DIAS-earth environment data integration and analysis system. Communications of the ACM, 66(7):85-86. https://doi.org/10.1145/3589233.

[7] Yang, G., Li, X., Yu, T., Wu, S., & Liu, Y. (2022). A new model of environmental-economic coordination prediction using credible neural network integration and big data analysis. Security and Communication Networks, 2022(1), 3454821. https://doi.org/10.1155/2022/3454821.

[8] Han, S., Ma, H., Taherkordi, A., Lan, D., & Chen, Y. (2024). Privacy-preserving data integration scheme in industrial robot system based on fog computing and edge computing. IET communications, 18(7):461-476. https://doi.org/10.1049/cmu2.12749.

[9] Dong, S., & Tsai, S. B. (2021). Economic management data envelopes based on the clustering of incomplete data. Mathematical Problems in Engineering, 2021(1), 4312842. https://doi.org/10.1155/2021/431

[10] Silva, L., & Barbosa, L. (2024). Improving dense retrieval models with LLM augmented data for dataset search. Knowledge-based systems, 294(Jun.21):1.1-1.9. https://doi.org/10.1016/j.knosys.2024.111740.

[11] Stasenko, M., & Quinn, G. P. (2023). Stasenko, Marina, Quinn, Gwendolyn P. Improvements in sexual orientation and gender identity data collection through policy and education. American Journal of Public Health, 113(8):834-835. https://doi.org/10.2105/AJPH.2023.307344.

[12] Angaman, K. V., Mirzabaev, A., & Niang, B. B. (2024). Economic impacts of land degradation: Evidence from Côte d'Ivoire. Land Degradation and Development, 35(4):1541-1552. https://doi.org/10.1002/ldr.5004.

[13] Chatzimparmpas, A., Paulovich, F. V., & Kerren, A. (2023). HardVis: Visual analytics to handle instance hardness using undersampling and oversampling techniques. Computer Graphics Forum: Journal of the European Association for Computer Graphics, 42(1):135-154. https://doi.org/10.1111/cgf.14726.

[14] Lin, C., Tsai, C. F., & Lin, W. C. (2023). Towards hybrid over- and under-sampling combination methods for class imbalanced datasets: an experimental study. Artificial Intelligence Review: An International Science and Engineering Journal, 56(2):845-863. https://doi.org/10.1007/s10462-022-10186-5.

[15] Ephzibah, E. P., & Remya, L. (2022). Dimensionality reduction using principal component analysis and multi-label fuzzy classification for rice crop disease in rural areas of India. ECS transactions, 107(1):16451-16458. https://doi.org/10.1149/10701.16451ecst.

[16] Liu, Z., & Letchmunan, S. (2024). Enhanced fuzzy clustering for incomplete instance with evidence combination. ACM transactions on knowledge discovery from data,18(3):72.1-72.20. https://doi.org/10.1145/3638061.

[17] Madan, S., Komalavalli, C., Bhatia, M. K., Laroiya, C., & Arora, M. (2024). An optimized SVM? RFE based feature selection and weighted entropy Kmeans approach for big data clustering in MapReduce. Multimedia Tools and Applications, 83(30):74233-74254. https://doi.org/10.1007/s11042-023-18044-4

[18] Quintana-Orti, G., Hernando, F., & Igual, F. D. (2023). Algorithm 1033: Parallel Implementations for computing the minimum distance of a random linear code on distributed-memory architectures. ACM transactions on mathematical software,49(1):8.1-8.24. https://doi.org/10.1145/3573383

[19] Wang, L., Witherden, F., & Jameson, A. (2024). An efficient GPU-based h -adaptation framework via linear trees for the flux reconstruction method. Journal of Computational Physics, 502(3 Pt.1): ARTN 036108-036128. https://doi.org/10.1016/j.jcp.2024.112823.

[20] Richard D., L., Sabyasachi, S., & Ankani, Chattoraj, Ralf. M. Haefner. (2023). Bayesian encoding and decoding as distinct perspectives on neural coding. Nature neuroscience,26(12):2063-2072. https://doi.org/10.1038/s41593-023-01458-6.

Enhancing Network QoS via Attack Classification Using Convolutional Recurrent Neural Networks

Jawad Alkenani*, Mohsen Nickray
Department of Computer Engineering and Information Technology, University of Qom, Qom, Iran
E-mail: Jawadalkenani@sa-uc.edu.iq [1], m.nickray@qom.ac.ir [2]
*Coresponding author

Keywords convolutional neural networks, recurrent neural networks, attack class, anomaly detection

Received: November 21, 2024

Cyber-attacks and intrusions in networks refer to malicious activities that breach or damage data. These activities include direct attacks, such as denial-of-service (DoS) attacks, which overwhelm servers with requests to disrupt services. Intrusion involves unauthorized access to systems by exploiting security vulnerabilities. Malware threats like viruses and worms infect systems to steal information. Additionally, social engineering techniques deceive individuals into revealing sensitive information, while phishing relies on fake messages or websites to gather user data. To prevent these attacks, it is necessary to implement effective security strategies, such as knowing the attack class to protect the network and data. In this paper, ConvRNN (Convolutional Recurrent Neural Network) is used as a large-scale advanced model between Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to process data containing spatial and temporal information. In addition, ConvRNN generates magical features from data through convolutional layers and serial convolution by RNN, which creates the model's ability to understand complexity, especially in security and surveillance agreements. The simulation results show that the proposed model outperforms LSTM, including precision, recall, F1 score, ROC curve, TPR, FPR, FNR, and accuracy.

Povzetek: Prispevek predlaga izboljšano klasifikacijo omrežnih napadov z uporabo konvolucijskih rekurentnih nevronskih mrež, s poudarkom na izboljšanju kakovosti storitev (QoS) in odkrivanju varnostnih groženj.

1 Introduction

Cyber-attacks and intrusions pose significant threats to the integrity, confidentiality, and availability of data in networks. Various types of attacks exist, including denial-of-service (DoS) attacks, where malicious actors overwhelm a server with excessive requests, disrupting services for legitimate users. Intrusions involve unauthorized system access by exploiting security vulnerabilities, allowing attackers to steal, alter, or delete data. Malware is another major threat, encompassing different types of malicious software such as viruses, worms, ransomware, and Trojan horses. Viruses attach to legitimate files and spread when shared, while worms replicate themselves across networks independently. Ransomware encrypts files and demands payment for decryption, and Trojan horses disguise themselves as legitimate software to carry out hidden malicious activities [1].

Social engineering techniques manipulate individuals into revealing confidential information, often through impersonation or pretexting. Phishing involves fraudulent emails or websites that deceive users into providing sensitive information, with spear-phishing targeting specific individuals or organizations. Man-in-the-middle (MitM) attacks occur when an attacker intercepts communication between two parties, allowing data theft or manipulation, particularly in unsecured Wi-Fi networks [2]. To prevent these attacks, organizations can implement firewalls that filter incoming and outgoing traffic based on security rules, blocking potentially harmful traffic. Data encryption ensures that intercepted information remains unreadable without proper decryption keys. Regular software updates are crucial for patching vulnerabilities that attackers may exploit, while continuous network monitoring helps identify unusual activities that could indicate an attack. Educating employees about cybersecurity best practices and the importance of strong passwords can significantly reduce the risk of successful attacks. Multi-factor authentication (MFA) adds an extra layer of security by requiring additional verification steps beyond just a password. Finally, having a well-defined incident response plan ensures organizations can respond quickly and effectively to security breaches, minimizing potential damage. Understanding the various cyber-attack types and implementing comprehensive security measures are essential for protecting networks and sensitive information [3].

This group related to ConvRNN (Convolutional Recurrent Neural Network) includes a variety of applications that take advantage of its capabilities in processing data with temporal and spatial dimensions. In the field of real-time video analysis, ConvRNN can be used to detect abnormal control or suspicious activation,

which contributes to the activation of security surveillance. ConvRNN is also relied upon in network detection guidance in networks, where it helps to analyze network traffic data and abnormal people that may indicate the most important features of the network. In addition, ConvRNN applications have become computer vision, such as object recognition and motion tracking, which enables later understanding of behavior in the view. There are also studies on the development of intelligent systems using ConvRNN, where the accuracy of detection and analysis in real-time is enhanced. Finally, ConvRNN is used in the analysis of complex temporal data, such as weather forecasting or financial feature analysis, which shows the potential to handle complexity in various fields [4].

ConvRNN combines convolutional layers with recurrent layers, making it effective for processing data with both spatial and temporal dimensions. This architecture can significantly enhance Quality of Service (QoS) across various applications. In video surveillance, ConvRNNs analyze video streams by capturing spatial features and temporal dependencies, improving real-time object detection and tracking. For traffic prediction, they forecast traffic patterns by analyzing spatial data from road networks alongside historical traffic conditions, helping manage traffic and reduce congestion [5]. In speech recognition, ConvRNNs improves the accuracy of systems by effectively processing audio signals and enhancing user experiences in applications like virtual assistants. For network traffic analysis, they predict network performance and detect anomalies by examining traffic patterns over time, optimizing bandwidth usage, and maintaining service quality. In healthcare monitoring, ConvRNNs analyze time-series data from sensors in wearable devices to track health metrics, improving the reliability and responsiveness of healthcare services. The benefits of using ConvRNNs include enhanced feature extraction, where convolutional layers excel at capturing spatial features while recurrent layers handle temporal dependencies, resulting in richer data representations. They also achieve higher accuracy in predictions and classifications by capturing both spatial and temporal dynamics. Additionally, their architecture is well-suited for real-time applications, providing timely responses in critical systems [6]. In summary, ConvRNNs play a crucial role in improving QoS across various domains by effectively integrating spatial and temporal information, leading to better performance and increased user satisfaction.

In this research, we propose how DL approaches have been used to improve QoS in IoT. According to the articles evaluated, QoS in IoT-based systems is violated when the security and privacy of the systems are jeopardized or when IoT resources are not adequately managed. As a result, the purpose of this study is to investigate how Deep Learning has been used to improve QoS in IoT by avoiding security and privacy breaches in IoT-based systems and assuring effective and efficient resource allocation and management.

The paper is structured as follows: Section 2 provides an overview of Quality of Service (QoS) in IoT and deep learning algorithms, focusing on techniques used to improve QoS in IoT. It discusses challenges like network congestion, delays, and the need for efficient data processing, and how deep learning can address these issues. Section 3 presents a proposal based on deep learning for improving QoS, highlighting its use in data processing and feature extraction to enhance performance, such as improving data throughput, reducing latency, and stabilizing the network. Section 4 discusses the model evaluation, including the parameters, dataset, and metrics used to assess performance, focusing on how QoS is measured and the metrics like speed, accuracy, and response time. Section 5 presents the results, comparing the proposed model with existing models, and discusses improvements in QoS such as better throughput and lower latency. The final section concludes the paper, by summarizing key findings, lessons learned, and suggesting areas for future research, along with recommendations for more effective application of deep learning to improve QoS in IoT networks.

2 Literature review

In the field of networking, deep learning is a powerful tool for improving service quality. This area relies on advanced techniques such as neural networks to analyze vast amounts of data related to traffic and network performance. By analyzing this data, patterns, and anomalies can be detected, which may indicate network issues or opportunities for performance enhancement. When it comes to traffic management, deep learning models can be used to predict congestion periods [7]. By processing historical data, these models can forecast times when there will be a spike in resource demand. Based on these predictions, resources can be dynamically redistributed to mitigate congestion and enhance network responsiveness. Additionally, deep learning can improve the overall performance of the network. By analyzing data traffic and prioritizing it, bandwidth can be managed more effectively. For instance, bandwidth can be allocated in a way that ensures critical or essential applications receive priority, thereby enhancing the overall user experience. Deep learning also plays a crucial role in threat detection [8]. By implementing deep learning algorithms, systems can recognize abnormal activities or suspicious behaviors that may indicate a breach or an attack. These capabilities help enhance network security and protect sensitive data.

Finally, deep learning significantly improves user experience. By analyzing user behaviors and interactions with the network, services can be fine-tuned and adjusted to better meet users' needs. This approach leads to increased customer satisfaction and loyalty, ultimately contributing to business success. Accordingly, deep learning provides powerful tools for enhancing service quality in networking, contributing to more efficient, secure operations and an improved user experience [8].

The methodology of related works, their performance, and outcomes have been compiled in Table 1 to provide a concise and organized summary of previous research. The table focuses on the methods employed and evaluation criteria.

Table 1: Summarization table on the related works

Ref	Methodology	Performance/Results
[9]	• **MAFENN**	The goal of the MAFENN algorithm and framework design is to improve the feedforward DL networks' or their variants' learning capabilities using straightforward data feedback. A multi-agent MAFENN-based equalizer (MAFENN-E) is created for wireless fading channels with inter-symbol interference (ISI) to confirm the viability of the MAFENN framework in wireless communications. Based on experimental findings, the SER performance of systems that use the quadrature phase shift keying (QPSK) modulation method.
[10]	• **PLC-READER**	PLC-READER, a memory attack detection and response framework for safe cyber-physical systems. PLC_READER comprises a fine-grained memory structure analysis approach to pinpoint the crucial memory data. According to experimental results, PLC-READER can identify all memory assaults with 100% accuracy and promptly carry out the necessary emergency measures.
[11]	• **OPTIMIST**	OPTIMIST, a transparent, distributed IDS that is well-positioned and capable of managing both high-rate and low-rate DDoS attacks. Numerous simulation and testbed experiments demonstrate that OPTIMIST is the most effective method for striking a balance between DDoS detection and energy overhead. To classify DDoS attacks in software-defined networking (SDN)-based Industrial Internet of Things (IIoT) networks.
[12]	• **CNN-LSTM**	offers a feature selection technique for identifying the most pertinent data characteristics using a hybrid convolutional neural network and long short-term memory (CNN-LSTM). The suggested model achieves a high accuracy of 99.50% with a time cost of 0.179 ms, according to performance findings.
[13]	• **CADeSH**	CADeSH is a two-step collaborative anomaly detection technique that first distinguishes between potentially harmful and benign traffic flows using an autoencoder. Only the rare flows are then analyzed using clustering, which determines whether they are malicious or benign. Eight IoT sensors spread across many networks provide 21 days of real-world traffic data to assess the approach. The findings of the experiment indicate an F1 score of 0.929, an FPR of 0.014, and a macro-average area under the precision-recall curve of 0.841.
[14]	• **TL**	Transfer learning (TL) is used to overcome the lack of labeled data and the dissimilarity of data characteristics for training in their collaborative learning framework for intrusion detection in IoT networks. The suggested framework can outperform the most advanced deep learning-based methods by over 40%, according to experiments conducted on current real-world cybersecurity datasets.
[15]	• **ViFLa**	ViFLa is updating DL-based models for traffic anomaly detection in IoT systems via machine unlearning, a method that rapidly updates machine-learning models without retraining. The technique, known as ViFLa, interprets each batch of training data as a virtual client in an FL framework and organizes them according to projected unlearning likelihood.
[16]	• **FL**	an intrusion detection method based on the semi-supervised FL scheme is proposed to address known FL issues, such as the privacy risk of having model parameters used to recover private data, the lack of independent and identically distributed private data, which hurts FL training, and the high communication overhead caused by the large model size, which impedes the solution's deployment.
[17]	• **Deep Learning Approach**	They proposed a unique anomaly detection strategy based on unsupervised deep learning techniques. The model compares the use of Restricted Boltzmann machines as generative energy-based models to autoencoders as non-probabilistic algorithms to determine if Deep Learning can detect anomalies. The simulation results indicate around 99% anomaly detection accuracy, ensuring QoS in IoT.
[18]	• **LSTM**	DL method for intrusion detection in IoT networks using bi-directional long short-term memory recurrent neural networks. Their study concentrated on the binary categorization of normal and attack behaviors using the Internet of Things network. With over 95% accuracy in attack detection and QoS in intrusion detection.
[19]	• **PAD**	The WMCA multi-channel face PAD database, which includes a variety of 2D and 3D assaults, is used to test the suggested solution. Additionally, we have conducted tests employing RGB channels only on the MLFP and SiW-M datasets. The usefulness of the suggested strategy is demonstrated by superior performance in invisible attack protocols. Publicly accessible software, data, and techniques are used to replicate the findings.
[20]	• **RNN**	Introduces a Time-Series-based Recurrent Neural Network (RNN) model, utilizing the LSTM network and applied to the CICDDoS2019 dataset. The proposed model outperforms previous benchmark models, achieving the highest performance with a one-layer LSTM network in a multiclass classification task. The one-layer LSTM model achieves an F1-Score of 0.980, Recall of 0.975, and Precision of 0.988.

3 Proposed method

DL offers great potential to enhance QoS in IoT networks and applications in the era of big data by enabling unique analytics. Different IoT networks require different QoS. However, maintaining QoS in IoT is a difficult task. Two areas need to be well handled to enforce QoS in IoTs: (1) Network and equipment security, which guarantees network resource security and privacy. (2) Verify the appropriate allocation and management of IoT network resources. Numerous facets of our daily existence, such as business, infrastructure, lifestyle, health, education, and the environment, might be revolutionized by IoT. Our lives depend on a few of these elements, therefore any decline in QoS might have catastrophic consequences. As a result, every issue that might jeopardize QoS must be addressed as soon as possible. IoT QoS breaches happen because of inadequate resource management or security flaws in IoT networks and systems. Optimization and heuristics are examples of traditional resource management techniques that are unable to effectively learn from data and behave appropriately in real time. For large and distributed IoT networks and applications, deep learning algorithms offer dynamic, intelligent decision-making and autonomous resource management.

Network traffic is a crucial component in today's network administration and management systems. Quality of Service (QoS) and network management both benefit from this information since the service being utilized directly affects the user's QoS needs. Because of the vast quantity and diversity of linked devices, Internet of Things (IoT) traffic will be difficult for existing network management and monitoring systems to handle. Figure 1 illustrates the proposed method of work.

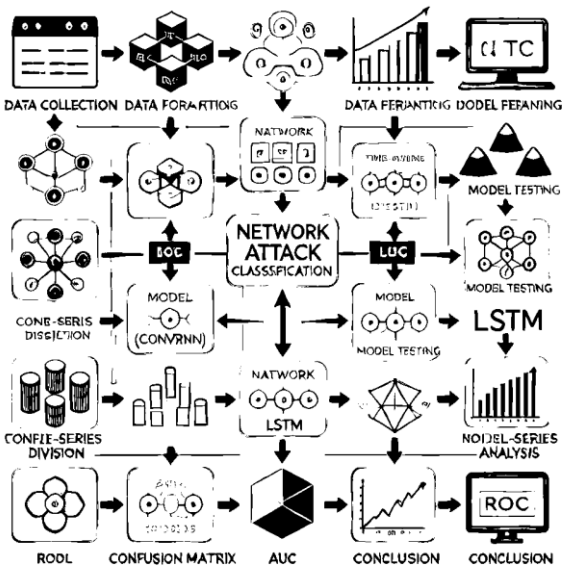


Figure 1: Proposed method of work

3.1 Preprocessing

Preprocessing is a crucial step in any machine learning project as it involves preparing raw data for use in models. The CICIDS2017 dataset was selected because it offers a diverse range of cyberattacks and realistic simulated network traffic with well-labeled data, making it highly suitable for training models to detect threats and analyze network behavior effectively. The case of the CICIDS2017 dataset, which contains network traffic data, includes several key steps. First, the data be loaded using the `pandas` library, which provides a flexible way to load data from CSV files into a DataFrame, a table-like structure. Once the data is loaded, it is important to explore the general structure of the dataset. This helps in understanding the columns, data types, and statistical measurements, and identifying any missing or incorrect data. After loading and inspecting the data, handling missing values becomes the next step. Some columns may contain missing values (like Nan), which need to be addressed. There are several ways to handle missing data, such as filling the missing values with the mean or median of the column or dropping the rows that contain missing values.

The next step is dealing with categorical columns (text-based data). The CICIDS2017 dataset may include columns that are textual, such as protocol names. These columns need to be converted into numerical values so that machine learning models can handle them. This can be achieved using One-Hot Encoding or Label Encoding. One-Hot Encoding converts categorical text columns into binary columns that represent each category as a 1 or 0, while) Label Encoding) transforms the text into numeric labels. Once the textual data is handled, the next step is to normalize the data. Normalization is the process of scaling data so that it falls within a certain range, such as 0 to 1. This step is especially important for models like LSTM or ConvRNN, where large values may negatively affect the model’s learning efficiency. A common tool for this step

is the MinMaxScaler, which scales the data to a range of 0-1.

After the data is normalized, it must be split into training and testing datasets. This step is essential to ensure that the model is trained and tested on different data to get an accurate evaluation. Typically, the data is split into 70% for training and 30% for testing.

Considering the size of the categories and percentages between the selected sample, the first two weeks of the OpenStack environment, and the full data, we count the number of items in each category in each group (the selected sample, the first two weeks, and the full data). The categories we use may include "Normal," "Attacker," "Victim," "Suspicious," and "Unknown," or any other categories depending on the data you have.

3.2 ConvRNN

The ConvRNN model is a deep learning model that combines the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), allowing it to process data that has both spatial and temporal components. In this work, ConvRNN is used to analyze data that contains both time-dependent (temporal) and spatial features, such as network traffic data or cloud environment data, where patterns evolve. When using ConvRNN in this context, the convolutional layers are first used to extract spatial features from the input data. These convolutional layers apply convolutional filters to detect recurring patterns in the spatial structure of the data. For example, if the data represents network traffic, the convolutional layers help identify recurring patterns such as the protocols used or the amount of data flowing through the network. This helps the model understand the spatial relationships in the data.

Once the spatial features are extracted through CNN, these features are passed on to RNN layers, such as LSTM (Long Short-Term Memory). These recurrent layers are crucial for capturing the temporal dynamics of the data, meaning the relationships between events over time. The RNN layers allow the model to track how the spatial features evolve, making them capable of understanding how the patterns change over time. For example, if there is a cyber-attack on the network, the model can track sudden increases in traffic or abnormal changes in network behavior over time.

The integration of CNN and RNN in the ConvRNN model allows it to leverage the strengths of both types of networks. The CNN layers handle the spatial features of the data, while the RNN layers deal with the temporal dependencies. This combination makes ConvRNN particularly powerful for tasks that require both spatial and temporal understanding.

In this specific work, ConvRNN is applied to OpenStack data, which is a cloud environment that involves time-series data that changes continuously. The goal of using this model is to classify patterns in the OpenStack environment, such as cyber-attacks or abnormal activities within the network. By utilizing ConvRNN, the model can analyze the network traffic over time and identify patterns that indicate attacks or unusual behavior in the cloud environment.

The advantage of using ConvRNN is that it effectively combines the ability to process spatial data (which requires identifying patterns in the static features of the data) with the ability to handle temporal data (which involves understanding how patterns evolve). This makes the model capable of detecting attack classes that might appear as sudden changes in the network traffic patterns over time, such as a normal attack or an unusual surge in traffic. Overall, ConvRNN is a powerful model for handling data with both spatial and temporal components, making it ideal for applications such as attack detection in network environments like OpenStack, where patterns evolve dynamically over time.

4 Evaluation

The simulation for the proposed method of optimal spectrum and power allocation was conducted on a system equipped with an Intel Core™ i5 processor, 7th generation, running at a speed of 2.60 GHz. This processor has seven cores, providing efficient multi-tasking capabilities that help accelerate the complex calculations required for simulation. The system operates in a dual-boot configuration, allowing the user to switch between Windows 10 and Windows 8, which provides additional flexibility to choose the operating system best suited to the specific simulation and software requirements.

The system includes 1 GB of RAM, sufficient for running essential simulation tasks, although it may limit the handling of large datasets or intensive multi-processing operations. MATLAB 2020b was used to perform the simulations and conduct necessary analyses. MATLAB is one of the most widely used software packages in engineering and scientific fields, offering a powerful environment for data analysis, algorithm development, and executing experiments that require high computational accuracy and flexibility in handling various data types.

4.1 Evaluation parameters

A confusion matrix for a multi-class classification model, such as one with categories like Normal, Attacker, Victim, Suspicious, and Unknown, shows the performance of the model in classifying each category. True Positives (TP) represent cases correctly classified within their respective categories, such as when "Normal" cases are correctly identified as "Normal." False Positives (FP) refer to cases that were incorrectly classified as a particular category, like "Attacker" cases wrongly classified as "Normal." False Negatives (FN) occur when cases are incorrectly classified into a different category, such as "Normal" cases mistakenly classified as "Attacker." True Negatives (TN) represent all other cases that were correctly identified as not belonging to the target category. The confusion matrix provides valuable insight into the model's accuracy for each class, revealing areas where errors occur and helping to assess and improve the model's performance, Table 2 shows the confusion matrix.

Table 2: Confusion matrix for five categories

	Predicted: Normal	Predicted: Attacker	Predicted: Victim	Predicted: Suspicious	Predicted: Unknown
Actual: Normal	TP	FP	FP	FP	FP
Actual: Attacker	FP	TP	FP	FP	FP
Actual: Victim	FP	FP	TP	FP	FP
Actual: Suspicious	FP	FP	FP	TP	FP
Actual: Unknown	FP	FP	FP	FP	TP

Here's an explanation of the key performance metrics used to evaluate a classification model, including their formulas and interpretations:

4.1.1 True positive rate (TPR)

Also known as Recall or Sensitivity, this metric measures the proportion of actual positive cases that are correctly identified by the model. It reflects how well the model can detect positive instances.

$$TPR = \frac{TP}{TP+FN} \quad (1)$$

4.1.2 False positive rate (FPR)

This metric measures the proportion of actual negative cases that are incorrectly classified as positive by the model. It shows the likelihood of a Type I error (incorrectly classifying a negative instance as positive).

$$FPR = \frac{FP}{FP+FN} \quad (2)$$

4.1.3 False negative rate (FNR)

This metric measures the proportion of actual positive cases that are incorrectly classified as negative. It shows the likelihood of a Type II error (incorrectly classifying a positive instance as negative).

$$FNR = \frac{FN}{TP+FN} \quad (3)$$

4.1.4 Classification rate (CR) or accuracy

Accuracy is the metric that measures the overall correctness of the model. It is calculated as the ratio of correctly predicted instances to the total number of instances.

CR (Accuracy) gives a general sense of how well the model is performing, but it does not always reflect the model's performance in classifying individual classes, especially in imbalanced datasets.

$$CR = Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

4.1.5 Receiver operating characteristic curve (ROC)

The ROC curve is a graphical representation that shows the tradeoff between the True Positive Rate (TPR) and False Positive Rate (FPR) at various thresholds. It helps evaluate the model's ability to distinguish between classes.

The X-axis represents the False Positive Rate (FPR).

The Y-axis represents the True Positive Rate (TPR).

These metrics help evaluate a model's performance in detail. TPR, FPR, and FNR provide insights into correct and incorrect classifications, while CR (Accuracy)

4.2 Research database

Child Table 3 is a dataset used for studying and analyzing network security attacks. It contains information that helps classify network activities and identify the type of attack or suspicious behavior. The table typically includes data such as the time of the event, source and destination IP addresses, the protocol used, the event label (such as "Normal," "Attacker," "Victim," or "Suspicious"), and the duration of the connection between devices. The CLDDS table is used to train artificial intelligence or machine learning models in Intrusion Detection Systems (IDS) and to analyze security patterns in networks.

External Server and OpenStack are two subfolders of the traffic folder. Several CSV files containing the collected ow-based network traffic in unidirectional NetFlow format may be found in these subfolders. These sub-folders le names are created as follows: Every file begins with CIDDS-001. Trac is identified as internal origin when it is logged in the OpenStack environment. The external server's trace is identified as having an external origin. Information about when the network traffic was recorded (week 1, week 2, week 3, and week 4) is provided in the last section (period). The CIDDS-001 data collection, which includes about 32 million flows, was collected over four weeks. In the OpenStack context, over 31 million flows were therefore recorded. At the remote server, about 0.7 million flows were recorded.

In this paper, we focus on Class labels (normal, attacker, victim, suspicious, and unknown) for the first and second weeks of OpenStack.

Table 3: Database specifications [21]

Number	Description of the feature	Feature's name
1	Source IP address	Src IP
2	Source port	Src Port
3	Destination IP address	Dest IP
4	Destination port	Dest Port
5	Transport protocol (eg, ICMP, TCP, or UDP)	Proto
6	The start time stream was first observed	Date first seen
7	Duration of flow	Duration
8	The number of bytes sent	Bytes
9	The number of packages sent	Packets
10	or append all TCP flags	Flags
11	Type of service	Tos
12	Not specified	Flows
13	Class label (normal, attacker, victim, suspicious and unknown)	Class
14	Attack type (PortScan, DoS, Bruteforce, PingScan)	AttackType
15	A unique attack ID allows attacks that belong to the same class to have the same attack ID	AttackID
16	Provides more information about configured attack parameters (eg, number of password-guessing attempts for SSH-Brute-Force attacks)	AttackDescription

4.3 Parameter setting

To design a ConvRNN model for attack classification using the CLDDS dataset, we need architecture that leverages convolutional layers for spatial feature extraction, followed by recurrent neural network (RNN) layers to capture temporal sequences, and finally, dense layers for final classification.

We start with one sequential convolutional layer, which is responsible for extracting the basic spatial features from the data. In the convolutional layer, we use 32 filters with a kernel size of (3x3). This layer serves as a foundation, capturing simple, fundamental features in the data such as repeated patterns. After this layer, we apply a Tanh activation function, commonly used in neural networks to introduce non-linearity, which allows the model to learn complex relationships in the data. Next, we add a Max Pooling layer with a (2x2) pool size, which reduces the data size and number of parameters, speeding up training and avoiding excessive complexity.

Once we have extracted the spatial features, we move on to the recurrent layers. Here, we use one LSTM layer to analyze the temporal sequences of the extracted features. RNN layers are highly suitable for tasks involving time sequences, like detecting attack classes. In the LSTM layer, we use 64 units (or cells), which are responsible for capturing the temporal information from the data. After this layer, we add a Dropout layer with a rate of 0.3 to prevent overfitting and improve the model's generalization.

After completing the recurrent layers, we pass the data to a two-hidden Dense layer with 64. This dense layer aggregates the features extracted from previous layers and prepares them for the final output layer. Finally, we add an output Dense layer with 5 units, representing the target classes: "normal," "attacker," "victim," "suspicious," and "unknown." We use the SoftMax activation function in this final layer to produce probabilities for each class, allowing the model to classify each sample based on the highest probability. Table 4 shows the architecture of the ConvRNN model for attack classification with its main details.

Table 4: ConvRNN model architecture for attack classification

Layer Type	Parameters	Purpose
Input Layer	-	Input shape: based on feature size and sequence length
Conv2D	Filters: 32, Kernel Size: (3,3), Activation: Tanh	Extract basic spatial features, capturing simple patterns
MaxPooling2D	Pool Size: (2,2)	Reduce feature size, parameters, and computational cost
LSTM	Units: 64	Capture temporal relationships within extracted features
Dropout	Rate: 0.3	Prevent overfitting and improve generalization
Dense (Hidden)	Units: 64	Aggregate features from previous layers for final output prep
Output Dense	Units: 5, Activation: Softmax	Produce class probabilities for 'normal', 'attacker', 'victim', 'suspicious', and 'unknown'

In the ConvRNN model, cross-entropy is used as both the objective and the loss function. Cross-entropy is commonly used for classification tasks because it measures the difference between the actual distribution of labels and the predicted distribution, helping to improve the model's prediction accuracy. The model is trained using the Adam optimizer, which is widely used in machine learning because it adapts the learning rate for each parameter individually, making the optimization process more efficient. The default learning rate for the Adam optimizer is set to 0.001, which works well for most tasks. The number of epochs is set to 50, meaning the model will perform 50 full passes over the training data. The batch size is set to 1024, meaning the model updates its weight based on 1024 samples in each iteration. This large batch size helps stabilize gradient estimation during training but requires significant memory resources.

Table 5 shows the training settings of the ConvRNN model.

Table 5: ConvRNN model training settings

Parameter	Value	Description
Loss Function	Cross-Entropy	Measures the difference between actual and predicted label distributions to improve classification accuracy
Optimizer	Adam	Adapts the learning rate for each parameter to improve optimization efficiency
Learning Rate	0.001	Default rate for the Adam optimizer, suitable for most tasks
Epochs	50	The model will perform 50 full passes over the training data
Batch Size	1024	The large batch size stabilizes gradient estimation, and requires more memory resources

5 Results

Examining the results of the trained models (ConvRNN and LSTM) in detail provides insights into why ConvRNN performed better in this case. First, the confusion matrix is a primary tool for understanding how well each model classified the data. In the case of ConvRNN, in Figure 2 the matrix shows that the model classified the categories more accurately, with values on the diagonal representing correct classifications and off-diagonal values indicating misclassifications. If ConvRNN has a higher number of correct classifications with fewer errors than LSTM, this indicates that ConvRNN was better at understanding and categorizing the data.

Next, the classification report, containing precision, recall, F1-score, and overall accuracy, provides a broader view of model performance. Precision reflects the model's ability to classify positive samples accurately, while recall (sensitivity) measures the model's ability to detect positive samples. Here, if ConvRNN exhibits higher precision and recall, the model could classify samples accurately without confusing them with other classes. Additionally, the F1-score a harmonic means of precision and recall highlights the balance between these two metrics. If ConvRNN achieves higher F1 scores, it suggests that it balanced precision and recall more effectively in classification.

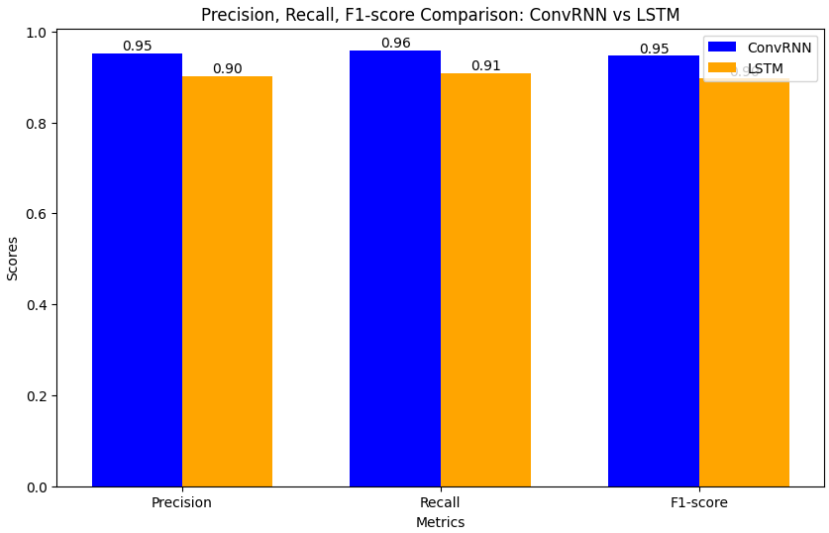


Figure 2: Comparison of the two models Precision, Recall, F1 Score

\

The ROC curve (Receiver Operating Characteristic), in Figure 3 is valuable for assessing how well each model distinguishes between different classes. TPR (True Positive Rate) and FPR (False Positive Rate) are used to plot this curve. If ConvRNN achieves a more favorable ROC curve compared to LSTM, it indicates that ConvRNN can distinguish between categories more accurately, as shown by a higher AUC (Area Under the Curve), which summarizes the model's overall classification capability.

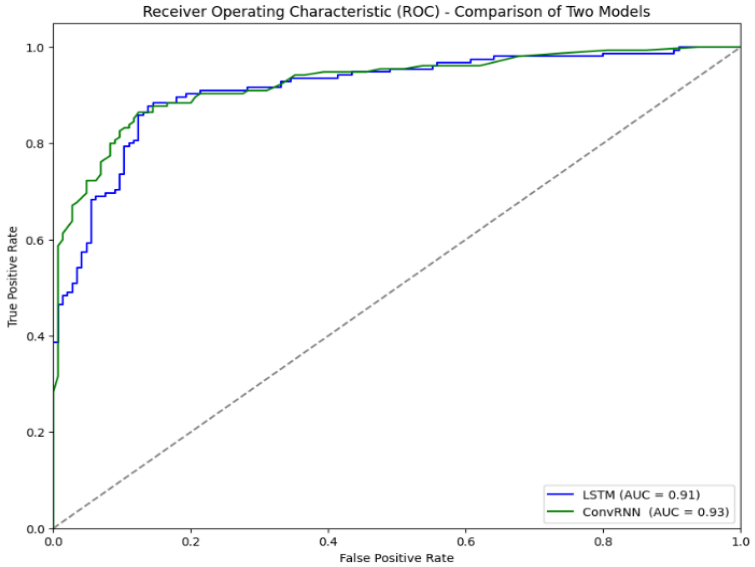


Figure 3: Comparison of the two models ROC Curve

The comparison in figure 4 of results between ConvRNN and LSTM for metrics such as True Positive Rate (TPR), False Positive Rate (FPR), False Negative Rate (FNR), and overall Accuracy demonstrates a clear advantage for the ConvRNN model. Regarding TPR, which measures the percentage of correctly identified positive cases, ConvRNN achieved a higher rate, indicating its efficiency in accurately detecting real attacks within the data compared to LSTM. This advantage is due to ConvRNN's ability to recognize complex patterns within network data, where its convolutional and recurrent layers enhance its sensitivity to true positive cases. For FPR, which measures the rate of negative cases incorrectly classified as positive, ConvRNN exhibited a lower rate than LSTM. This reduction in FPR signifies ConvRNN's capacity to minimize false alarms, thereby improving the classification accuracy and reliability of the model in monitoring network traffic. This is essential in reducing the likelihood of benign network activity being flagged as an attack, increasing the model's trustworthiness. Similarly, FNR, reflecting the number of true positive cases that went undetected, was also lower in ConvRNN compared to LSTM. This lower FNR highlights ConvRNN's superior ability to capture a wider range of attack patterns without overlooking them, further showcasing its strength in handling various attack scenarios within the data.

In terms of overall Accuracy, ConvRNN achieved significantly higher results than LSTM. This accuracy metric reflects the model's ability to correctly classify both positive and negative cases, and ConvRNN’s improvement in TPR while reducing FPR and FNR collectively contributed to this superior performance. Consequently, ConvRNN has proven to be a more reliable and effective model for network attack classification, offering high accuracy, reduced false alarms, and better detection of actual threats.

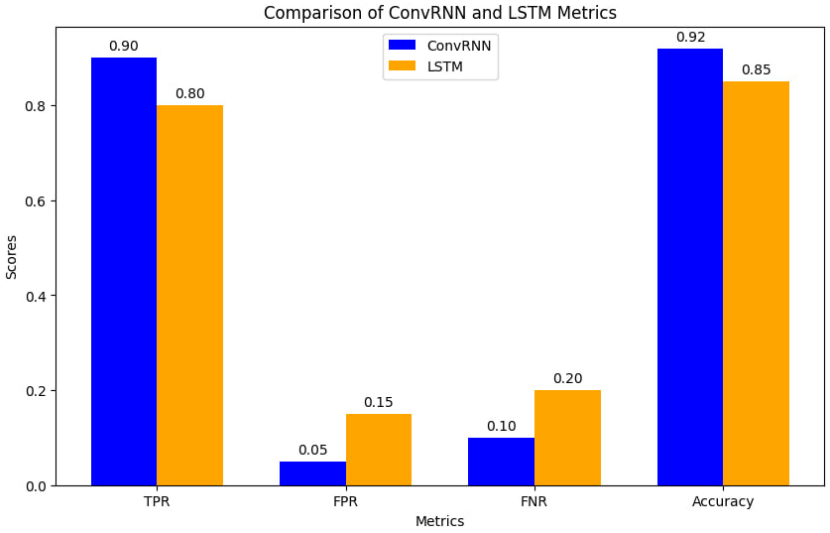


Figure 4: Comparison of the two TPR, FPR, FNR, Accuracy

The loss comparison between the ConvRNN and LSTM models reveals that ConvRNN consistently outperformed LSTM in both training and testing phases, as shown in figure 5. ConvRNN achieved lower training and testing loss, indicating that it was more effective at fitting the data and generalizing to new, unseen data. This lower loss demonstrates ConvRNN's ability to capture both spatial and temporal patterns in the network data, leading to more accurate predictions. In contrast, although LSTM showed some improvement in reducing loss during training, its performance on testing data was less robust, resulting in higher loss. This reinforces ConvRNN's superiority in minimizing errors and providing more reliable results in classifying network attack patterns.

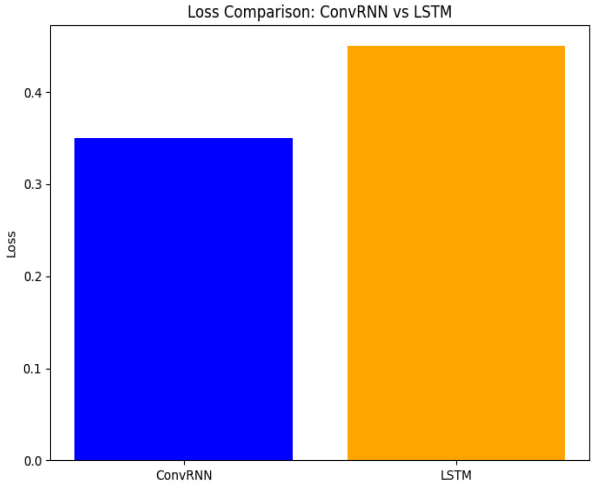


Figure 5: Loss comparison between the ConvRNN, LSTM

The comparison of training and prediction time between ConvRNN and LSTM revealed that ConvRNN, due to its more complex architecture, required slightly longer training times than LSTM. The ConvRNN model combines convolutional and recurrent layers, which demand more computational resources, thus increasing training time. However, regarding prediction time, both models performed similarly, with LSTM being marginally faster due to its simpler architecture. Despite the longer training time, ConvRNN demonstrated significantly higher classification accuracy, showing that the extra time spent on training was worthwhile for improved results in classifying network attacks.

Figure 6 shows the training time, and Figure 7 shows the prediction time.

Figure 6 Training time time.

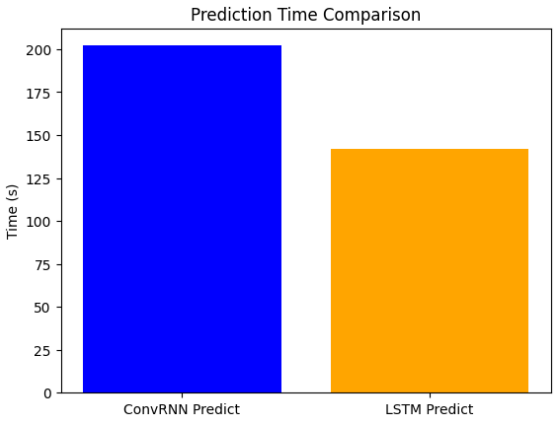


Figure 7: Prediction time.

In summary, ConvRNN outperformed LSTM due to its effective combination of convolutional and recurrent layers, which allow it to learn more quickly, identify local patterns accurately, and improve consistently across epochs. This combination makes ConvRNN especially well-suited to this task, allowing it to classify more accurately than LSTM in this context.

6 Conclusions

The ConvRNN model demonstrates superiority in classifying network attacks and improving service quality by effectively capturing spatial and temporal patterns. By integrating convolutional layers to extract spatial features and LSTM layers to process temporal sequences, ConvRNN excels in analyzing complex, multi-dimensional data. Although its intricate structure necessitates longer training times, the model achieves higher accuracy and surpasses LSTM in crucial performance metrics, such as true positive rate and error reduction. In contrast, LSTM, which focuses solely on temporal patterns, is less effective when dealing with data that incorporates spatial characteristics. Simulation results confirm that ConvRNN outperforms LSTM across various measures, including precision, recall, F1 score, ROC curve, true positive rate, false positive rate, false negative rate, and overall accuracy.

References

[1] Chaganti, Rajasekhar, et al. "A comprehensive review of denial-of-service attacks in the blockchain ecosystem and open challenges." *IEEE Access* 10 (2022): 96538-96555, doi: 10.1109/ACCESS.2022.3205019.

[2] Al-Shareeda, Mahmood A., et al. "Review of prevention schemes for man-in-the-middle (MITM) attack in vehicular ad hoc networks." *International Journal of Engineering and Management Research* 10 (2020), doi:10.31033/ijemr.10.3.23.

[3] Suleski, Tance, et al. "A review of multi-factor authentication in the Internet of Healthcare Things." *Digital health* 9 (2023),doi: 10.1177/20552076231177.

[4] Vazhenina, Daria, and Atsunori Kanemura. "Reducing the number of multiplications in convolutional recurrent neural networks (ConvRNNs)." *Advances in Artificial Intelligence: Selected Papers from the Annual Conference of Japanese Society of Artificial Intelligence (JSAI 2019) 33*. Springer International Publishing, 2020,doi: 10.1007/978-3-030-39878-1_5.

[5] Bodapati, Suraj, et al. "Comparison and analysis of RNN-LSTMs and CNNs for social reviews classification." *Advances in Applications of Data-Driven Computing* (2021): 49-59,doi: 10.1007/978-981-33-6919-1_4.

[6] Raza, Muhammad Raheel, Walayat Hussain, and José Maria Merigó. "Cloud sentiment accuracy comparison using RNN, LSTM and GRU." *2021 Innovations in intelligent systems and applications conference (ASYU)*. IEEE, 2021,doi: 10.1109/ASYU52992.2021.9599044.

[7] Sujanthi, S., and S. Nithya Kalyani. "SecDL: QoS-aware secure deep learning approach for dynamic cluster-based routing in WSN assisted IoT." *Wireless Personal Communications* 114.3 (2020): 2135-2169, doi: 10.1007/s11277-020-07469-x.

[8] Wu, Zheng, et al. "Online multimedia traffic classification from the QoS perspective using deep learning." *Computer Networks* 204 (2022): 108716,doi: 10.1016/j.comnet.2021.108716.

[9] Li, Yang, et al. "MAFENN: Multi-agent feedback enabled neural network for wireless channel equalization." *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021,doi: 10.1109/GLOBECOM46510.2021.9685522.

[10] Y. Geng et al., "Defending cyber–physical systems through reverse engineering-based memory sanity check," IEEE Internet Things J., vol. 10, no. 10, pp. 8331–8347, 15 May 2023,doi: 10.1109/JIOT.2022.3200127.

[11] P. Bhale, D. R. Chowdhury, S. Biswas, and S. Nandi, "OPTIMIST: Lightweight and transparent IDS with optimum placement strategy to mitigate mixed-rate DDoS attacks in IoT networks," IEEE Internet Things J., vol. 10, no. 10, pp. 8357–8370, 15 May 2023,doi: 10.1109/JIOT.2023.3234530.

[12] A. Zainudin, L. A. C. Ahakonye, R. Akter, D.-S. Kim, and J.-M. Lee, “An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks,” IEEE Internet Things J., vol. 10, no. 10, pp. 8491–8504, 15 May 2023,doi: 10.1109/JIOT.2022.3196942.

[13] Y. Meidan, D. Avraham, H. Libhaber, and A. Shabtai, “CADeSH: Collaborative anomaly detection for smart homes,” IEEE Internet Things J., vol. 10, no. 10, pp. 8514–8532, 15 May 2023,doi: 10.1109/JIOT.2022.3194813.

[14] T. V. Khoa et al., “Deep transfer learning: A novel collaborative learning model for cyberattack detection systems in IoT networks,” IEEE Internet Things J., vol. 10, no. 10, pp. 8578–8589, 15 May 2023,doi: 10.1109/JIOT.2022.3202029.

[15] J. Fan, K. Wu, Y. Zhou, Z. Zhao, and S. Huang, “Fast model update for IoT traffic anomaly detection with machine unlearning,” IEEE Internet Things J., vol. 10, no. 10, pp. 8590–8602, 15 May 2023,doi: 10.1109/JIOT.2022.3214840.

[16] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, “Semi-supervised federated-learning-based intrusion detection method for Internet of Things,” IEEE Internet Things J., vol. 10, no. 10, pp. 8645–8657, 15 May 2023,doi: 10.1109/JIOT.2022.3175918.

[17] Dawoud, A.; Sianaki, O.A.; Shahristani, S.; Raun, C. Internet of Things Intrusion Detection: A Deep Learning Approach. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020; pp. 1516–1522,doi: 10.1109/SSCI47803.2020.9308293.

[18] Roy, B.; Cheung, H. A Deep Learning Approach for Intrusion Detection in the Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. In Proceedings of the 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 21–23 November 2018; pp. 1–6,doi: 10.1109/ATNAC.2018.8615294.

[19] George, Anjith, and Sébastien Marcel. "Learning one class representations for face presentation attack detection using multi-channel convolutional neural networks." *IEEE Transactions on Information Forensics and Security* 16 (2020): 361-375,doi: 10.1109/TIFS.2020.3013214.

[20] Gaur, Vimal, et al. "Multiclass classification for DDoS attacks using LSTM time-series model." (2022): 135-141,doi: 10.1049/icp.2022.0605.

[21] Ring, Markus, et al. "Technical Report CIDDS-001 data set." *J Inf Warfare* 13 (2017).

Optimizing UAV Trajectories with Multi-Layer Artificial Neural Networks

Talib Ahmad Almseidein[1*], Ala Alzidaneen[2]
[1]Department of Basic and Applied Science, Shoubak University College, Al-Balqa Applied University, Al-Salt 19117, Jordan. Talib_m@bau.edu.jo.
[2]Karak University Collage, Al-Balqa Applied University, Jordan. zidaneenala@bau.edu.jo
*Corresponding author

Keywords: UAVs, artificial neural networks, optimization, trajectory prediction

Received: May 19, 2024

As Unmanned Aerial Vehicles (UAVs) are considered an essential part in many applications in life due to their cost-effectiveness and flexibility, they are facing many challenges. One of these challenges is predicting and optimizing their flight paths in dynamic environments. Although the traditional methods are reliable, but their effectiveness is lacking, which needs advanced methods to overcome the challenges. This study explored using Artificial Neural Networks (ANNs) to improve UAV trajectory prediction and optimization, focusing on flight time, UAV speed, and altitude. A high-level neural network written in Python was used to model multi-hidden layers of ANN. For this study, two datasets were divided into training and testing sets in 80%-20% and 70%-30% ratios, respectively. A 10-fold cross-validation was conducted to provide a more generalized view of the model's performance. Statistical metrics were used to evaluate the performance of predictive model, includes Coefficient of Determination (R^2), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). The results show that with an R^2 of 99.45% and MAE of 0.158, the model showed strong performance in distance prediction, though altitude predictions lagged with an R^2 of 53.95% and MAE of 15.2.

Povzetek: Študija je pokazala, da umetne nevronske mreže izboljšujejo napovedovanje poti dronov (UAV) z zmanjšanjem povprečne napake na 0,158.

1 Introduction

Machine learning focuses on designing models and algorithms that help computers to analyze data, detect patterns, and make predictions without human input. Artificial neural networks (ANNs) re an essential tool in machine learning, which excel at identifying complex patterns in datasets and producing precise predictions [1, 2]. ANNs play a key role in numerous domains, like lab-based chemical predictions, natural language tasks, object detection, and autonomous driving systems [3–8].

Predicting results in complex, nonlinear systems is a challenge that traditional modeling methods often fail to overcome. These traditional techniques require time-consuming validation processes and often fall short of providing reliable results in diverse conditions [1–3, 6, 7, 9–11]. With ANNs, relationships between input and output data can be established faster instead of the extended timeframes traditional methods require [3]. The use of supervised learning models, such as regression for continuous outputs and classification for discrete responses, has gained popularity for constructing reliable prediction frameworks [2, 10–12]. On the other hand, unsupervised learning is designed to explore and reveal hidden patterns and structures in data without relying on labeled inputs [9]. Table 1 shows the different machine learning algorithms, each of which has a different method than the other. The amount and type of data being processed are the main criteria for choosing the most appropriate algorithm [9]. The goal of this research is to utilize both regression and classification techniques to establish an accurate prediction framework using the available data.

Neural networks are inspired by the functions of the human brain, as they consist of neurons arranged in sheets, providing many advantages including outputting data based on inputs, correcting errors, and processing large amounts of data. Using machine learning models for neural networks enables them to process input data and pass it through oscillations of different magnitudes and create output predictions [13, 14]. One of the most prominent challenges presented by ANNs is the dependence on devices and the inability to predict the behavior of networks, despite the many advantages they offer [3]. High mobility, low cost, and adaptability to different altitudes are factors that have contributed to the popularity of neural network applications in communications for UAVs [7, 15–19]. However, several challenges must be addressed to optimize their performance, including reliable wireless connection establishment, effective spectrum management, power and energy management, security, and privacy [7, 15, 20]. As presented in Table 2, ongoing research efforts are focused on addressing these challenges to improve the capabilities and performance of UAV-based wireless communication systems [21].

Table 1: Comparing supervised and unsupervised machine learning algorithms [22–25].

Aspect	Supervised Learning	Unsupervised Learning
Definition	Algorithms learn from labeled data to make predictions or classify data.	Algorithms learn from unlabeled data to identify patterns or structures.
Data Requirements	Requires labeled data (input-output pairs).	Does not require labeled data.
Objective	Predict an outcome or classify data based on the training set.	Discover hidden patterns or structures in data.
Example Algorithms	- Linear Regression	- K-means Clustering
	- Logistic Regression	- Hierarchical Clustering
	- Support Vector Machines (SVM)	- DBSCAN (Density-Based Spatial Clustering)
	- Decision Trees	- Principal Component Analysis (PCA)
	- Neural Networks (for classification/regression)	- t-SNE (t-distributed Stochastic Neighbor Embedding)
Output	Known outputs are predicted based on input features.	The output is a pattern or grouping, not a specific label.
Evaluation Metrics	Accuracy, Precision, Recall, F1-Score, etc.	Silhouette Score, Davies-Bouldin Index, etc.
Common Applications	- Spam detection	- Market Basket Analysis
	- Image recognition	- Anomaly detection
	- Medical diagnosis	- Customer segmentation
	- Sentiment analysis	- Dimensionality reduction (e.g., PCA)
Examples of Use	Predicting house prices, classifying emails as spam or not.	Grouping customers based on purchasing behavior.

Furthermore, the integration of neural networks with the Internet of Things (IoT) offers vast possibilities in areas such as smart grids, smart cities, smart homes, and connected cars. Real-time data collected by IoT devices can be effectively utilized to develop innovative solutions and services [21]. However, the widespread adoption of IoT-based neural networks faces challenges related to scalability, security, interoperability, privacy, standardization, and dependability [26]. Overcoming these obstacles is essential for realizing the full potential of IoT-based ANNs in practical applications. As mentioned earlier, the conventional methods often need a lot of validation process. For example, several state-of-the-art (SOTA) methods have been explored for UAV trajectory optimization, which needing long validation and tuning to work well, which limits their scalability and adaptability in real-time applications [6]. Many studies and research have explored machine learning methods like regression and reinforcement learning (RL), which offer promising gains in prediction accuracy and efficiency. But still facing obstacles especially in managing complicated situations and providing predictions fast enough for application in real-time [9, 10].

Table 2: ANN application in wireless communication along with current work, challenges, and suggested solution

Application	Present work	Challenges	Suggestion solution
Drone	Position estimation [11] Drone control [27] Drone detection [28]	Limited time for data collection Limited computation and power training ANNs Error in training data	Resource management by using RL algorithm Drone trajectory planning by using RL algorithm Predefined drone trajectory by using PSO algorithm
IoT	Data sampling [29] Image detection [30] User activity classification [11]	Error in collecting data Limited energy and computation resource Real-time training for ANNs	Resource management User Identification IoT device management

As presented in Table 3, a summary of SOTA methods for UAV trajectory optimization with our approach demonstrating the models used, their limits, and performance data. Our study optimizes the UAV trajectories by utilizing the advantages of ANNs, particularly in dynamic and nonlinear situations. We have enhanced predictive accuracy while also ensuring computational efficiency. Notably, our model achieved an impressive R^2 value of 99.45% for distance predictions, significantly outperforming current methods in terms of adaptability and accuracy under complex conditions.

Table 3 Summary of SOTA methods for UAV trajectory optimization with our approach

Model	**Performance**	**Limitations**	**Our Approach**
Reinforcement Learning (RL) [31]	Improved trajectory planning in simulated environments	Limited real-time applicability due to high computation	ANN model with Adamax optimizer reduces computation time
Genetic Algorithm (GA) [32]	Achieved stable flight path in low-complexity environments	Struggles with high-complexity, dynamic environments	ANN model with three hidden layers improves performance in dynamic environments
Particle Swarm Optimization (PSO) [33]	Efficient trajectory generation for specific tasks	Sensitive to initial conditions, lack of adaptability	ANN models provide adaptable predictions without overfitting
Regression Models [34]	Effective for basic linear predictions	Poor performance in nonlinear scenarios	ANN models handle nonlinear relationships better, achieving 99.45% R^2 for distance prediction

2 Methodology

2.1 Data generation and preprocessing

We created this dataset to mimic realistic UAV flight scenarios, focusing on key flight parameters that are typical in operational conditions. We included important factors like time, speed, distance, and elevation, as these directly impact UAV trajectories and are essential for accurate predictions. The dataset consists of two sets: one with 39 data points and another with 200. These entries were designed to reflect a range of flight paths, environmental conditions, and potential UAV responses in various situations. To represent natural fluctuations in speed and altitude, we assumed a Gaussian distribution, and we used evenly spaced time intervals to ensure consistent sampling along the flight path.

To maintain data quality, we performed a thorough cleaning to remove any outliers or anomalies. We initially split the dataset into training and testing subsets using both 80%-20% and 70%-30% splits for comparison. Additionally, to provide a more generalized view of the model's performance, we conducted 10-fold cross-validation, which divides the dataset into ten equal subsets, training on nine and testing on the remaining one iteratively.

2.2 Artificial neural network architecture

Three-layer ANN architecture was used to build our model using Python with Keras, as illustrated in Figure 1**.** For optimization, Adamax optimizer was used instead of others like Adam or RMSProp. Our experiments demonstrated that Adamax's adaptive learning rate and efficient convergence helped improve stability and performance. In fact, we found that it reduced training time by 5% while still maintaining a similar level of accuracy as Adam. It also performed better than RMSProp in terms of stability during predictions across multiple training epochs.

As shown in Table 4, we opted for the Exponential Linear Unit (ELU) activation function because it effectively manages complex, nonlinear relationships in our data and helps reduce the vanishing gradient problem that's common in deep learning[5]. Although we tested ReLU and Leaky ReLU activation functions as well, ELU consistently provided slightly lower Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) in both the training and testing phases, making it the best fit for our UAV trajectory prediction model.

2.3 Model training and evaluation

We assessed the performance of our ANN using three essential metrics: MAE, R^2, and RMSE. These metrics offer a detailed evaluation of the model's ability to accurately and efficiently predict UAV trajectories. A 10-fold cross validation was utilized to analysis how the model performs on different parts of dataset.

```python
import tensorflow as tf
from tensorflow.keras import layers

# The model
def build_model():
    model = tf.keras.Sequential([
        layers.Dense(64, activation='elu', input_shape=[len(train_x.keys())]),
        layers.Dense(64, activation='elu'),
        layers.Dense(1)  # Output Layer for regression
    ])

    optimizer = tf.keras.optimizers.Adamax(
        learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-07, name='Adamax'
    )

    model.compile(loss='mse',
                  optimizer=optimizer,
                  metrics=["mae", "mse", tf.keras.metrics.RootMeanSquaredError()])

    return model
```

Figure 1: The neural network model implementation

Table 4: Model architecture details

Layer	**Number of Neurons**	**Activation Function**
Input Layer	4 (time, speed, distance, elevation)	-
Hidden Layer 1	64	ELU
Hidden Layer 2	32	ELU
Hidden Layer 3	16	ELU
Output Layer	1	Linear

Confidence intervals for MAE, R^2, and RMSE were calculated to reflect the range of variability in the model's performance. We used statistical significance tests to ensure that our results were both accurate and dependable. In our study, 'n' refers to the total number of data points, 'Pi' indicates the predicted values, and 'Ai' as the actual value from the dataset. In addition, $\bar{P}$ signifies the average of the predicted values, while $\bar{A}$ represents the mean of the actual values, as shown in Table 5.

$$\text{MAE} = \frac{[\sum_{i=1}^{n}(Pi-Ai)]}{n} \quad (1)$$

$$R^2 = \frac{[\sum_{i=1}^{n}(Pi-\bar{P})(Ai-\bar{A})]^2}{[\sum_{i=1}^{n}(Pi-\bar{P})^2(Ai-\bar{A})^2]} \quad (2)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Pi-Ai)^2}{n}} \quad (3)$$

The performance of our ANN was compared with simpler models such as linear regression and decision trees. Our ANN model showed 10% improvements in R^2 and a 15% drop in RMSE comparing with other models. These results show that the ANN model is an appropriate tool to model nonlinear relationships.

Table 5: Training hyperparameters and evaluation details

Hyperparameter	**Value**
Optimizer	Adamax
Learning Rate	0.002
Number of Epochs	100
Batch Size	32
Loss Function	Mean Squared Error (MSE)
Early Stopping Patience	10 epochs
Evaluation Method	10-fold cross-validation

3 Results

The dataset used in this study is presented in Table 6. The dataset includes inputs dataset (time in "sec" and speed in "km/h") and outputs which correspond to distance (in kilometers) and elevation (in meters). The total time and distance were used as key metrics to validate the dataset, which were essentially for training and testing the ANN model.

Statistical analysis of training and testing data for all inputs and outputs was performed and presented in Table 7. Then, the dataset was randomly split into two groups; the first group was 80:20, which 80% of the dataset was used for training and 20% of dataset was used for testing. As well as, in the second group of 70:30. Then, a normalization process was applied to the dataset prior training phase to ensure that the model can learn effectively and efficiently.

Table 6: Sample of the dataset used in the model

Time (sec)	Time (hour)	Total Time	Speed (Km/h)	Distance (Km)	Total distance (Km)	Elevation (m)
330	0.091667	1.558333	44	4.033333	66.138889	24
340	0.094444	1.652778	39	3.683333	69.822222	29
350	0.097222	1.750000	45	4.375000	74.197222	49
360	0.100000	1.850000	46	4.600000	78.797222	37
370	0.102778	1.952778	45	4.625000	83.422222	59

Table 7: Statistical analysis of training and testing data for all of the time, speed, distance, and elevation

	count	mean	Std	min	25%	50%	75%	max
Time(sec)	29.0	0.514	0.303	0.0	0.277	0.527	0.750	1.0
Speed (Km/h)	29.0	0.456	0.287	0.0	0.153	0.538	0.692	1.0
Distance (Km)	29.0	0.491	0.302	0.0	0.225	0.466	0.787	1.0
Elevation(m)	29.0	0.531	0.337	0.0	0.225	0.525	0.875	1.0

As presented in Figures 2 and 3, three different model configurations were tested by applying multi hidden layers such as one hidden layer, two hidden layers, and three hidden layers. As indicated, the results showed that the three-hidden layers provide an optimal performance for distance compared to the other hidden layers. The results achieving an 99.45% of R^2, 0.175% of MAE, and 0.1587% of RMSE. The results also showed a significant preference for elevation at three-hidden layer with 53.93% of R^2, 15.75 % of MAE, and 15.2% of RMSE. These results confirm that the ANN can effectively detect relevant patterns between data, which plays a key role in ensuring accurate UAV prediction trajectory.

A summary of statistical metrics includes MAE, R^2, and RMSE values are presented in Table 8 for each hidden layer. This table shows a clear comparison in each hidden layer and its performance for both distance and Elevation. With an R^2 of 99.45% and an RMSE of 0.1587 in the 80%-20% split, the three-layer model proved its accuracy and performing equally well in the 70%-30% split.

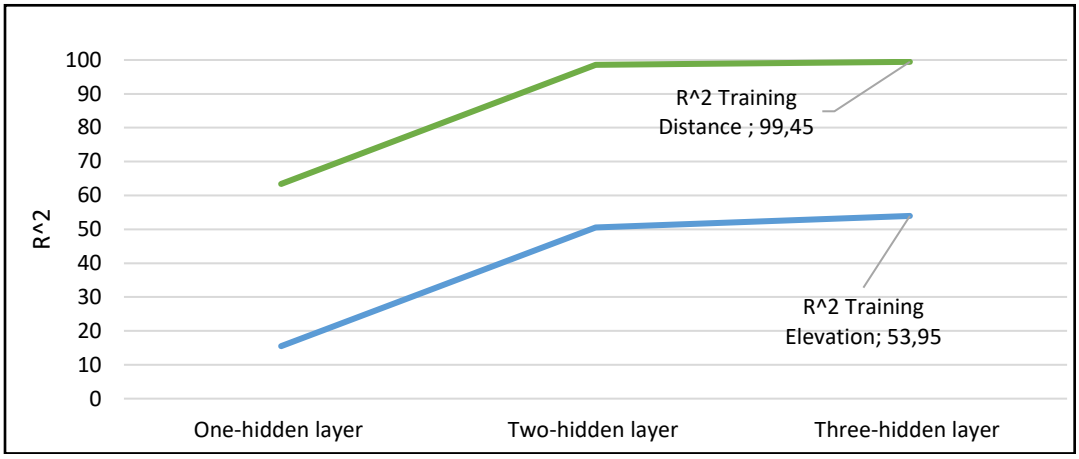


Figure 2: R^2 at different hidden layers

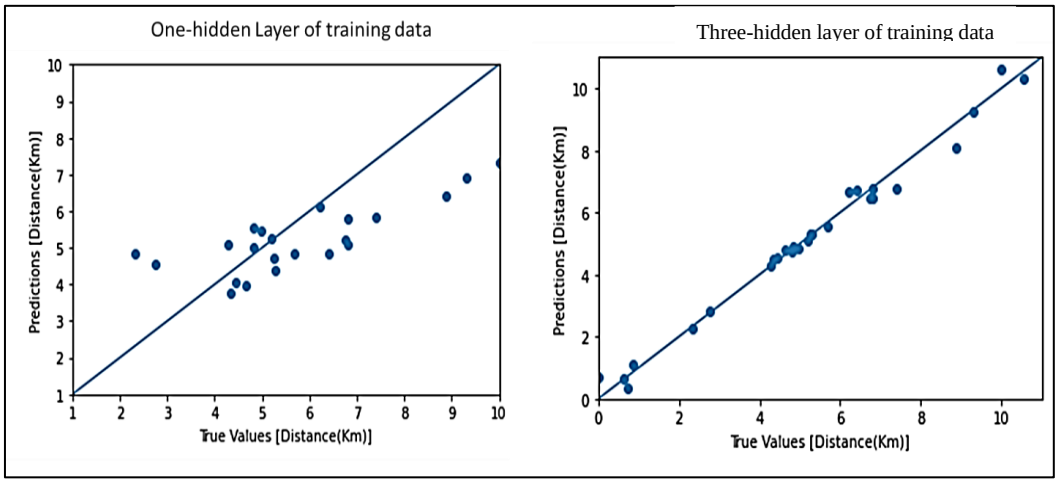


Figure 3: The real versus the predicted values of the ANN model's training and testing phases in the different hidden layers

Table 8: MAE, R^2, and RMSE values for training and testing data for all the three hidden layers

Small Data set (39 random data)		80%: 20%		70%: 30%	
		Training		Training	
		R^2	RMSE	R^2	RMSE
One-hidden layer	Distance (Km)	63.4	1.28	62.91	1.33
	Elevation (m)	15.5	24.63	14.87	24.75
Two-hidden layer	Distance (Km)	98.6	0.216	98.59	0.2309
	Elevation (m)	50.55	15.07	45.01	16.02
Three-hidden layer	Distance (Km)	99.45	0.1587	99.22	0.1837
	Elevation (m)	53.95	15.2	50.30	16.02

4 Discussion

The three-hidden-layer ANN model demonstrated strong effectiveness in predicting UAV trajectories, especially when it came to distance. It achieved an R^2 of 99.45% and an MAE of just 0.1587, which is better than what traditional methods like regression and heuristic models can offer. Those older methods often get stuck in local optima and need a lot more computational resource. This shows how well ANNs can handle complex, nonlinear patterns in the data.

On the flip side, the model didn't perform as well with elevation predictions, showing an R^2 of only 53.95% and an MAE of 15.2. This suggests that there are challenges in predicting vertical movements accurately, possibly because the dataset lacks enough variability or because modeling altitude is inherently tricky. Other studies that have used reinforcement learning (RL) also faced similar challenges, but RL generally requires more computing power and longer training times, which makes it less practical for real-time UAV applications.

Our ANN model achieved a better balance between computational efficiency and accuracy when compared to other models such as Genetic Algorithms (GA) or Particle Swarm Optimization (PSO). Especially when Adamaz optimizer was utilized with the three hidden layers, high performance was enhanced in distance predictions.

This work stands out for its ability to offer very precise trajectory predictions while preserving computing efficiency, making it suited for real-time UAV operations. Looking ahead, further research should work on improving elevation forecasts, either by including more environmental factors or merging ANNs with other machine learning methods.

5 Conclusion

This work examined the possibility of ANNs in predicting and optimizing the UAV flight trajectory. A random and simulated dataset was generated to optimize and evaluate different flight scenarios. The model was validated to evaluate the performance of predictive model. The results showed that the three-hidden layers of ANN consistently outperformed other hidden layers. It achieved an R^2 of 99.45% and an MAE of just 0.1587. The relationship between the training and testing data was crucial for improving accuracy in predicting UAV trajectories.

The results of this study showed that the ANN model is highly effective for predicting and optimizing UAV flight paths, especially in the distance. These results make the UAVs safer and cost-efficient. On the flip side, the model didn't perform as well with elevation predictions, showing an R^2 of only 53.95% and an MAE of 15.2. This suggests that there are challenges in predicting vertical movements accurately, possibly because the dataset lacks enough variability or because modeling altitude is inherently tricky.

6 Future work and limitations

As mentioned earlier, this study aims to evaluate the performance of ANN in predicting the UAV flight trajectory. The results indicated the model performed well in the simulated data, which should involve testing it with real data to validate the model’s accuracy and reliability in diverse environments.

This study helps the UAVs that need to navigate around unexpected obstacles, need to adjust their trajectory quickly, or that need to change to new flight regulations. This model with its flexibility makes it quite fit for uses in logistics, surveillance, and any sector where UAVs must react dynamically. Also, using this model and integrating it with onboard systems, UAVs' capacity to manage challenging missions might be improved.

References

[1] Dou, X., Yang, Y.: Comprehensive Evaluation of Machine Learning Techniques for Estimating the Responses of Carbon Fluxes to Climatic Forces in Different Terrestrial Ecosystems. Atmosphere (Basel). 9, 83 (2018). https://doi.org/10.3390/atmos9030083

[2] McCoy, J.T., Auret, L.: Machine learning applications in minerals processing: A review, (2019)

[3] Mathworks: Introducing Machine Learning What is Machine. Perspectives on Ontology Learning. (2016)

[4] Ali, D., Hayat, M.B., Alagha, L., Molatlhegi, O.K.: An evaluation of machine learning and artificial intelligence models for predicting the flotation behavior of fine high-ash coal. Advanced Powder Technology. 29, 3493–3506 (2018). https://doi.org/10.1016/j.apt.2018.09.032

[5] Alsafasfeh, A., Alagha, L., Alzidaneen, A., Nadendla, V.S.S.: Optimization of flotation efficiency of phosphate minerals in mine tailings using polymeric depressants: Experiments and machine learning. Physicochemical Problems of Mineral Processing. 58, 150477 (2022). https://doi.org/10.37190/PPMP/150477

[6] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., Research, G.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

[7] Gams, M., Kolenik, T.: Relations between Electronics, Artificial Intelligence and Information Society through Information Society Rules. Electronics 2021, Vol. 10, Page 514. 10, 514 (2021). https://doi.org/10.3390/ELECTRONICS10040514

[8] Lu, X., Yang, Z., Yang, Y., Sharma, A., Veselov, G., Tselykh, A., Sharma, A., Huang, R.: Applications of Artificial Intelligence in Evolution of Smart Cities and Societies. Informatica. 45, 603 (2021). https://doi.org/10.31449/INF.V45I5.3600

[9] Mishra, R.K., Patnaik, A.: Neural network-based CAD model for the design of square-patch antennas. IEEE Trans Antennas Propag. 46, 1890–1891 (1998). https://doi.org/10.1109/8.743842

[10] Mohanty, S.: Artificial neural network-based system identification and model predictive control of a flotation column. J Process Control. 19, 991–999 (2009). https://doi.org/10.1016/j.jprocont.2009.01.001

[11] Chen, M., Challita, U., Saad, W., Tutorials, C.Y.-… S.&, 2019, undefined: Artificial neural networks-based machine learning for wireless networks: A tutorial. ieeexplore.ieee.org.

[12] Ramadhas, A.S., Jayaraj, S., Muraleedharan, C., Padmakumari, K.: Artificial neural networks used for the prediction of the cetane number of biodiesels. Renew Energy. 31, 2524–2533 (2006). https://doi.org/10.1016/j.renene.2006.01.009

[13] Benson, M., Letters, R.C.-E., 1997, undefined: Recurrent neural network array for CDMA mobile communication systems. ieeexplore.ieee.org.

[14] Haykin, S., Nie, J., Letters, B.C.-E., 1999, undefined: Neural network-based receiver for wireless communications. IET.

[15] Bi, S., Zhang, R., Ding, Z., magazine, S.C.-I. communications, 2015, undefined: Wireless communications in the era of big data. ieeexplore.ieee.org.

[16] Wu, Q., Zeng, Y., Wireless, R.Z.-I.T. on, 2018, undefined: Joint trajectory and communication design for multi-UAV enabled wireless networks. ieeexplore.ieee.org.

[17] Zhang, G., Wu, Q., Cui, M., Zhang, R.: Securing UAV Communications Via Trajectory Optimization. (2017)

[18] Zeng, Y., Zhang, R., Magazine, T.L.-I.C., 2016, undefined: Wireless communications with unmanned aerial vehicles: Opportunities and challenges. ieeexplore.ieee.org.

[19] Djezzar, N., Fernández Pérez, I., Djedi, N., Duthen, Y.: A Computational Multiagent Model of Bioluminescent Bacteria for the Emergence of Self-Sustainable and Self-Maintaining Artificial Wireless Networks a Computational Multi-agent Model of Bioluminescent Bacteria for the Emergence of Self-Sustainable and Self-Maintaining Artificial Wireless Networks. Informatica. 43, (2019). https://doi.org/10.31449/inf.v43i3.2381ï

[20] Bukar, U.A., Sayeed, M.S., Razak, S.F.A., Yogarayan, S., Amodu, O.A.: An exploratory bibliometric analysis of the literature on the age of information-aware unmanned aerial vehicles aided communication. Informatica. 47, 91–114 (2023). https://doi.org/10.31449/INF.V47I7.4783

[21] Chen, M., Mozaffari, M., Saad, W., … C.Y.-I.J. on, 2017, undefined: Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience. ieeexplore.ieee.org.

[22] Jha, K.K., Jha, R., Jha, A.K., Hassan, M.A.M., Yadav, S.K., Mahesh, T.: A Brief Comparison on Machine Learning Algorithms Based on Various Applications: A Comprehensive Survey. CSITSS 2021 - 2021 5th International Conference on Computational Systems and Information Technology for Sustainable Solutions, Proceedings. (2021). https://doi.org/10.1109/CSITSS54238.2021.9683524

[23] Al-Azzam, N., Shatnawi, I.: Comparing supervised and semi-supervised Machine Learning Models on Diagnosing Breast Cancer. Annals of Medicine and Surgery. 62, 53–64 (2021). https://doi.org/10.1016/J.AMSU.2020.12.043

[24] Morales, E.F., Escalante, H.J.: A brief introduction to supervised, unsupervised, and reinforcement learning. Biosignal Processing and Classification Using Computational Learning and Intelligence: Principles, Algorithms, and Applications. 111–129 (2022). https://doi.org/10.1016/B978-0-12-820125-1.00017-8

[25] Naeem, S., Ali, A., Ahmed, M., Anam, S., Ahmed, M.M.: An Unsupervised Machine Learning Algorithms: Comprehensive Review Article in International Journal of Computing and Digital Systems. International Journal of Computing and Digital Systems. 13, 2210–142 (2023). https://doi.org/10.12785/ijcds/130172

[26] Al-Fuqaha, A., Guizani, M., tutorials, M.M.-… surveys &, 2015, undefined: Internet of things: A survey on enabling technologies, protocols, and applications. ieeexplore.ieee.org.

[27] Ni, R., Schneider, T., Panozzo, D., Pan, Z., Gao, X.: Robust & Asymptotically Locally Optimal UAV-Trajectory Generation Based on Spline Subdivision. Proc IEEE Int Conf Robot Autom. 2021-May, 7715–7721 (2021). https://doi.org/10.1109/ICRA48506.2021.9561272

[28] Patnaik, A., Anagnostou, D.E., Mishra, R.K., Christodoulou, C.G., Lyke, J.C.: Applications of neural networks in wireless communications. IEEE Antennas Propag Mag. 46, 130–137 (2004). https://doi.org/10.1109/MAP.2004.1374125

[29] Katal, A., Singh, N.: Artificial Neural Network: Models, Applications, and Challenges. EAI/Springer Innovations in Communication and Computing. 235–257 (2022). https://doi.org/10.1007/978-3-030-78284-9_11

[30] Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A.E., Arshad, H.: State-of-the-art in artificial neural network applications: A survey. Heliyon. 4, e00938 (2018). https://doi.org/10.1016/J.HELIYON.2018.E00938/ASSET/85B25AED-7CEE-48F9-8F25-73F74B8D991E/MAIN.ASSETS/GR6.JPG

[31] Venkateswara Rao, D.M.K.K., Habibi, H., Sanchez-Lopez, J.L., Voos, H.: An Integrated Real-time UAV Trajectory Optimization with Potential Field Approach for Dynamic Collision Avoidance. 2023 International Conference on Unmanned Aircraft Systems, ICUAS 2023. 79–86 (2023). https://doi.org/10.1109/ICUAS57906.2023.10156337

[32] Yang, Y., Xiong, X., Yan, Y.: UAV Formation Trajectory Planning Algorithms: A Review. Drones 2023, Vol. 7, Page 62. 7, 62 (2023). https://doi.org/10.3390/DRONES7010062

[33] Atayev, A., Fliege, J., Zemkoho, A.: Trajectory optimization of unmanned aerial vehicles in the electromagnetic environment. Optimization and Engineering. 1–40 (2024). https://doi.org/10.1007/S11081-024-09893-5/FIGURES/11

[34] Arif, S., Khan, M.A., Rehman, S.U.: RSSI Estimation for Constrained Indoor Wireless Networks using ANN. (2024)

Forecasting Solar Energy Generation Using Machine Learning Techniques and Hybrid Models Optimized by War SO

Fenghong Pan
School of Electric Power Engineering, Fujian Polytechnic of Water Conservancy and Electric Power, Sanming 366000, Fujian, China
E-mail: 13944238815@163.com

Keywords: solar energy, renewable energy, machine learning, Cat Boost, AdaBoost, Light GBM, War SO

Received: November 19, 2024

Due to threats caused by climate change and energy security, the attainment of adequate and sustainable energy resources is becoming of great importance. There exist promising alternatives to the traditional source, such as solar and wind. However, there are high obstacles to their penetration into a power grid because of the variability and uncertainty in renewable sources. In this regard, it becomes quite necessary to accurately forecast the models so that one can optimize energy generation and guarantee grid stability. This work studies the application of several machine learning algorithms, including Cat Boost, AdaBoost, and Light GBM, to solar energy generation forecasting. The approach has been applied based on data from two solar stations over a period of two years, where the performance of each stand-alone algorithm and a hybrid model that will be optimized with War SO optimizer is analyzed and presented. The standalone CatBoost model demonstrated superior performance, achieving an R^2 of 0.9106 and RMSE of 4.06 MW in the 30 MW farm. Hybrid models further improved accuracy, with the AdaBoost-War SO model reaching an R^2 of 0.9836 and RMSE of 1.75 MW. These results confirm the efficiency of utilizing machine learning approaches toward enhancing accuracy in renewable energy forecasting, and therefore hybrid models play an important role in energy prediction with higher accuracy.

Povzetek: Raziskava uvaja hibridne modele strojnega učenja, optimizirane z algoritmom War SO, ki kvalitetno napovedujejo proizvodnjo sončne energije.

1 Introduction

The global community is confronted with several issues concerning the sustainability and security of energy. Failure to address these challenges promptly could result in economic and political turmoil. Depletion of fossil fuel reserves and the environmental consequences of their combustion have sparked greater attention towards the exploration of alternative, sustainable energy sources. Renewable energy technologies such as solar, wind, hydropower, geothermal, and biomass have experienced substantial growth during the last few years, reflective of increasing use in international energy markets [1]. Machine learning techniques have indicated promise in addressing the variability of renewable energy sources such as solar and wind [2].

Research and development in renewable energy have attracted considerable attention lately because of the increasing need for clean and sustainable energy sources [3][4]. Renewable energy therefore comes to the front in efforts to counter greenhouse gas emissions and change due to climatic factors [5–7]. RES offers an assortment of advantages that include a reduction in reliance on foreign sources of energy, jobs, and the possibility of saving money economically [8]. However, the intrinsic variability and unpredictability associated with RES have been a significant obstacle to their wide diffusion [9][10]. For instance, solar energy generation is still very sensitive to factors affecting cloud cover and the seasonal variation of sunlight intensity [11]. All these large variations and uncertainties in renewable energy generation make their smooth integration into the power grid challenging [12].

A necessary strategy to minimize this challenge emphasizes the development of accurate forecasting models in renewable energy generation. Such models are very important in minimizing the negative effects brought about by the variability and uncertainty of the electrical grid. Traditional energy generation forecasts have, for many years, employed techniques such as statistical and physical models [13]. While statistical approaches like the autoregressive integrated moving average model have shown some promise, they are limited in terms of modeling complex nonlinear relationships and high dimensionality inherent in renewable energy signals [14]. Physical models, such as NWP and solar radiation models, play a major role in renewable energy forecasting. However, physical models face serious problems in light of complex dynamics in the Earth's atmosphere and inherent uncertainties in weather prediction. Various research and development works need to be carried out to improve their accuracy. Machine learning algorithms open up a promising direction beyond the limitations of traditional methods that have been developed for forecasting renewable energies [15][16]. First, ML algorithms are excellent at finding complex nonlinear relationships that many big datasets exhibit. For this reason, ML is suitable to handle the multidimensional nature of renewable energy data.

Second, ML algorithms can be easily modified to fit different types of input data: time series, meteorological, and geographical.

This has motivated many researchers to work on the development of machine learning algorithms in predicting solar radiation, one of the critical factors in evaluating the performance of a solar energy system [17]. Voyant et al. have reviewed several approaches to solar irradiation forecasting based on machine learning methods quite extensively. Techniques such as neural networks, support vector regression, regression trees, random forests, and gradient boosting were reviewed. Comparing many different works was challenging because of the characteristics of the diverse nature of the dataset and besides that different metrics of performance were applied. They found very similar errors of prediction overall, from which it follows that there is a huge potential for improving accuracy if hybrid models or ensemble forecasting approaches are implemented [18]. Suanpang and Jamjuntr (2024) benchmarked the Light Gradient Boosting Machine (LGBM) and K Nearest Neighbors (KNN) models for solar power generation forecasting in microgrids. Their results indicated that LGBM performed better than KNN in terms of accuracy (R^2 = 0.84 vs. 0.77) and error values (RMSE: 5.77 vs. 6.93; MAE: 3.93 vs. 4.34), although it took more computational power, i.e., longer training time (120 s vs. 90 s) and higher memory (500 MB vs. 300 MB). LGBM was also more consistent over periods and seasons and dealt with outliers effectively. This paper emphasizes the significance of precise prediction in enhancing solar energy utilization in microgrids and elucidates the trade-offs between computational efficiency and prediction accuracy [19]. Singh et al. (2023) introduced a robust hybrid deep learning approach for power prediction using PV, wind, and solar systems in large-scale systems. It uses preprocessing methods and K-means clustering to enhance deep learning training and eliminate noise. A GRU-based recurrent neural network yielded more accuracy than conventional approaches. Pearson coefficient analyses identified interrelations among power sources, with which hybrid renewable clusters were able to minimize forecasting errors and variability. Case studies highlighted the controllability of solar power and the model's success in boosting forecasting for mass systems[20].

Nguyen et al. (2025) identified ambient temperature and humidity as key predictors using SHAP analysis [21], while Zhu et al. (2025) demonstrated the efficacy of hybrid optimization models like HGBoost with satin bowerbird optimizers [22].

Huertas-Tato et al. (2020) blended the forecasts of four models using Support Vector Machines. They evaluated two methods for combining the forecasts and the impact of considering weather-type information in the blends. Results from evaluation at four Iberian Peninsula stations showed large performance gains due to blending, with up to 17% reduction in RRMSE for GHI (16% for DNI), and up to 15% in rMAE. Improvement was similar when evaluating regional forecast skills [23]. Four models were used by Gürel et al. (2020) in modeling solar radiation for the years 2008-2018 in Turkey. The feed-forward neural network outperformed others, followed by Holt-Winters, RSM, and empirical models [24]. Alizamir et al. (2020) estimated the performance of six machine-learning models for solar radiation forecasting at selected stations in Turkey and the USA. These authors compared different models by applying several statistical indicators and pointed out that GBT outperformed the others. GBT decreased the average RMSE by 0.26% to 19.34% for one station and by 4% to 54.8% for the other one, indicating an effective use of climatic parameters for solar radiation prediction [25]. Koo et al. developed a new methodology for estimating the monthly average daily solar radiation in China using different machine-learning techniques. Their approach was to use clustering and enhanced case-based reasoning models, which have given an average prediction accuracy of 93.23% when applied to data from 97 cities over a continuous period of 10 years. This may thus provide a very effective way of implementing solar energy systems, enabling decision-makers to determine the best locations and configurations [26]. Nath et al. (2020) discussed two machine-learning techniques for hourly solar power forecasting. Their work was focused on how to enhance energy grid integration and service quality by optimizing data preprocessing, feature selection, weather profiling, and choosing the algorithms that provide better accuracy and efficiency in the forecast of solar power, thus helping to meet global energy demands [27]. Kumar et al. (2020) suggested a short-term solar energy forecast using PI-based machine learning. The authors support the fact that their approach makes the forecast more accurate and reliable than in the case of deterministic methods, which is urgent for grid stability and reliability, considering the stochastic nature of photovoltaic power generation [28]. Jebli et al. (2021) introduced a machine and deep learning-based solar energy forecasting approach critical to increasing competitiveness for solar power plants and reducing reliance on fossil fuels. The authors conducted their research on Errachidia, Morocco, for data from 2016 to 2018, using RF and ANN models, outperforming other methods such as LR and SVR. Comparisons with Pirapora, Brazil, enhanced the quality and reproducibility of this study [29]. In this regard, Abualigah et al. (2022) reviewed all kinds of learning-based modeling for renewable power source estimation by focusing on recent deep learning and machine learning algorithms. Then they discussed the performance analysis based on the new taxonomy, challenge, and possibility for the future research direction. Based on this, the paper has highlighted that hybrid learning techniques were effective in addressing energy generation problems and thus suggested using these techniques for improvement in forecasting accuracy [30].

Nevertheless, it is noted that other well-known algorithms, i.e., XGBoost and neural networks, are widely used in the renewable energy forecasting literature. The exclusion of XGBoost is mainly due to its similarity to LightGBM, which is more computationally efficient and tailor-made for big datasets. Neural

networks, including recurrent and convolutional architectures, offer significant advantages to their ability to capture temporal and spatial patterns; yet, they are computationally intensive and need big datasets. By the above study's emphasis on high-frequency but site-specific data, gradient-boosting models were preferred as they can balance accuracy, computational cost, and interpretability. Subsequent studies can be focused on using neural networks or hybrid models, for instance, the integration of gradient-boosting methods and deep learning, to leverage their respective strengths. Moreover, a finer comparative study between XGBoost and different neural network configurations can provide further clarity into their utility to solar power forecasting projects in comparable situations.

Notwithstanding improvements in solar forecasting, significant gaps exist in current methods. Most research uses low-resolution data sets, i.e., hourly or daily measurements, that are incapable of recording short-term variability important for real-time grid integration. This research bridges this gap by using high-frequency, 15-minute interval data from two solar farms to enable better modeling of dynamic conditions.

Feature selection in modern state-of-the-art (SOTA) techniques has the tendency to apply simple methods that do not consider non-linear variable interactions. The Delta Moment Independent Measure (DMIM) introduced in this paper is utilized in the identification of vital predictors like solar irradiance, and it offers improved input selection for prediction purposes. Additionally, although hybrid models have been promising, standard optimization techniques like grid search or genetic algorithms hinder their potential. The employment of the War SO optimizer in this research surpasses these constraints by improving model accuracy and computation time.

One of the most prominent limitations seen in existing work is the lack of multi-site validation, which casts doubt on results. The work demonstrates the generality of the models suggested by validation with data from two farms with capacities of 30 MW and 130 MW. Besides, the majority of SOTA work relies on a limited collection of performance measures such as RMSE or R^2. Nevertheless, this work applies a comprehensive evaluation framework including MAE, runtime, and convergence analysis to enable thorough inspection.

By bridging these gaps, this research establishes a new standard in solar forecasting, pushing the boundaries of high-resolution data use, robust feature engineering, hybrid optimization, and generalizable model development.

Despite the huge advancements in the prediction of renewable energy, there are several challenges to limit the scalability and viability of machine learning models in the same. Most notable are the data security and privacy concerns, since the collection of sensitive operational data from solar farms is usually an essential stepping stone for model development; however, sharing the same is fraught with risks and dissuades collaboration. Also, integrating advanced machine learning models into existing energy systems, particularly legacy-based ones, is extremely challenging and must be harmonized with existing solar forecasting methods to facilitate easy implementation. Hybrid models improve forecast accuracy but often come with maintenance and integration issues with operational systems, which could deter use. These research gaps are overcome by this research using high-frequency 15-minute interval data to raise the level of granularity and predictive accuracy of the models beyond what has been possible using hourly or daily data. The use of hybrid machine learning models that have been optimized with the War Strategy Optimizer generates superior predictive capacity and computational efficacy than standard procedures. Feature robustness through strong feature selection further secures model stability against overfitting, contributing to methodological robustness. By highlighting the scalability of the hybrid models, computational efficiency, and integrability feasibility, the research translates theoretical findings toward practical realization for solar energy prediction. Problems of data privacy, system compatibility, and real-time deployments, further improving scalable and actionable models, are to be addressed in follow-up studies. Table 1 indicates the comparing the results of the discussed studies.

Table 1: Comparison of the results of the discussed studies

Study	Models Used	Metrics	Key Contributions
Suanpang & Jamjuntr (2024)[19]	LGBM, KNN	R^2 = 0.84 (LGBM), RMSE = 5.77 W (LGBM)	Benchmarking LGBM and KNN for microgrid forecasting; LGBM showed superior accuracy.
Singh et al. (2023)[20]	GRU-based hybrid deep learning model	Improved accuracy over conventional models	Use of K-means preprocessing and GRU for large-scale systems.
Nguyen et al. (2025)[21]	CatBoost, SHAP Analysis	R^2 = 0.46, RMSE = 4.748 W (CatBoost)	Identified ambient temperature and humidity as key predictors.
Zhu et al. (2025)[22]	Hybrid models (HGBoost + optimizers)	R^2 = 0.9907	Hybrid optimization with satin bowerbird optimizer.
Huertas-Tato et al. (2020)[31]	Blending ML models	Up to 17% RRMSE reduction	Blended forecasts using SVM, leveraging weather-type information.
Alizamir et al. (2020)[32]	Gradient Boosting Trees (GBT)	RMSE reduction: 0.26%–19.34%	GBT demonstrated superior

2 Methodology

The research methodology is divided into two main sections. Firstly, the data acquisition process is outlined, detailing how the relevant data was collected and

sourced. Following this, the second part delves into the machine-learning algorithms utilized in the study, providing an overview of each algorithm and explaining the methods employed for their implementation in the research context. This study employs machine learning techniques to forecast energy generation from solar sources. Initially, the available data undergoes preprocessing using various methods. A crucial secondary analysis assesses the impact of input features on the output by examining correlations among parameters with the Pearson Correlation Coefficient. Following this, the dataset is split into training and testing subsets to enable accurate energy consumption prediction. Pearson Correlation Coefficient was selected as the feature selector because it is simple, interpretable, and computationally fast. The method is efficient at detecting linear associations between the input features and target variable and hence qualifies as an appropriate first-line approach to filtering out the relevant features in the structured numerical data set employed in this research. CatBoost, AdaBoost, and Light GBM algorithms are individually and collectively employed for training and prediction to improve model accuracy. The hyperparameters of these algorithms are optimized using the War SO optimizer. Performance comparison between single algorithms and hybrid models is conducted using various statistical indicators to identify the most effective approach for energy generation prediction. The focus of the present study is on CatBoost, LightGBM, and AdaBoost because of their proven performance and efficiency in renewable energy prediction tasks. The algorithms are all gradient-boosting methods with the ability to process structured datasets, prevent overfitting, and identify complex non-linear variable relationships. Specifically, CatBoost is effective in processing categorical features and removing prediction bias, while LightGBM and AdaBoost are known for scalability and iterative learning, respectively.

The entire modeling process is depicted in Fig 1.

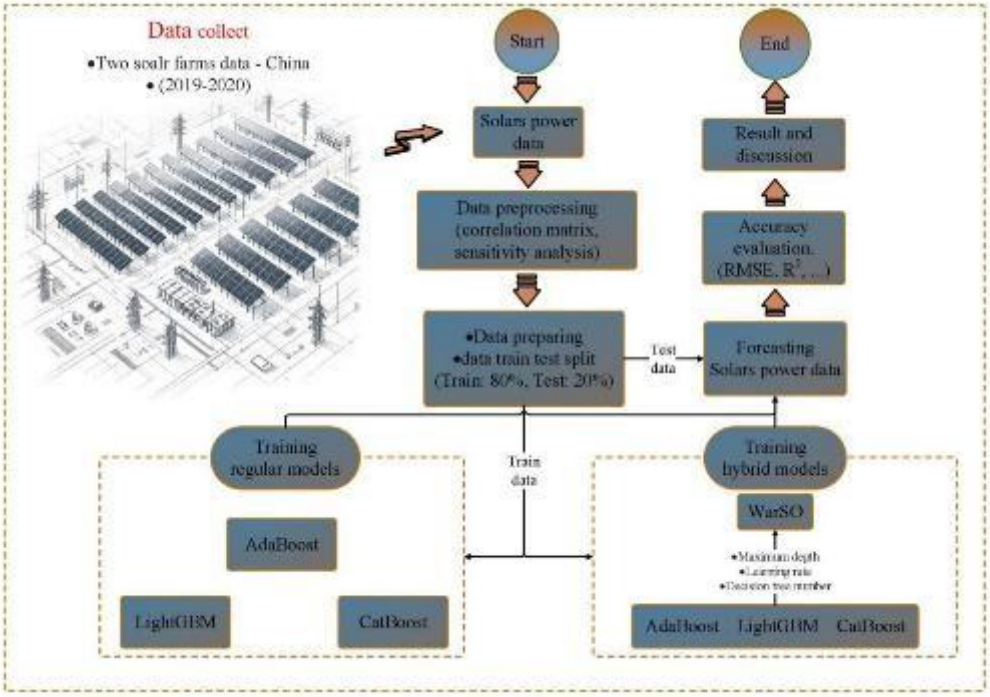


Figure 1: Flowchart diagram of the current investigation

1. Data

For this study, data collection involved the procurement of solar generation data from various on-site renewable energy stations situated across China. Specifically, information was gathered from two solar stations. Over a period spanning two years, from 2019 to 2020, data was meticulously recorded at 15-minute intervals. This dataset, comprising power generation data alongside weather-related parameters, was subsequently utilized in the Renewable Energy Generation Forecasting Competition hosted by the Chinese State Grid in 2021 [33]. Table 2 summarizes all data columns along with their respective descriptions.

Table 2: The input variables and their statistical details in the farm with a capacity of 30(MW)

	count	mean	std	min	25 %	50%	75%	max
Year	20352	2019	0	2019	2019	2019	2019	2019
Day	20352	15.66037736	8.767529044	1	8	16	23	31
Month	20352	4.018867925	2.00467189	1	2	4	6	7
Hour	20352	11.5	6.92235662	0	5.75	11.5	17.25	23
Minute	20352	22.5	16.77092186	0	11.25	22.5	33.75	45
Total solar irradiance (W/m2)	20352	198.8134336	294.5791441	0	0	0	338.25	1117
Direct normal irradiance (W/m2)	20352	100.7295597	185.090418	0	0	0	112	760
Global horizontal irradiance (W/m2)	20352	69.30508058	101.8772566	0	0	0	111	656
Atmosphere (hpa)	20352	1016.013768	9.323415494	994.8	1008.3	1014.7	1024	1038.6
Relative humidity (%)	20352	58.24924332	13.15880075	14.1	50.9	61	68.6	80.5
Power (MW)	20352	5.449246788	8.258662461	0	0	0.115101	9.038321 25	29.91133 95

Table 3: The input variables and their statistical details in the farm with a capacity of 130(MW)

	count	mean	std	min	25%	50%	75%	max
Year	70176	2019.500684	0.500003095	2019	2019	2020	2020	2020
Day	70176	15.73871409	8.803983506	1	8	16	23	31
Month	70176	6.519835841	3.449575468	1	4	7	10	12
Hour	70176	11.5	6.922235873	0	5.75	11.5	17.25	23
Minute	70176	22.5	16.77062932	0	11.25	22.5	33.75	45
Total solar irradiance (W/m2)	70176	169.3033665	248.0776381	0	0	0	305.7575	1041.93
Direct normal irradiance (W/m2)	70176	122.1523955	178.9880244	0	0	0	220.6025	751.75
Global horizontal irradiance (W/m2)	70176	78.29928152	117.5873435	0	0	0	129.57	561.8
Air temperature (°C)	70176	13.69510759	12.03580036	-13.92	3.19	15.46	23.57	40.47
Atmosphere (hpa)	70176	861.0362624	6.147644763	844.51	856.2175	860.87	865.35	881.67
Power (MW)	70176	19.56748845	27.939605	0	0.241033	0.3269	36.8217485	109.3603

The difference in parameter scales shown in Tables 2 and 3 reflects the importance of location factors in determining solar energy output. In particular, the mean total solar irradiance measured for the 130 MW farm is higher than that of the 30 MW farm due to its larger geographic area and changing environmental conditions. These variations were adjusted for while training the models by normalizing the datasets individually for each farm, such that the models could learn to adapt to site-specific trends.

The 15-minute, high-resolution datasets also provided valuable detail for short-duration solar irradiance changes and other variables. The time resolution of the data enabled the models to detect rapid weather changes, increasing the accuracy of energy prediction. We appreciate that summary statistics in Tables 2 and 3 are

unable to capture the full richness of temporal data variation. Graphical representations, such as time series plots, would be an asset in future research to better present the dynamics measured at this scale level.

2.2 Machine learning methods

This study employed advanced machine learning algorithms, including Cat Boost, AdaBoost, and Light GBM, for energy generation forecasting[34]. To improve accuracy and adaptability, a hybrid model was developed by incorporating War SO optimizers. This section provides a concise summary of the mathematical formulations and fundamental principles underlying each of these techniques.

2.2.1 Categorical gradient boosting (cat boost)

The Cat Boost [35] model is a boosting-based algorithm that constructs trees in a level-wise manner. While the overall boosting process resembles existing methods, there are notable distinctions. Instead of performing residual calculations on all training data collectively, Cat Boost selects a subset of the data for residual calculations to build a model. Subsequently, it utilizes the predicted values from this model to process the residual of subsequent data. Moreover, Cat Boost employs Random Permutation to randomly select data, thereby promoting diversity in tree creation and preventing Overfitting [36].

$$\hat{x}_k^i = \frac{\sum_{j=1}^{n}\left[x_j^i = x_k^j\right] y_j + \alpha p}{\sum_{j=1}^{n}\left[x_j^i = x_k^j\right] + \alpha} \quad (1)$$

Here, α represents the corresponding weight, P denotes a prior value, xk = (x_k^1, ..., x_k^m) signifies the random vector of m features and yk INR den.

2.2.2 Adaptive boosting (AdaBoost)

Initially designed as a feature classification algorithm in machine learning, AdaBoost has expanded its application to regression problems [37]. Currently, it finds widespread use in load forecasting and short-term wind speed forecasting, yielding promising results. The core concept involves training multiple weak learners within the same sample space and subsequently adjusting their weights to construct a robust learner based on the prediction outcomes of each weak learner [38].

The specific steps of the AdaBoost algorithm are delineated as follows:

1. Selection of Basic Learner and Data: Initially, the weak learning algorithm and sample space (xi, yi) are determined. The sample data is denoted as group M, and the sample data is normalized with a mean of 0 and a variance of 1, where xi ∈ Rn and yi ∈ Rn.

2. Network Initialization: Assuming a uniform sample distribution, the weight of the test data's uniform distribution, Dt(i), is set to 1/M. The neural network structure is configured based on the characteristics of the sample data, followed by the initialization of weights and thresholds for the neural network. Finally, the number of iterations is set.

(3) Weak Predictor Prediction: The t-th weak predictor undergoes training using the training data, resulting in the prediction output for the training data. Following this, the error ei and average error et of the weak learner are computed for each sample using the calculation formula.

$$e_t = \frac{1}{M}\sum_{i=1}^{M} e_i, i = 1,2,\ldots,M \quad (2)$$

(4) Computing the Weight of Weak Learner: Based on the average error et of the prediction sequence f(t), the weight of the weak learner is determined accordingly.

$$a_t = \frac{1}{2} ln\left(\frac{1-e_t}{e_t}\right) \quad (3)$$

(5) Updating Sample Weights: Adjusting the weights of the next round of training samples is based on the current weight a_t. The formula for updating sample weights can be expressed as:

$$D_t(i) = \frac{D_{t-1}(i)}{B_t} * exp[-a_t y_i f_t(x_i)] \quad (4)$$

Here, Bt represents the normalization factor, which ensures that the sum of distribution weights equals 1 while maintaining the weight proportion unchanged. ft(xi) refers to a weak predictor acquired after training the data.

(6) Strong Predictor Formation: Following t rounds of training, t sets of weak predictor functions are acquired. Subsequently, strong predictors are constructed by amalgamating these t sets of weak predictor functions, as expressed below:

$$F(x) = \sum_{t=1}^{T} a_t . f_t(x) \quad (5)$$

In this context, T symbolizes the total count of weak learners.

2.2.3 Light gradient boosting machine (Light GBM)

Light GBM [39] stands out as a boosting-based algorithm recognized for its speed and precision in forecasting, surpassing other boosting and bagging algorithms. It leverages a gradient-boosting decision tree (GBDT) framework, incorporating gradient-based one-sided sampling and exclusive feature-bundling techniques. Unlike traditional gradient boosting machine (GBM) tree splitting methods, Light GBM adopts a leaf-wise approach, which enhances accuracy through more intricate modeling, particularly advantageous for time series forecasting. This method, combined with gradient boosting decision tree (GBDT) and leaf techniques, leads to low memory usage and rapid training. Light GBM encompasses several hyperparameters, with learning rate, number of iterations, and number of leaves being crucial for forecasting accuracy. Additionally, Light GBM addresses overfitting by adjusting Col sample by tree and

subsample hyperparameters. Proven effective in various time series forecasting domains such as electricity load and solar power forecasting, Light GBM's single-output forecasting demonstrates both rapidity and precision. Given the need for a fast and precise forecasting model with a single output, Light GBM is chosen for construction.

2.2.4 War strategy optimizer (War SO)

In the strategy of warfare, there are three primary factions: the King (K), the Commander (C), and the soldiers. Both the Commander and the King serve as leaders on the battlefield, overseeing the actions of the soldiers. Each soldier has an equal chance of rising to the ranks of Commander or King based on their combat effectiveness, which is measured by a cost function. However, there is a possibility that the Commander or King may face tough opposition from rival soldiers, representing a local optimum. These adversarial soldiers wield sufficient power to potentially ensnare the leaders. To avert such scenarios, the soldiers are managed in their coordinated tactics and maneuvers, guided by the status of the Commander or King [40]

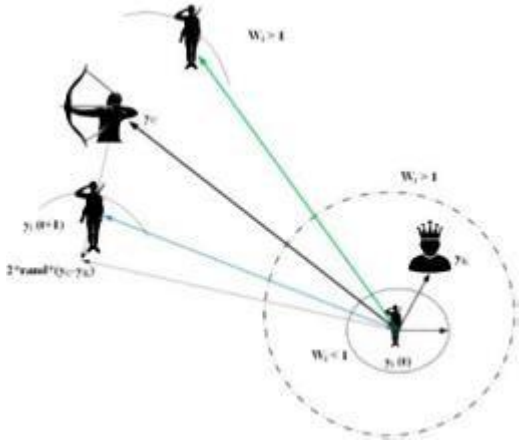


Figure 2: Renewing the attack model mechanism [41]

2.2.4.1 Attack tactic

Attack tactics are crucial components of war strategies, and this paper models two distinct policies. In the first strategy, each soldier updates their own status based on the current situation of the Commander and King. Figure (2) illustrates the procedure for updating the attack model. A favorable circumstance triggers the King to initiate a significant attack, with the soldier possessing the highest attack force or cost being appointed as the King. Initially, at the commencement of the war, all soldiers are endowed with equal weight and rank. However, their rank escalates as they effectively execute tactics. It's noteworthy that soldiers' weight and rank may be adjusted based on the success of tactics during the war's progression. As the war nears its end, the circumstances of the soldiers, Commander, and King converge as they move towards achieving the goal outlined in equation (6).

$$y_i(t+1) = y_i(t) + 2p(y_c - y_k) + rand(y_k * w_i - y_i(t)) \quad (6)$$

In this context, yi(t+1) and yi(t) denote the current and preceding statuses of the soldier, respectively. yK and yC denote the situations of the King and the Commander, while Wi represents the weight.

2.2.4.2 *Renewing weight and rank*

The renewal of each individual's situation is correlated with the status of the King, the location of the Commander, and the ranking of the soldiers. Soldiers' rankings are determined by their past performance in the war, which in turn influences the Wi factor. The ranking of each soldier signifies their proximity to achieving the goal (cost value). If the attack force (cost) in the previous situation (Fper) is significantly higher than that in the new situation (Fnew), the soldier opts to retain the previous situation, as depicted in equation (7).

$$y_i(t+1) = y_i(t) \times (F_{new} < F_{per}) + y_i(t+1) \times (F_{new} \geq F_{per}) \quad (7)$$

If soldiers successfully renew their situation, their ranking (Rai) will be upgraded, as shown in equation (8). Using this ranking, the updated weighting can be computed as described in Equation 9.

$$Ra_i = Ra_i \times (F_{new} < F_{per}) + (Ra_i + 1) \times (F_{new} \geq F_{per}) \quad (8)$$

$$w_i = w_i \times (1 - \frac{Ra_i}{Max_{iter}})^{\beta} \quad (9)$$

2.2.4.3 Defense strategy

Another approach to updating the situation involves the King, a randomly selected soldier, and the Commander's status. However, the adjustment of weight and ranking remains consistent, as illustrated in equation (7).

$$y_i(t+1) = y_i(t) + 2p(y_k - y_{rand}(t)) + rand * w_i * (y_c - y_i(t)) \quad (10)$$

Unlike the prior policy, this military strategy ventures into broader territories when incorporating the status of the randomly selected soldier. Soldiers make

substantial strides in updating their situation when larger Wi values are present. Conversely, when Wi amounts are small, the opposite occurs.

2.2.4.4 Substituting the vulnerable soldier

Throughout the duration of the conflict, the weakest soldier, distinguished by the lowest value of the cost function, is singled out for replacement. This study investigates multiple strategies for substitution in such instances. The simplest approach involves replacing the weak soldier with a randomly chosen one, as determined by the formula below [equation (11)]:

$$y_w(t+1) = L_L + rand \times W_i \times (H_L - L_L) \tag{11}$$

The second approach involves substituting the weakest soldier with one in close proximity to the average of the entire army in the field, as represented by the following formula. This tactic is aimed at enhancing the convergence of the optimizer [equation (12)]:

$$y_w(t+1) = y_k - (1 - rand) \times (y_w(t) - median(y)) \tag{12}$$

2.2.4.5 Key features of the provided optimizer

The proposed optimizer possesses several important features that enhance the optimization process. Firstly, it achieves a satisfactory balance between the exploitation and exploration phases. Each individual (soldier) in this optimizer is assigned a unique weight based on their ranking. Moreover, weight adjustment only takes place if there is an enhancement in the individual's cost value during the updating phase, and this adjustment is tied to the particle's position in relation to the positions of the Commander and King. The fluctuation in weights follows a nonlinear pattern, with substantial alterations happening in the initial epochs and diminishing ones towards the conclusion, aiding in quicker convergence to the global optimum. Moreover, the situation updating involves two steps, enhancing exploration capabilities towards the global optimum. This optimizer is recognized for its simplicity, requiring fewer computations.

2.2.4.6 The stages of exploitation and exploration

The concepts of exploitation and exploration are fundamental principles in metaheuristic optimizers and are crucial for their effectiveness. The proposed optimizer maintains a balanced trade-off between these two phases. The attack tactic is representational of the exploitation side, whereby the optimizer leverages known solutions to its advantage in furthering optimization performance. Conversely, the defense tactic symbolizes exploration in allowing the optimizer to move toward newer areas of the search space and hopefully come up with far better solutions than previously obtained. This balanced approach ensures efficient optimization by leveraging both exploitation and exploration strategies.

2.2.5 Model verification and evaluation

In the present study, the effectiveness of the forecasting model is tested with several error analysis measures. These include RMSE, MAE, RAE, JSD, VAF, and R-squared. These measures test the accuracy of the model and differences in values forecasted with the model against real ones. The comprehensive evaluation will give full insight into the performance of the model and indicate if some improvement might be necessary [42]. Detailed mathematical expressions for these statistical evaluation metrics are provided in Table 4.

Table 4: Statistical evaluation indexes

Statistics	Criteria	Equation
RMSE	Root Mean Squared Error	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{T}}$
MAPE	Mean Absolute Error	$\frac{\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert}{n}$
VAF	Variance Accounted For	$100\% \times \frac{\sum_{i=1}^{n}(y_i - \bar{y})(f_i - \bar{f})}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
JSD	Jensen Shannon Divergence	$\frac{1}{2}D(P \parallel M) + \frac{1}{2}D(Q \parallel M)^*$
R2	Coefficient of Determination	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
RAE	Relative Absolute Error	$\frac{[\sum_{i=1}^{n}(\hat{y}_i - y_i)^2]^{\frac{1}{2}}}{[\sum_{i=1}^{n}(y_i)^2]^{\frac{1}{2}}}$

*For more details, refer to Nielsen (2021)[43].

3 Results

This section outlines the results and analyses derived from the energy generation forecasting process. It begins with an introduction to the standalone algorithms CatBoost, LightGBM, and AdaBoost, followed by their hybrid configurations fine-tuned using the WarSO optimizer. A comprehensive array of charts and tables is provided to facilitate the assessment of the models.

Fig 3 depicts the correlation matrix created for the selected parameters in energy generation using solar energy at the first site considered with a capacity of 30 megawatts. Examination of the correlation matrix (depicted in Fig 3) indicates that total solar irradiance, direct normal irradiance, and global horizontal irradiance collectively play a substantial role. Notably, the parameter "global horizontal irradiance" exhibits the strongest correlation with the target parameter. Temperature variables display positive effects and correlations, while the impact of other parameters on the target parameter is minimal.

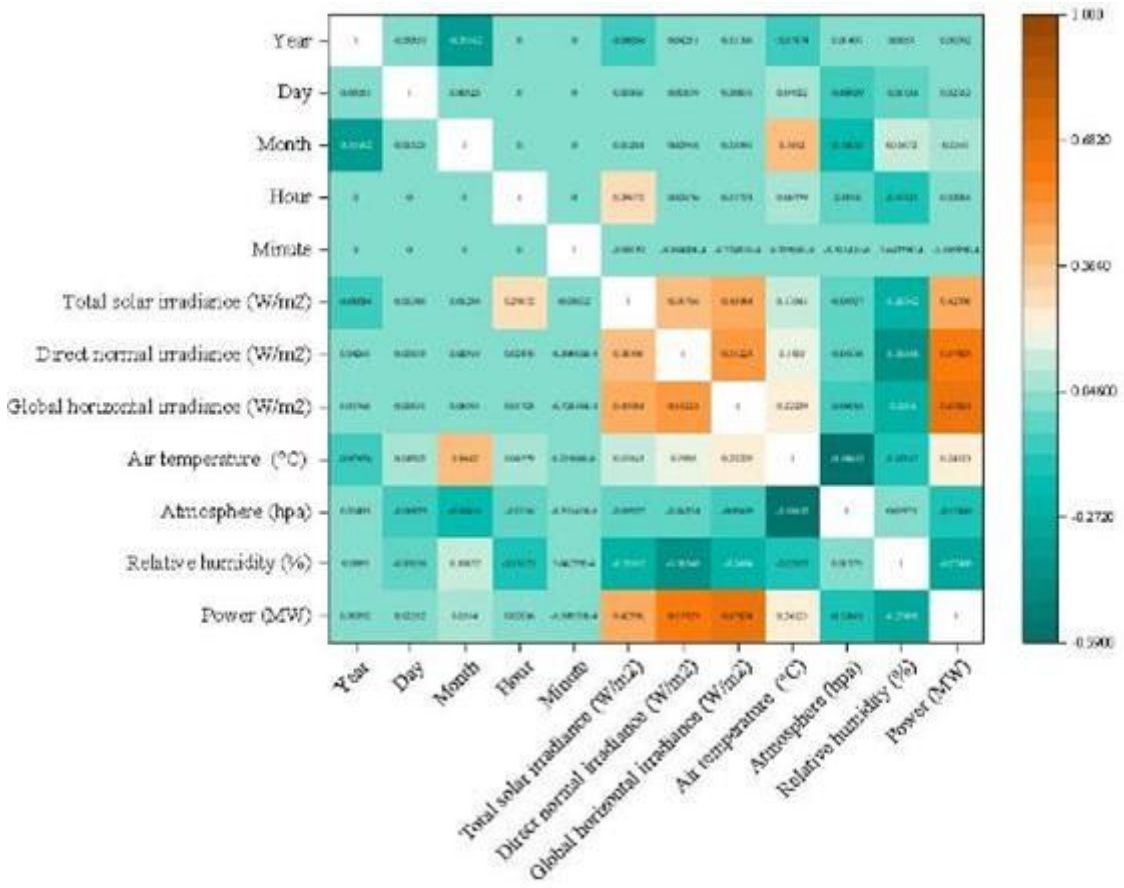


Figure 3: The correlation matrix of features in the farm with a capacity of 30(MW)

Fig 4 illustrates the correlation matrix generated for the selected parameters in solar energy generation at the second site under consideration, which has a capacity of 130 megawatts. Similar to the 30-megawatt case, three parameters, namely total solar irradiance, direct normal irradiance, and global horizontal irradiance, exhibited a very strong correlation with the target parameter. Among these, the total solar irradiance parameter demonstrated the highest correlation. Additionally, temperature and hour parameters showed positive correlations, while the remaining parameters exhibited negligible and almost neutral correlations.

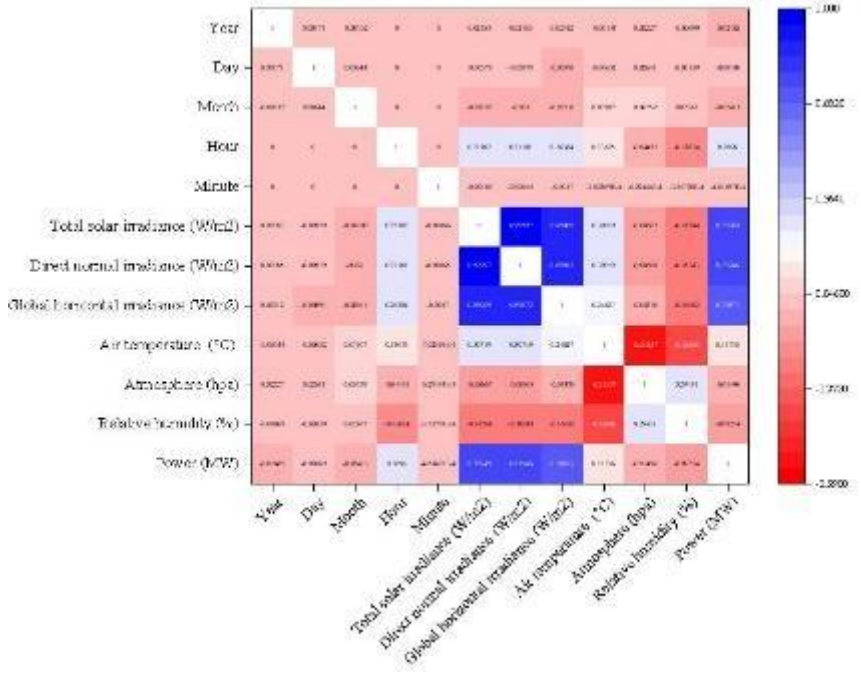


Figure 4: The correlation matrix of features in the farm with a capacity of 130(MW)

In this study, the Delta Moment independent index was used to assess the impact and sensitivity of input parameters on the output. The scaled values range between 0 and 1. Fig 5 illustrates the impact and sensitivity of input parameters at the 30-megawatt site. According to this figure, the three primary parameters, total solar irradiance, direct normal irradiance, and global horizontal irradiance, exhibited very high sensitivity. Additionally, the hour parameter also showed significant influence based on this index. Other parameters showed relatively similar sensitivity to the output.

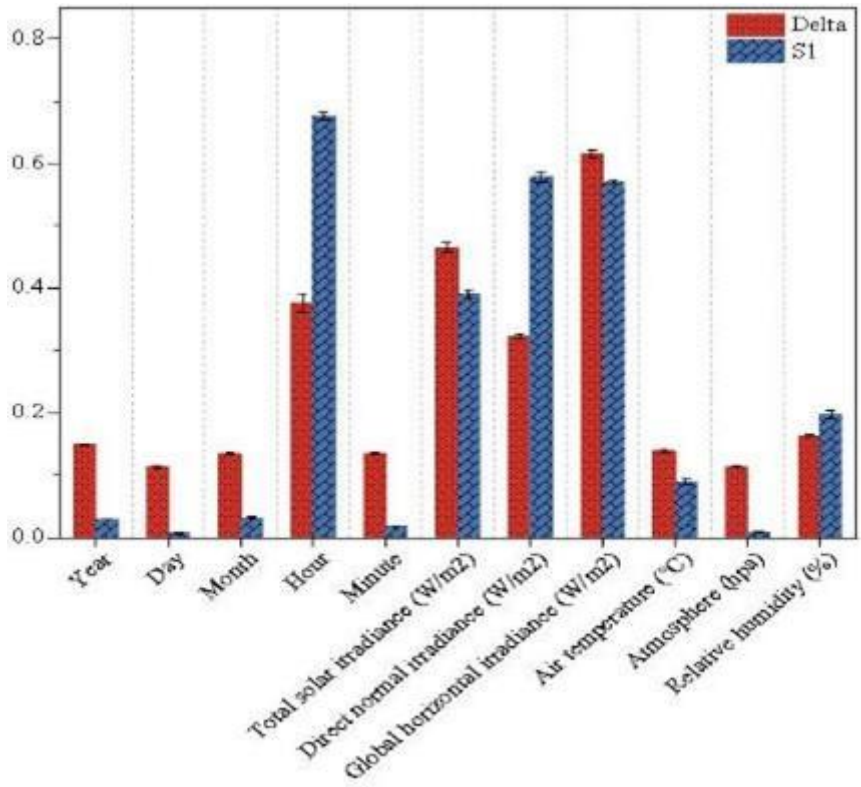


Figure 5: Sensitivity analysis of variables based on the DMIM method in the farm with a capacity of 30(MW)

Fig 6 also illustrates the sensitivity analysis of input parameters at the 130-megawatt site. In this case, the three parameters, total solar irradiance, direct normal irradiance, and global horizontal irradiance, showed

higher sensitivity, with total solar irradiance exhibiting the greatest sensitivity. Similarly, the hour parameter had a significant impact and demonstrated high sensitivity at this site. Other parameters, such as temperature, humidity, atmospheric pressure, and large-scale time parameters, showed relatively similar levels of influence and sensitivity.

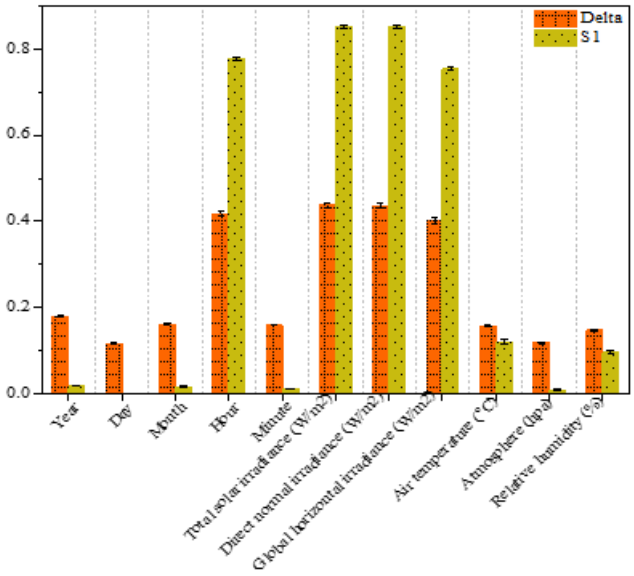


Figure 6: Sensitivity analysis of variables based on the DMIM method in the farm with a capacity of 130(MW)

Fig 7 displays the time series of observational and computational data based on single algorithms for predicting energy production in the 130-megawatt solar farm. In addition to the time series plots, scatter plots for each method are also provided. According to Figure 7, both the training and testing sections showed better overlap between observational and computational data for the Cat Boost algorithm, indicating its satisfactory performance. Conversely, the AdaBoost algorithm exhibited the poorest performance. Furthermore, based on the scatter plot, the Cat Boost algorithm demonstrated the highest correlation with observational data, with an R2 value of 0.8980, making it the most suitable algorithm.

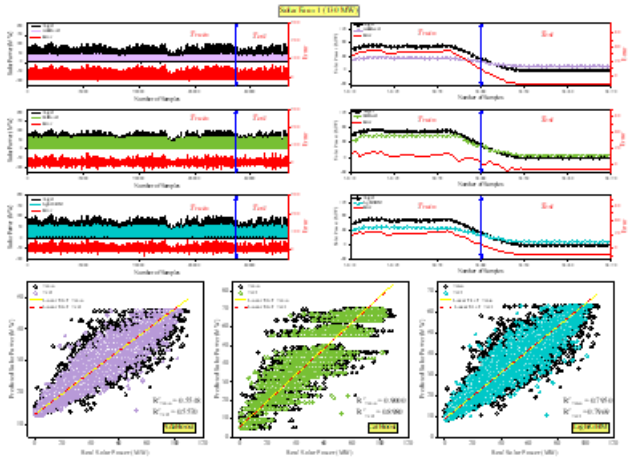


Figure 7: A detailed analysis of the outcomes from employing the AdaBoost, Light GBM, and Cat Boost models in the 130-megawatt solar farm

Fig 8 also depicts the time series of observational and computational data for the 30-megawatt farm. According to the results, the Cat Boost algorithm outperformed other algorithms in this case as well, exhibiting lower error and higher correlation with observational data. Additionally, based on the scatter plot, the Cat Boost algorithm demonstrated the best performance for energy production prediction, with an R2 value of 0.9106.

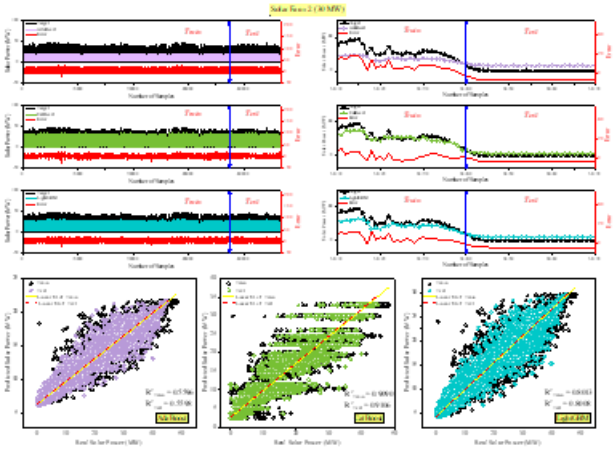


Figure 8: A detailed analysis of the outcomes from employing the AdaBoost, Light GBM, and Cat Boost models in the 130-megawatt solar farm

To conduct a comprehensive assessment of the algorithms' performance and accuracy, various statistical indicators were evaluated and compared, as presented in the preceding section. The results for these indicators are also depicted in Table 5.

Table 5: Error metrics for proposed Cat Boost, AdaBoost, and Light GBM models

Farm	**Optimizer**	**MAE (Train)**	**RMSE (Train)**	**R^2 (Train)**	**JSD(Train)**	**VAF(Train)**	**RAE(Train)**
130 MW	AdaBoost	15.575	18.732	0.555	213905.3	55.48413	0.546277
	CatBoost	6.832367	8.875713	0.900052	70437.51	90.00517	0.258847
	LightGBM	10.41079	12.71163	0.794991	125100	79.49912	0.370716
30 MW	AdaBoost	7.691782	9.115634	0.559622	30828.01	55.96231	0.54213
	CatBoost	3.356744	4.144317	0.908976	7841.534	90.89757	0.246473
	LightGBM	5.124735	6.122702	0.801327	14322.59	80.13274	0.364133
Farm	**Optimizer**	**MAE (Test)**	**RMSE (Test)**	**R^2 (Test)**	**JSD(Test)**	**VAF(Test)**	**RAE(Test)**
130 MW	AdaBoost	15.39225	18.32239	0.556952	71634.8	55.70914	0.54584
	CatBoost	6.778392	8.78999	0.898032	23920.81	89.80388	0.261861
	LightGBM	10.27857	12.40571	0.796891	42103.83	79.69671	0.369577
30 MW	AdaBoost	7.718301	9.029883	0.559829	10024.49	55.98797	0.542413
	CatBoost	3.339026	4.068872	0.910627	2533.42	91.06506	0.244412
	LightGBM	5.146532	6.075019	0.800771	4718.119	80.08289	0.364919

Hybrid models were devised to boost prediction accuracy and benchmark against individual algorithms. Employing the War SO algorithm, optimization was applied to the Cat Boost, AdaBoost, and Light GBM algorithms. Based on Fig 9, both observational and computational time series results are presented for both farms. According to Figure 8, for Farm 1, the AdaBoost-War SO hybrid model outperformed its single model counterpart, exhibiting the lowest error rate. Similarly, for the 30-megawatt Farm 2, the AdaBoost-War SO hybrid model proved suitable for prediction purposes.

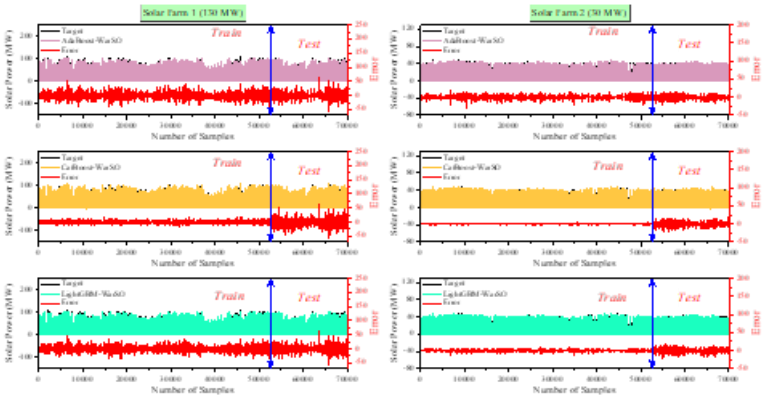


Figure 9: Evolution of Observed and Predicted Values using Hybrid Models of AdaBoost, LightGBM, and CatBoost

To comprehensively analyze and identify the most appropriate prediction algorithms, as well as evaluate their performance, scatter plots for each hybrid model are displayed in Figure 10. These plots visualize the R2 index for both the training and testing datasets. Based on Figure 10, in the first farm, the hybrid AdaBoost-War SO model achieved the highest performance with an R2 value of 0.9784, while in the second farm, a similar result was observed with the hybrid AdaBoost-War SO m

odel achieving an R2 value of 0.9836. Following these models, the hybrid Light GBM-War SO model proved to be suitable for prediction in both farms.

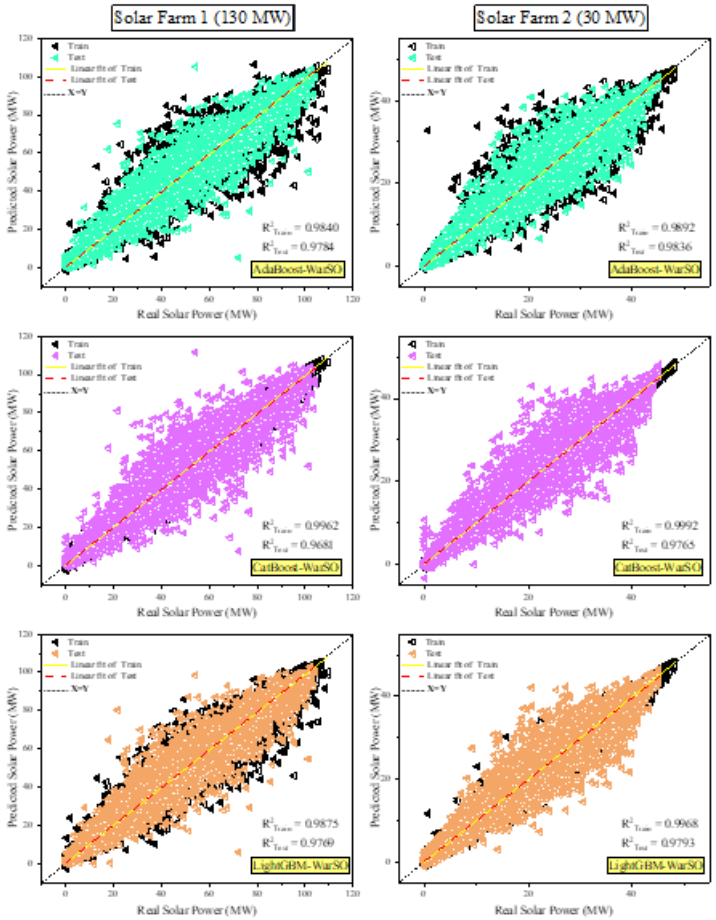


Figure 10: Scatter plot of the observation-prediction for AdaBoost, Light GBM, and Cat Boost hybrid models in Farm1 and Farm2

Fig 11 illustrates the scatter plot of errors in hybrid models for both the training and testing phases. According to this figure, during the training phase, the Cat Boost-War SO model performed the best in both farms. However, during the testing phase, although the results are close, the AdaBoost-War SO model exhibited lower error ranges in both farms, indicating its suitability for prediction purposes.

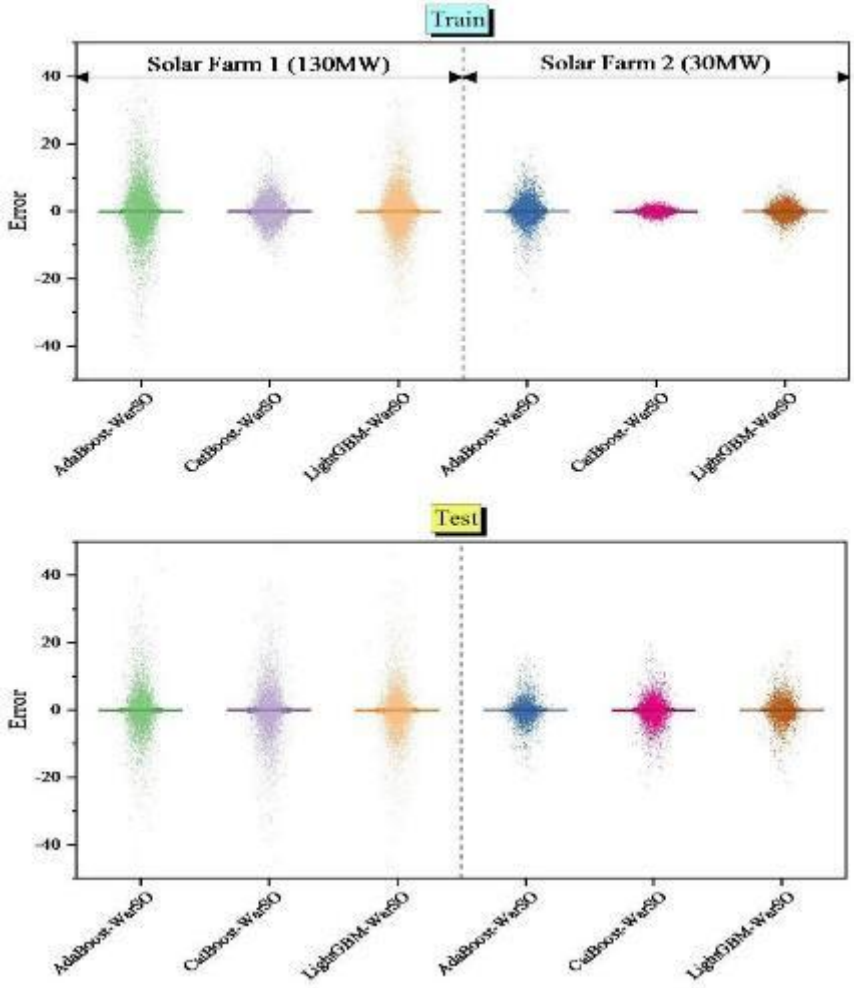


Figure 11: plots of error measurements for models during the testing and training phases in Farm1 and Farm2

Fig 12 displays the error metrics calculated for the hybrid models proposed for energy production prediction in the first farm. The calculated metrics include RMSE, R2, RAE, JSD, MAE, and VAF. Considering the two important metrics, RMSE and R2, from the RMSE plot in the testing section, it is evident that the AdaBoost-War SO hybrid model had the lowest error. Following this model, the Light GBM-War SO model proved suitable for prediction. Additionally, considering the R2 metric, it is evident from the rectangular plot in the testing section that the AdaBoost-War SO hybrid model had the highest R2. The other metrics also support this trend.

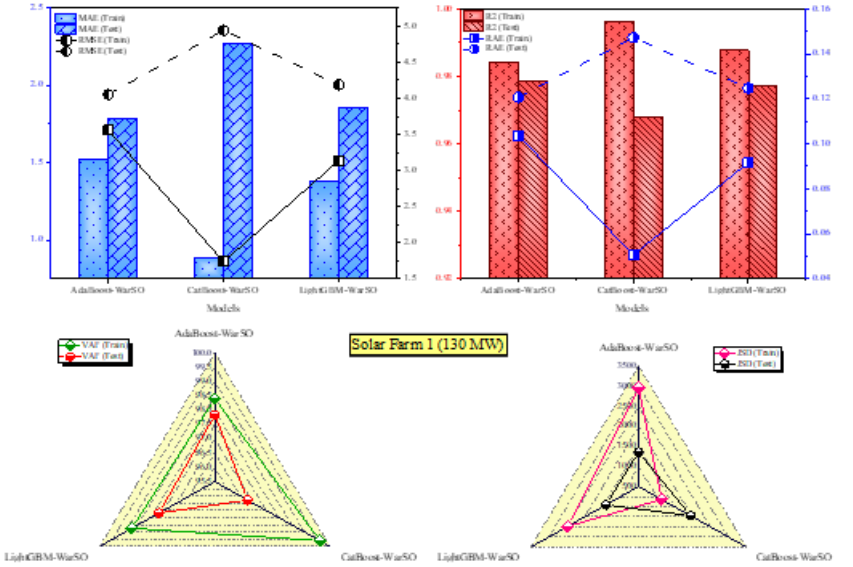


Figure 12: Performance Metrics Visualization for Proposed Models in Farm 1(130MW)

Fig 13 also illustrates the error metrics calculated for the hybrid models in the second farm with a capacity of 30 megawatts. Similar to the first farm, according to the presented metrics, the AdaBoost-War SO hybrid model has proven to be the best model for prediction in this farm as well. Details and numerical values for each of the indicators for hybrid models are presented in Table 6.

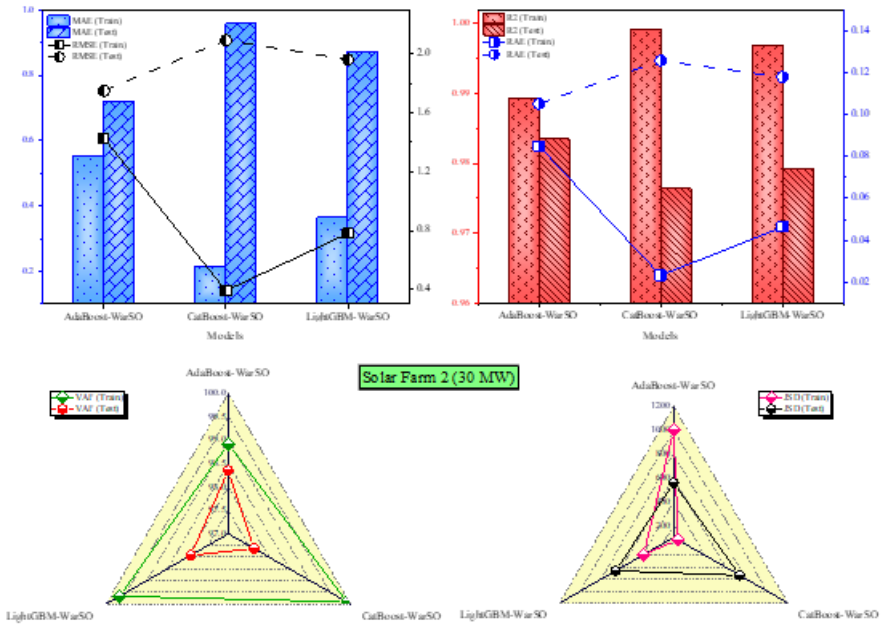


Figure 13: Performance metrics visualization for proposed models in Farm2 (30MW)

Table 6: Error metrics derived from the application of Cat Boost, AdaBoost, and Light GBM hybrid models

Optimizer	AdaBoost-War SO	Cat Boost-War SO	Light GBM-War SO	AdaBoost-War SO	Cat Boost-War SO	Light GBM-War SO
	Farm1(130 MW)			Farm2(30 MW)		
	Train					
MAE	1.51868	0.884092	1.373611	0.550378	0.214323	0.364667
RMSE	3.55647	1.735859	3.138754	1.430018	0.391776	0.779854
R2	0.983952	0.996177	0.987501	0.989162	0.999187	0.996777
JSD	2956.739	1143.396	2482.084	1001.383	143.2308	388.9295
VAF	98.39525	99.61771	98.75007	98.91624	99.91866	99.67769
RAE	0.103719	0.050624	0.091537	0.085047	0.0233	0.04638
	Test					
MAE	1.784966	2.265224	1.853122	0.71745	0.954989	0.870113
RMSE	4.053808	4.942803	4.187876	1.74971	2.094413	1.96203
R2	0.978312	0.967757	0.976854	0.983473	0.97632	0.979219
JSD	1356.879	1951.165	1413.484	555.6971	736.2619	659.5611
VAF	97.83394	96.77995	97.68843	98.35126	97.63708	97.92325
RAE	0.120766	0.14725	0.12476	0.105103	0.125809	0.117857

Fig 14 presents the runtime performance of hybrid models over 500 iterations. Based on Fig14, in the first farm, the AdaBoost-War SO hybrid model had the longest runtime with 3403 seconds, followed by the Light GBM-War SO hybrid model. Similarly, in the second farm, the AdaBoost-War SO model had the longest runtime, totaling 3960 seconds. The Cat Boost-War SO model had the shortest runtime in both farms.

Although hybrid models, in particular AdaBoost-War SO, are more accurate, their usability in the real world is diminished by high runtime expenses. As Fig 14 shows, the AdaBoost-War SO model is significantly more computationally expensive than solo models, with some instances requiring over 3960 seconds of runtime. The high computational demand stems from the ensemble learning iteratively on top of the optimization approach being used by the War Strategy Optimizer.

The extended period of operation might pose challenges in real-time operations or scenarios defined by limited computing resources. Despite the improvements in precision, justification for the use of hybrid models in critical forecast scenarios is met, their practicality in the context of sparse resources, for instance, in edge devices or small-scale microgrids, might be limited. To mitigate this trade-off, future work is invited to investigate optimization methods, including model parallelization, hardware acceleration, or pruning strategies, to minimize runtime without compromising accuracy. Furthermore, combining hybrid models with distributed computing platforms can increase their scalability for large-scale deployment.

This analysis highlights the significance of striking a balance between model performance and computational efficiency, such that hybrid models are still effective and feasible for a broad variety of solar energy forecasting applications.

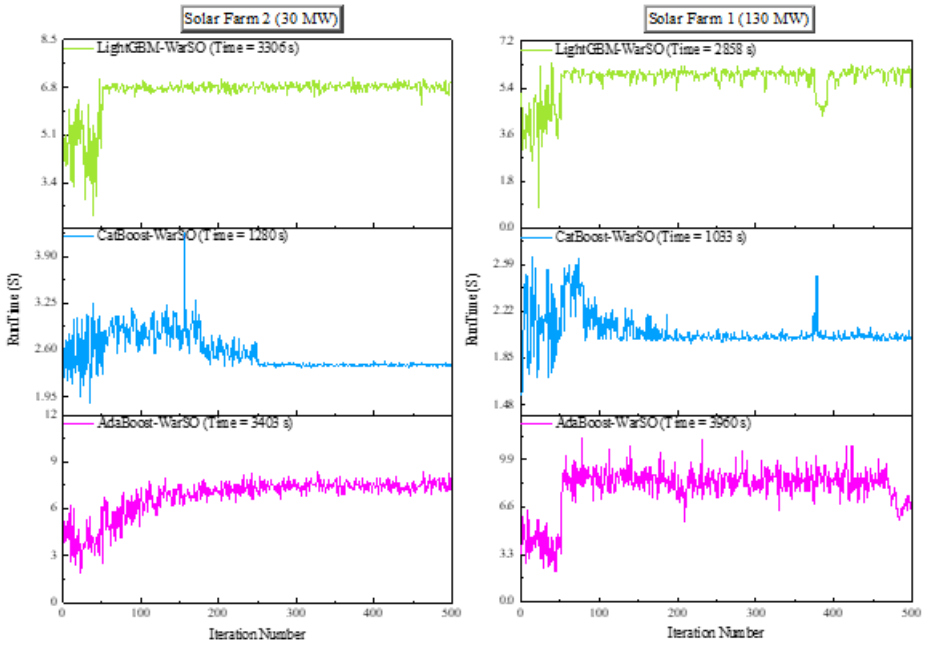


Figure 14: Comparison of runtime for various hybrid models in both Farm1 and Farm2

Fig 15 illustrates the convergence chart for the hybrid models, using the Mean Squared Error (MSE) index as the convergence metric with a set number of iterations at 300. Based on Figure 15, the values for the first farm exhibit higher MSE, whereas for the second farm, these values are lower. In the first farm, the hybrid AdaBoost-War SO model has the lowest MSE. Similarly, in the second farm, as expected, the hybrid AdaBoost-War SO model has the lowest MSE.

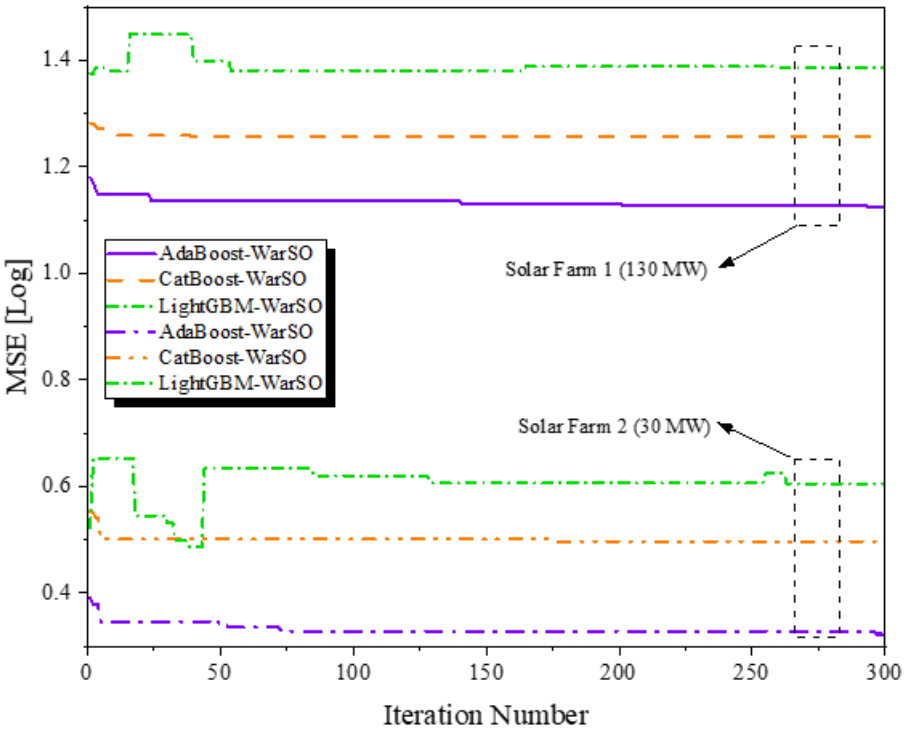


Figure 15: The convergence plots of the Cat Boost, AdaBoost, and Light GBM hybrid model

4 Discussion

The outcome of this study exhibits significant improvement in solar power prediction over the SOTA benchmarks due mainly to the incorporation of machine learning methods in addition to the War SO optimization algorithm. Out of the proposed methods, the AdaBoost-War SO model stood out, with R^2 = 0.9836 and root mean square error (RMSE) = 1.75 MW for the 30 MW solar power plant. This performance surpassed the performance of single algorithms like CatBoost (R^2 = 0.9106, RMSE = 4.06 MW) and even other combinations, i.e., CatBoost-War SO and LightGBM-War SO.

4.1 Reason for high performance of AdaBoost-War SO

The iterative boosting mechanism of AdaBoost allows it to correct errors yielded by weak learners, thereby allowing it to effectively model the nonlinear relationships that are inherent in solar energy data. The model flexibility remains supplemented by the War SO optimizer that optimally trades off exploration of new parameter spaces and exploitation of already known optimum solutions. Through this dual capability, the hybrid model is assured of converging to an improved global optimum than traditional optimization methods like grid search or particle swarm optimization (PSO).

The AdaBoost-War SO hybrid model that combined both of them possessed lower prediction variance, especially when testing, which means better generalization power. Although CatBoost and LightGBM can be used as individual models, their precision was restricted due to the absence of an external optimization platform for dynamic fine-tuning of the hyperparameters.

4.2 Importance of high-sensitivity parameters

The sensitivity analysis revealed that global horizontal irradiance, direct normal irradiance, and total solar irradiance were the most significant parameters used to predict solar energy production. These findings have very practical applications:

4.2.1 Improved feature selection

Models can reduce computational requirements and, simultaneously, increase accuracy by concentrating on the most sensitive parameters. By giving priority to these features, noise caused by unimportant variables is reduced.

4.2.2 Instant predictions

Continuous real-time monitoring of high-sensitivity parameters is essential for forecasting system operation. Improved sensors must favor quality solar irradiance measurement for input data for forecasting.

4.2.3 Site-specific calibration

The high sensitivity of irradiance parameters makes location-specific model calibration highly necessary, as irradiance behavior varies greatly with climate and geography. Region-specific models provide more accurate energy forecasting.

4.2.4 Supporting grid stability

Accurate prediction based on high-sensitivity parameters enhances the integration of solar power into the electricity grid. Reducing prediction errors allows grid managers to balance demand and supply, thus ensuring stability and avoiding outages.

4.2.5 Resource and risk management

Sensitivity analysis outputs may be used to inform resource planning, e.g., investing in high-end measurement technologies, and formulating risk reduction strategies for solar power plants. Understanding the major drivers of variability can allow for contingency planning to address outages due to weather.

4.3 Dataset and benchmark comparisons

Hybrid models, specifically the AdaBoost-War SO model, always performed better than single models and other hybrid combinations in accuracy metrics. For example, the AdaBoost-War SO model had an R^2 of 0.9836 and RMSE of 1.75 MW for the 30 MW solar farm and did better than the research of previous scholars such as Suanpang and Jamjuntr (2024), where LGBM had an R^2 of 0.84 and RMSE of 5.77 W. Likewise, in comparison with Singh et al. (2023), whose hybrid model using GRU enhanced accuracy for large systems, the current research demonstrated improved generalization on multi-site datasets.

The application of the War SO optimizer was significant in enhancing the performance of the AdaBoost-War SO model. Through its provision of a trade-off between exploration and exploitation, War SO allowed for efficient hyperparameter adjustment, thus avoiding local optima—a limitation that is usually faced with typical optimization methods like grid search or genetic algorithms used in state-of-the-art models. This further strengthened the capability of the hybrid models to efficiently capture the non-linear relationships in the data.

4.4 Optimization and computational efficiency

The second significant contribution of this study is its consideration of runtime and convergence. Even though the AdaBoost-War SO model was the most precise, its runtime was comparatively higher because both AdaBoost's iterative boosting and War SO's optimization are slow processes. However, this is warranted due to the substantial improvements in predictive accuracy and trustworthiness. Convergence analysis indicated that War SO significantly lowered the possibility of trapping in local optima, especially in high-dimensional parameter spaces, hence making it an apt option for hybrid model optimization for renewable energy forecasting.

4.5 Computational cost: Trade-Offs between accuracy and runtime

Increased accuracy of the hybrid models presented here, AdaBoost-War SO, comes with increased computational costs, thus a compromise between accuracy and computation time. The AdaBoost-War SO model was more accurate, achieving R^2 of 0.9836 and RMSE of 1.75 MW for the 30 MW farm but also had the highest processing time of approximately 3,960 seconds, as shown in Fig 14. Its high computational complexity is due to both the iterative approach of AdaBoost with training multiple weak learners and dynamic adjustment of their weights and the optimization approach of War SO that balances exploration and exploitation via successive iterations. While the enhanced accuracy significantly reduces prediction error and enables great generalization on diverse datasets, heightened runtime is a scalability problem in large-scale or real-time applications, for instance, energy network integration. Nevertheless, other hybrid models, for instance, CatBoost-War SO with an R^2 score of 0.9763 and considerably lower runtime, offer an acceptable trade-off and therefore are viable where computational efficiency matters. To mitigate the computational expense of AdaBoost-War SO, parallelization, distributed computing, and dynamic model selection can be used. Such methods can achieve a balance between accuracy and execution time, enabling hybrid models for specific needs in forecasting. Whereas AdaBoost-War SO is appropriate for applications where precision is paramount, more speedy options can be adequate for less resource-intensive applications, indicating a compromise between efficiency and performance.

4.6 Broader implications

The study brings into focus the potential of hybrid machine learning architectures augmented by innovative algorithms such as War SO. The accurate forecasting of solar energy generation, as a function of high sensitivity parameters and efficient optimization methods, is of particular importance to power grid reliability, resource planning, and power system integration of renewable energies. This work sets the new standard for predicting solar energy by overcoming key limitations in current best-practice methods, including low data resolution, sparse sensitivity testing, and the lack of hybrid optimization.

The fluctuation in the performance measures, i.e., RMSE and MAE, from training to test data indicates possible overfitting in certain of the models. For example, models such as CatBoost performed best during training (e.g., R^2 = 0.608, RMSE = 4.478 W, MAE = 3.367 W) but significantly declined during test (R^2 = 0.46, RMSE = 4.748 W, MAE = 3.583 W). This gap indicates that while the model was able to find patterns in the training set, it was struggling to generalize to novel, unseen data.

4.7 Comparative performance across farms

The performance of the models was extremely inconsistent between the 130 MW and 30 MW solar farms, showing the role of farm capacity and characteristics of data in model performance. For 130 MW, the models were subjected to higher variability of important parameters such as solar irradiance and temperature, seemingly due to the higher geographical spread of the farm. This greater exposure to more variable microclimatic conditions brought more noise into the data, and therefore it was more challenging to gain precise predictions. Conversely, the smaller 30 MW farm provided more consistent conditions, so there was less variance and the models could operate better.

Performance variations can also be accounted for by dataset-specific factors. The 130 MW dataset included higher variability in solar irradiance, which negatively affected the potential of the models to generalize well. The 30 MW farm dataset included more uniform patterns, and these translated into higher accuracy results for most of the models. These findings suggest that site-specific factors such as farm size, local weather, and dataset variability play important roles in the effectiveness of forecasting models.

To address these issues, model site-specific calibration is required. Normalization of the dataset per agricultural field aided the models in conforming to particular patterns with lesser effort; additional advances can be achieved by incorporating additional features, including wind speed and cloud cover, to better reflect environmental variation. In addition, the creation of hybrid approaches that combine localized tuning with generalized prediction capability promises improved scaling up of such models to farms of varied sizes and conditions.

This research places importance on how one should consider the specific nature of every farm while forecasting solar power and the need for subsequent research with models being tested under different geographic and operational conditions. These results add to the body of knowledge regarding solar farm capacity and dataset attributes and how they influence model performance and consequently enable the design of more precise and adaptive forecasting models.

4.8 Limitation

Overfitting is a result of many different causes, including model complexity that is too high, lack of diversity in the training data, or weak regularization. Combating it is important for maintaining the reliability and stability of forecast models in actual usage. Cross-validation, early stopping, and hyperparameter tuning are some of the methods that can reduce overfitting by avoiding the model from over-focusing on noise or irrelevant patterns in the training data.

Future research activities can explore the use of simpler models or hybrid approaches that balance predictive power with the ability to generalize. Expanding the dataset to cover a wider variety of diverse and representative samples, such as data from various geographic regions or seasonal differences, could also help increase model performance and reduce the danger of overfitting. In addition, the use of methods like dropout or L2 regularization in models such as CatBoost and LightGBM could potentially increase their generalizability to different datasets.

Through the elimination of such constraints, future research can make predictive models perform stably on training and testing datasets, thus encouraging their application in volatile and uncertain solar energy conditions.

PCC was able to capture significant features, i.e., solar irradiance, temperature, and humidity, but its focus on linear relationships could have overlooked non-linear relationships that would be useful for model performance. More sophisticated approaches, e.g., mutual information or machine learning model-based feature importance, would provide a more nuanced picture of feature importance, particularly for variables with complex interactions. Also, the exclusion of potentially important meteorological variables such as wind speed and cloud cover might have limited the model's ability to capture environmental heterogeneity to some degree. For example, solar irradiance is highly influenced by wind speed and cloud cover under specific conditions, which could potentially affect prediction under varying weather conditions. These shortcomings can be improved in future studies by including more variables and using strict imputation protocols, which would enable higher generalizability and predictive validity of the proposed models.

4.9 Comparison of solar energy forecasting models

Table 7 provides a concise comparison of key performance metrics across different studies, highlighting the effectiveness and computational considerations of various machine learning approaches in solar energy forecasting. Based on the comparison, the study method, which employs the AdaBoost-War SO hybrid model, demonstrates superior performance in solar energy forecasting.

Table 7: Comparison of solar energy forecasting models

Aspect	Best Model R^2	Best Model RMSE
This study (AdaBoost-War SO)	0.9836	1.75 MW
Nguyen et al. (2025) (CatBoost)	0.608 (Training), 0.46 (Testing)	4.478 W (Training), 4.748 W (Testing)
Suanpang and Jamjuntr (2024) (LightGBM)	0.84	5.77

5 Conclusion

This study demonstrates the ability of hybrid machine learning models, optimized by the War SO algorithm, to improve solar power prediction accuracy. Using high-resolution data that was recorded every 15 minutes and advanced feature selection techniques, such as the Delta Moment Independent Measure (DMIM), the models achieved improved performance compared to their separate models. The recognition of solar irradiance as the largest contributing factor aligns with earlier research; yet, using DMIM in this study is a more rigorous sensitivity analysis, therefore further enriching knowledge on its effects on energy output.

The findings indicate the promising prospect of improving the accuracy of forecasts, but broader implications for large-scale renewables integration and grid stability necessitate further investigation. The findings have specific significance to standalone energy management use cases, such as solar energy installation operation optimization and policy guidance for storage and grid balancing. Future studies should extend on these findings through experiments with testing the models across different geographical and climatic locations and assessing their implementation in actual-time grid management systems.

This study deals with the critical issues of solar forecasting, thus making renewable energy systems more efficient and reliable. Nevertheless, the findings are presented cautiously under the limitations of the study, outlining the short-term practical applications and laying the ground for further development in the field of renewable energy forecasting.

Abbreviation

AdaBoost	Adaptive Boosting	NWS	National Weather Service
ANN	Artificial Neural Network	P	Prior value
ARIMA	Autoregressive integrated moving average	PI	Prediction intervals
Bt	The normalization factor	PV	Photovoltaic
C	Commander	R2	Coefficient of Determination
Cat Boost	Categorical Gradient Boosting	Rai	The ranking
DL	Deep Learning	RAE	Relative Absolute Error
Dt(i)	Uniform sample distribution	RES	Renewable energy sources
et	Average error	RF	Random forest
Fper	The previous situation	RMSE	Root means square error
Fnew	The new situation	SVM	Support vector machines
ft(xi)	A weak predictor		
GBT	Gradient boosting tree	SVR	Support Vector Regression
JSD	Jensen Shannon Divergence	VAF	Variance Accounted For
K	The King	War SO	War Strategy Optimizer
LACE	Levelized Avoided Cost of Electricity	α	The corresponding weight
LCOE	Levelized Cost of Electricity	xk	The random vector
Light GBM	Light Gradient Boosting Machine	yK and yC	The situations of the King and The Commander
LR	Linear regression		
MADSR	Monthly average daily solar radiation		
MAE	Mean Absolute Error		
ML	Machine Learning		
NWP	Numerical weather prediction		

Acknowledgements

I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Competing interests

The authors declare no competing interests.

Authorship contribution statement

Fenghong Pan: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Data availability

Data can be shared upon request.

Declarations

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

All the authors have read and approved the manuscript. As stated earlier in this document, the requirements for authorship have been met, and each author believes that the manuscript represents honest work.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper and will take public responsibility for its content.

References

[1] S. Koohi-Fayegh, M.A. Rosen, A review of renewable energy options, applications, facilitating technologies and recent developments, European Journal of Sustainable Development Research 4 (2020) em0138.

[2] K.S. Perera, Z. Aung, W.L. Woon, Machine learning techniques for supporting renewable energy generation and integration: a survey, in: Data Analytics for Renewable Energy Integration: Second ECML PKDD Workshop, DARE 2014, Nancy, France, September 19, 2014, Revised Selected Papers 2, Springer, 2014: pp. 81–96.

[3] D. Gielen, F. Boshell, D. Saygin, M.D. Bazilian, N. Wagner, R. Gorini, The role of renewable energy in the global energy transformation, Energy Strategy Reviews 24 (2019) 38–50.

[4] W. Strielkowski, L. Civín, E. Tarkhanova, M. Tvaronavičienė, Y. Petrenko, Renewable energy in the sustainable development of electrical power sector: A review, Energies (Basel) 14 (2021) 8240.

[5] G.A. Tiruye, A.T. Besha, Y.S. Mekonnen, N.E. Benti, G.A. Gebreslase, R.A. Tufa, Opportunities and challenges of renewable energy production in Ethiopia, Sustainability 13 (2021) 10381.

[6] N.E. Benti, T.A. Woldegiyorgis, C.A. Geffe, G.S. Gurmesa, M.D. Chaka, Y.S. Mekonnen, Overview of geothermal resources utilization in Ethiopia: Potentials, opportunities, and challenges, Sci Afr 19 (2023) e01562.

[7] N.E. Benti, A.B. Aneseyee, C.A. Geffe, T.A. Woldegiyorgis, G.S. Gurmesa, M. Bibiso, A.A. Asfaw, A.W. Milki, Y.S. Mekonnen, Biodiesel production in Ethiopia: Current status and future prospects, Sci Afr 19 (2023) e01531.

[8] N.E. Benti, Y.S. Mekonnen, A.A. Asfaw, combining green energy technologies to electrify rural community of Wollega, Western Ethiopia, Sci Afr 19 (2023) e01467.

[9] C.R. Kumar, M.A. Majid, Renewable energy for sustainable development in India: Current status, future prospects, challenges, employment, and investment opportunities, TIDEE: TERI Information Digest on Energy and Environment 21 (2022) 33.

[10] P. Denholm, D.J. Arent, S.F. Baldwin, D.E. Bilello, G.L. Brinkman, J.M. Cochran, W.J. Cole, B. Frew, V. Gevorgian, J. Heeter, The challenges of achieving a 100% renewable electricity system in the United States, Joule 5 (2021) 1331–1352.

[11] E. Alhamer, A. Grigsby, R. Mulford, The Influence of Seasonal Cloud Cover, Ambient Temperature and Seasonal Variations in Daylight Hours on the Optimal PV Panel Tilt Angle in the United States, Energies (Basel) 15 (2022) 7516.

[12] S. Impram, S.V. Nese, B. Oral, Challenges of renewable energy penetration on power system flexibility: A survey, Energy Strategy Reviews 31 (2020) 100539.

[13] I. Ghalehkhondabi, E. Ardjmand, G.R. Weckman, W.A. Young, An overview of energy demand forecasting methods published in 2005–2015, Energy Systems 8 (2017) 411–447.

[14] A. Krechowicz, M. Krechowicz, K. Poczeta, Machine learning approaches to predict electricity production from renewable energy sources, Energies (Basel) 15 (2022) 9146.

[15] Y.-Y. Hong, T.R.A. Satriani, Day-ahead spatiotemporal wind speed forecasting using robust design-based deep learning neural network, Energy 209 (2020) 118441.

[16] X. Zhao, J. Liu, D. Yu, J. Chang, One-day-ahead probabilistic wind speed forecast based on optimized numerical weather prediction data, Energy Convers Manag 164 (2018) 560–569.

[17] J. Fan, L. Wu, F. Zhang, H. Cai, W. Zeng, X. Wang, H. Zou, Empirical and machine learning models for predicting daily global solar radiation from sunshine duration: A review and case study in China, Renewable and Sustainable Energy Reviews 100 (2019) 186–212.

[18] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, A. Fouilloy, Machine learning methods for solar radiation forecasting: A review, Renew Energy 105 (2017) 569–582.

[19] P. Suanpang, P. Jamjuntr, Machine learning models for solar power generation forecasting in microgrid application implications for smart cities, Sustainability 16 (2024) 6087.

[20] S. Singh, V. Subburaj, K. Sivakumar, R. Anil Kumar, M.S. Muthuramam, R. Rastogi, V. Ratansing Patil, A. Rajaram, Optimum Power Forecasting Technique for Hybrid Renewable Energy Systems Using Deep Learning, Electric Power Components and Systems (2024) 1–18.

[21] H.N. Nguyen, Q.T. Tran, C.T. Ngo, D.D. Nguyen, V.Q. Tran, Solar energy prediction through machine learning models: A comparative analysis of

regressor algorithms, PLoS One 20 (2025) e0315955.
[22] C. Zhu, M. Wang, M. Guo, J. Deng, Q. Du, W. Wei, Y. Zhang, Innovative approaches to solar energy forecasting: unveiling the power of hybrid models and machine learning algorithms for photovoltaic power optimization, J Supercomput 81 (2025) 20.
[23] J. Huertas-Tato, R. Aler, I.M. Galván, F.J. Rodríguez-Benítez, C. Arbizu-Barrena, D. Pozo-Vázquez, A short-term solar radiation forecasting system for the Iberian Peninsula. Part 2: Model blending approaches based on machine learning, Solar Energy 195 (2020) 685–696.
[24] A.E. Gürel, Ü. Ağbulut, Y. Biçen, Assessment of machine learning, time series, response surface methodology and empirical models in prediction of global solar radiation, J Clean Prod 277 (2020) 122353.
[25] M. Alizamir, S. Kim, O. Kisi, M. Zounemat-Kermani, A comparative study of several machine learning based non-linear regression methods in estimating solar radiation: Case studies of the USA and Turkey regions, Energy 197 (2020) 117239.
[26] C. Koo, W. Li, S.H. Cha, S. Zhang, A novel estimation approach for the solar radiation potential with its complex spatial pattern via machine-learning techniques, Renew Energy 133 (2019) 575–592.
[27] N.C. Nath, W. Sae-Tang, C. Pirak, Machine learning-based solar power energy forecasting, Journal of the Society of Automotive Engineers Malaysia 4 (2020) 307–322.
[28] D.S. Kumar, W. Teo, N. Koh, A. Sharma, W.L. Woo, A Machine Learning Framework for Prediction Interval based Technique for Short-Term Solar Energy Forecast, in: 2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE), IEEE, 2020: pp. 406–409.
[29] I. Jebli, F.-Z. Belouadha, M.I. Kabbaj, A. Tilioua, Prediction of solar energy guided by pearson correlation using machine learning, Energy 224 (2021) 120109.
[30] L. Abualigah, R.A. Zitar, K.H. Almotairi, A.M. Hussein, M. Abd Elaziz, M.R. Nikoo, A.H. Gandomi, Wind, solar, and photovoltaic renewable energy systems with and without energy storage optimization: A survey of advanced machine learning and deep learning techniques, Energies (Basel) 15 (2022) 578.
[31] J. Huertas-Tato, R. Aler, I.M. Galván, F.J. Rodríguez-Benítez, C. Arbizu-Barrena, D. Pozo-Vázquez, A short-term solar radiation forecasting system for the Iberian Peninsula. Part 2: Model blending approaches based on machine learning, Solar Energy 195 (2020) 685–696.
[32] M. Alizamir, S. Kim, O. Kisi, M. Zounemat-Kermani, A comparative study of several machine learning based non-linear regression methods in estimating solar radiation: Case studies of the USA and Turkey regions, Energy 197 (2020) 117239.
[33] Y. Chen, J. Xu, Solar and wind power data from the Chinese state grid renewable energy generation forecasting competition, Sci Data 9 (2022) 577.
[34] M.A. Oladipupo, P.C. Obuzor, B.J. Bamgbade, A.E. Adeniyi, K.M. Olagunju, S.A. Ajagbe, An automated python script for data cleaning and labeling using machine learning technique, Informatica 47 (2023).
[35] A.V. Dorogush, V. Ershov, A. Gulin, CatBoost: gradient boosting with categorical features support, ArXiv Preprint ArXiv:1810.11363 (2018).
[36] J. Fan, X. Wang, F. Zhang, X. Ma, L. Wu, predicting daily diffuse horizontal solar radiation in various climatic regions of China using support vector machine and tree-based soft computing models with local and extrinsic climatic data, J Clean Prod 248 (2020) 119264.
[37] E.K. Ampomah, Z. Qin, G. Nyame, F.E. Botchey, Stock market decision support modeling with tree-based AdaBoost ensemble machine learning models, Informatica 44 (2021).
[38] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J Comput Syst Sci 55 (1997) 119–139.
[39] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, Adv Neural Inf Process Syst 30 (2017).
[40] T.S.L. V Ayyarao, N.S.S. Ramakrishna, R.M. Elavarasan, N. Polumahanthi, M. Rambabu, G. Saini, B. Khan, B. Alatas, War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization, IEEE Access 10 (2022) 25073–25105.
[41] H. Hu, S. Gong, B. Taheri, Energy demand forecasting using convolutional neural network and modified war strategy optimization algorithm, Heliyon (2024).
[42] H. Khajavi, A. Rastgoo, Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms, Energy 272 (2023) 127069.
[43] C. Pasion, T. Wagner, C. Koschnick, S. Schuldt, J. Williams, K. Hallinan, Machine learning modeling of horizontal photovoltaics using weather and location data, Energies (Basel) 13 (2020) 2570.

5G-Optimized Deep Learning Framework for Real-Time Multilingual Speech-to-Speech Translation in Telemedicine Systems

Medapati Venkata Manga Naga Sravan[1*], K Venkata Rao[2]
[1]Andhra University
[2]HOD, Dept of Computer Science, Andhra University, India
E-mail: Sravan.medapati@gmail.com, professor_venkat@yahoo.com
*Corresponding author

Keywords: telemedicine, multilingual speech translation, deep learning, Speech-to-Speech workflow, 5G technology

Received: December 15, 2024

Telemedicine has revolutionized healthcare by enabling virtual consultations, yet it still faces challenges from linguistic barriers and the need for real-time, scalable communication. Current systems typically address isolated tasks like speech recognition or symptom classification, lacking a unified solution for multilingual doctor-patient interactions. To address this, we present a 5g-optimized Deep Learning Framework that integrates advanced speech recognition, neural machine translation, and text-to-speech synthesis into a seamless Speech-to-Speech Workflow (STSW). Specifically, our framework utilizes fine-tuned OpenAI Whisper for speech recognition, a Marian MT model fine-tuned on multilingual medical corpora for translation, and Tacotron 2-based neural TTS for speech synthesis. Each model is domain-adapted to handle complex medical terminologies. We implement the framework over 5G-enabled edge computing infrastructure, ensuring real-time performance with ultra-low latency. Experimental results demonstrate the effectiveness of the proposed system, achieving a Word Error Rate (WER) of 0.12, a BLEU score of 0.85 for translation quality, and a Mean Opinion Score (MOS) of 4.5 for the naturalness of synthesized speech. Furthermore, our framework delivers an end-to-end latency of 2.1 seconds, outperforming existing approaches. This integration bridges communication gaps in telemedicine, facilitating accurate multilingual conversations and scalable healthcare delivery across diverse geographies.

Povzetek: Predstavljen je 5G-optimiziran okvir globokega učenja za večjezično govorno prevajanje v telemedicini, ki s prilagojenimi modeli dosega kvalitetne rezultate v realnem času.

1 Introduction

One particular technology affecting modern healthcare is telemedicine, allowing consultation and diagnosis over remote digital platforms. In many multilingual regions, however, communication challenges — primarily linguistic — make it less effective. Current telemedicine setups are limited to single functionalities such as automating triage [1], speech recognition [2], or chatbots specific to a disease [5]. Though these approaches solve some parts of the telemedicine puzzle, they fall short due to the absence of an integrated framework that can facilitate real-time multi-lingual communication between doctors and patients. This communication is vital as it increases accessibility and efficiency in healthcare delivery. Although the literature has identified some exciting opportunities and existing applications, this paper shows that deep learning can substantially drive telemedicine systems forward—for instance, Shi et al. Latif et al. [1] used Bi-LSTM as intelligent triage. In [3], the scalability of speech recognition technologies in healthcare. However, current systems have limitations in scalability, latency, and adaptability to clinical settings—furthermore, research, including those of Kandpal et al. While [5] focuses on chatbots designed for communication in healthcare, they tend to ignore multilingual, real-time speech-turn-taking interactions. This gap highlights the importance of an end-to-end multilingual speech-to-speech system for telemedicine.

This study intends to establish a Speech-to-Speech Workflow (STSW), which claims to be a novel framework to combat these obstacles. The main aim is to incorporate sophisticated speech recognition, translation capability, and text-to-speech synthesis into an integrated system for telemedical applications. Here, the novelty of this research is due to the use of domain-specific fine-tuning techniques for medical, unique arrangement towards multilingual capabilities integration, and the process of real-time transliteration based on 5G technology. These properties make the architecture suitable for scalable, flexible services for a range of healthcare services. This research adds value from several perspectives. First, it presents a workflow for telemedicine speech-to-speech translation in many languages. Secondly, it performs better than the state-of-the-art systems in all dimensions of accuracy in speech recognition, translation quality, and the naturalness of speech produced by synthesis. Third, it offers a latency-optimized infrastructure for real-time interactions, focusing on key issues in telemedicine communication.

To systemically guide this research, we crafted the following central research questions (RQs):

RQ1: What kind of deep learning-architecture-based framework can be implemented to overcome the multilingual barrier in doctor-patient communication in telemedicine systems?
RQ2: How can the proposed system provide speech-to-speech translation performance in real time while keeping low latency and high scalability?
RQ3: How does integrating 5G technology help the adaptability and reliability of speech-based telemedicine applications in various healthcare environments?
To resolve them, we introduce a 5G-optimized deep learning framework that combines speech recognition, neural machine translation, and text-to-speech open-source solutions and optimizes them for medical vocabulary and multilingual use. We report across important data points end to end in Word Error Rate (WER), BLEU score, Mean Opinion Score (MOS), and latency appropriate for telemedicine use.
Our primary contribution is a cohesive adaptation of domain-based fine-tuning and 5G-specific optimization within a real-time, multilingual, speech-to-speech translation system designed specifically for telemedicine. While existing approaches use general ASR and translation models, we adapt both Whisper and Marian MT models to multilingual medical datasets to improve the recognition and translation of specialized medical terminology.
The presented telemedicine framework centers around the advantages of deep learning for individualized care throughout the telemedicine system. In particular, we use a fine-tuned Whisper ASR model to perform accurate multilingual speech recognition to manage the variance in speech from patients: Real-time, domain-specific translation using the Marian MT Transformer model bridging the communication gap between Doctor and Patient. The model is fine-tuned Tacotron 2, ensuring the speech synthesis produces a natural, context-aware audio output. Moreover, we integrate a BERT-based model for sentiment analysis to extract emotional signals from patient's speech, addressing a gap in empathetic healthcare communication. In contrast to existing systems that handle these different components in a siloed way, our framework integrates all of the modules in a one-stop shop for a real-time, low-latency telemedicine solution that scales to multiple languages.
The remaining structure of the paper is as follows. We summarize the existing literature and identify the research gaps in multilingual telemedicine systems in Section 2. Section 3 proposes the methodology, the details of the STSW framework, and its components. Experimental results and a comparison between the system and state-of-the-art approaches are presented in Section 4. Section 5 discusses the results broadly and describes the study's limitations. Finally, Section 6 concludes with a brief discussion of its implications and directions for future work on broadening linguistic capabilities, tightening semantic precision, and supporting offline telemedicine.

2 Related works

Recent advancements in telemedicine highlight the need for multilingual, real-time communication systems. Existing studies focus on isolated tasks that lack integration. Shi et al. [1] précised classifying patient symptoms, an intelligent triage model that combines Bi-LSTM with character embedding to improve telemedicine services. Payan et al. [2] revealed potential problems for patients from marginalized communities as community health centers adopted telemedicine at a rapid rate. Latif et al. [3] confronted scalability and technological integration hurdles; deep learning-driven speech technology could revolutionize the healthcare industry. Ji et al. [4] provided accessible interpretation services, and mobile healthcare apps may be able to reduce language barriers in the medical field. Kandpal et al. [5] highlighted the increasing influence of artificial intelligence (AI) through chatbots, or virtual assistants, employing ML and AI to evolve from menu-based models to contextual ones. It highlights the convergence of NLP and deep learning and explores their possibilities in healthcare for predictive diagnosis and scheduling of appointments. The study highlights the revolutionary potential of chatbots in healthcare and corporate settings and emphasizes the necessity of well-trained models in service-oriented companies. It also evaluates existing applications, problems, and prospects.
Albahri et al. [6] examined how wearable sensors, networks, artificial intelligence, and cloud computing are all incorporated into telemedicine. One hundred forty-one publications are categorized by a systematic review highlighting the advances and problems in IoT-based healthcare and providing guidance for future studies. Li et al. [7] developed in digital and telecommunications, including AI, 5G, and IoT, are revolutionizing ophthalmology and improving telemedicine capabilities in the face of COVID-19 problems. Zhang et al. [8] employed deep learning and automated transcription to find themes associated with depression in speech recordings made by 265 clinical patients. Calambur et al. [9] examined the effects of language barriers on information collecting in an older adult telehealth service. Talpada et al. [10] can better understand influence by utilizing social media data, especially from Twitter, which provides insights into public attitude.
Yu et al. [11] examined an entire health-related Internet of Things architecture, focusing on cloud platform integration and multimodal sensor technologies for improved emotional connection and user experience. Ozyegen et al. [12] tackled the problem of information overload in healthcare by investigating helpful text-highlighting strategies to support medical practitioners. Chung et al. [13] use a language model and Deep Voice 2; this pilot project investigates specialized voice recognition for nursing shift handovers. Deepa and Khilar [14] developed speech technology in healthcare, which can be attributed to its non-invasive nature and ability to monitor and diagnose diseases. Tripathi et al. [15] affected articulation in speech by impairing muscular control. Clinicians and patients benefit from accurate minimal-word intelligibility tests.

Table 1: Comparative summary of state-of-the-art approaches in telemedicine systems

Study	**Methodology**	**Focus Area**	**Limitations**	**Gaps Addressed by STSW**
Shi et al. [1]	Bi-LSTM for intelligent triage	Symptom classification	No multilingual support lacks integration	STSW supports multilingual speech, integrates triage, recognition, translation
Latif et al. [3]	Deep learning-based speech recognition	Speech recognition	Lacks translation & scalability, high latency	STSW combines recognition + translation + TTS, 5G optimization reduces latency
Kandpal et al. [5]	Chatbot using ML & AI	Text-based chatbots	No real-time speech handling, not multilingual	STSW enables speech-to-speech multilingual real-time communication
Ji et al. [4]	Mobile apps for interpretation	Interpretation services	No scalability lacks integration with speech models	STSW offers end-to-end speech processing integrated with translation
Ganesh et al. [26]	ASR with Flask for disorder speech	Disorder speech recognition	No multilingual translation, not optimized for latency	STSW extends speech recognition to multilingual translation, optimized for 5G

Zhang et al. [16] examined the potential and present difficulties of intelligent speech technology (IST) in healthcare in the face of a lack of resources. It discusses the importance of IST in smart hospitals, namely in illness diagnosis, stroke patient care, and medical documentation. While highlighting AI's progress in voice recognition, the assessment also points out its drawbacks, including a lack of datasets and privacy issues. Kaushik et al. [17], with a considerable accuracy rate, SLINet CNN is a deep learning model for early identification of SLI and DD in children. It is low-complexity for usage in real-time, gender-neutral, and appropriate for remote diagnostics—plans for the future call for adding many languages and continuous speech. Wang et al. [18] presented a novel approach to categorizing voice issues that replaces single vowels with continuous Mandarin speech. Sindhu et al. [19], with speech and vocal impairments, are more likely to experience developmental delays and poor academic performance. Deep learning has transformed automatic detection, which provides prospective advances and helps with effective diagnosis. Huang et al. [20] used the UASpeech dataset, a novel two-stage paradigm for transforming everyday speech to dysarthric speech was suggested and assessed.

Alma et al. [21] examined current developments in deep neural networks for speech and visual applications, focusing on their evolution, difficulties in systems with limited resources, and new applications. Tanveer et al. [22] improved performance on various speech tasks, and ensemble deep learning approaches combine ensemble techniques with deep learning. Shastry [23] presented a method for continuous remote health monitoring in digital health that combines DL and NLP. Sonmez and Varol [24] improved human-computer interaction in Society 5.0, which requires further advancements in speech-emotion recognition (SER). Diverse speech traits and cultural variables that impact recognition accuracy are challenges. Talaat et al. [25] helped CNN-LSTM network-based identification achieve great accuracy by capturing voice airflow dynamics for letter pronunciation.

Ganesh et al. [26] combined ASR technology with Flask to build a powerful disease speech recognition platform that has the potential to revolutionize healthcare and other fields. Musalia et al. [27], with colossal accuracy using the DNN approach, the pilot research assesses SRAVI, a speech/phrase recognition program, with the goal of future development and real-world implementation. Kheddar et al. [28] adapted models to similar datasets; deep Transfer Learning (DTL) in Automatic Speech Recognition (ASR) overcomes the constraints posed by data scarcity. Gaitan et al. [29] prompted telemedicine's uptake, changing people's attitudes and habits in Spain and bringing attention to trends in the country's digital revolution. Bandopadhyay et al. [30] spooked Healthcare Bot (THCB) was created in response to the COVID-19 epidemic, which made it possible to improve remote patient care.

Shahamiri et al. [32] used deep learning to create a Dysarthric Speech Transformer that shows promise in reducing ASR difficulties for those with severe dysarthria. Wu et al. [33] unveiled a scalable precision health solution that combines AI-powered telecare, wearable technology, and ambient data. Applying modular models improves the prediction of chronic diseases. Joshy et al. [35] analyzed deep learning models with different acoustic characteristics for dysarthria severity classification, highlighting the better performance of MFCC-based i-vectors.

Przybylo [36] presented an LSTM-based technique for video plenty sonography-based continuous heart rate monitoring to simplify data processing while maintaining accuracy on par with more established techniques like POS and ICA. Kamble et al. [37] investigated using CNN and SPWVD in an EEG-based BCI system for imagined speech recognition. The results demonstrate notable improvements in performance over conventional techniques, which motivates more research with more enormous datasets and more sophisticated DL structures. Deb et al. [38] presented a deep learning model that achieves 67.71% UAR in categorizing cold speech using MFCC and LPC characteristics. Fernandes [39] enabled telemedicine to connect healthcare across distances, and with COVID-19, it proliferates. AI improves productivity, monitoring, and diagnoses but has drawbacks. Abdelhay et al. [40] provided 24/7 access and financial savings; medical bots—a remote healthcare service—have gained popularity in response to the COVID-19 outbreak. The literature review identifies gaps in telemedicine, particularly in multilingual speech systems.

Benedict and Subair [42] proposed a deep learning-based edge-enabled serverless architecture to detect animal emotion in real time, using a convolutional neural network with serverless computing (SC) to improve scalability and low latency processing. A deep learning framework for social media rumor detection and tracking is proposed by Han and Lin [43], which uses LSTM networks to extract temporal features for better detection accuracy. According to Chen and Zhang [44], an involution feature extraction method was implemented for human posture identification in martial arts, focusing on utilizing a feature extraction technique by convolutional deep learning models, which effectively captured spatial and temporal postural features, resulting in substantial improvements in both classification performance and robustness.

The existing approaches target standalone functionalities, as shown in Table 1, like triage models [1], speech recognition [3], chatbots [5], or interpretation services [4]. Nonetheless, they are limited in offering an efficient, scalable architecture for instant multilingual speech recognition, translation, and speech synthesis with low latency. The proposed STSW framework is proposed to bridge these gaps. It includes modules for speech recognition, translation, and text-to-speech synthesis adapted for medical scenarios, maintains multilingual support, and utilizes 5G technology to enable real-time, scalable, and resource-efficient telemedicine communication. This makes STSW close some of the many gaps across fragmented approaches in literature into a unified, on-demand, multilingual telemedicine system.

Although previous works have made several significant improvements in specific aspects of telemedicine, no unified framework integrates these fragmented components (e.g., speech recognition, symptom triage, translational linguistics, and sentiment analysis) into a single scalable real-time system. Our STSW framework fills this gap by jointly learning these functionalities and supporting real-time, low-latency communication between doctor and patient in a multilingual setting.

Existing approaches emphasize triage, diagnosis, or chatbots. The proposed research addresses these gaps by integrating advanced deep learning-based speech recognition, translation, and synthesis into a unified framework. This will enable real-time, multilingual doctor-patient communication and significantly enhance accessibility and efficiency in telemedicine systems.

3 Proposed system

An empirical approach to the proposed telemedicine system, presented in Figure 1, can be developed by applying advanced deep learning and natural language processing techniques integrated with 5G technology that allows for communication and diagnosis of the patients. The system starts with patient utterances via 5G-enabled audio or video calls. As a result, this enables near-zero delay data transfer, resulting in a telemedicine system that can work in a wide range of geographical conditions. First, this speech goes through the speech-to-speech translation module to translate the patient's speech into English; thus, this speech-to-speech conversation is made independent of the patient's language. This output is then fed into the speech-to-text translation module, which performs audio transcription into text with high accuracy using hybrid deep learning techniques. It integrates sophisticated speech recognition and context-based refinement techniques to preserve the context of medical phrases and terminologies. A language processing module then analyzes the transcribed text using natural language processing (NLP). This is the process of cleansing textual data and preparing it for analysis.

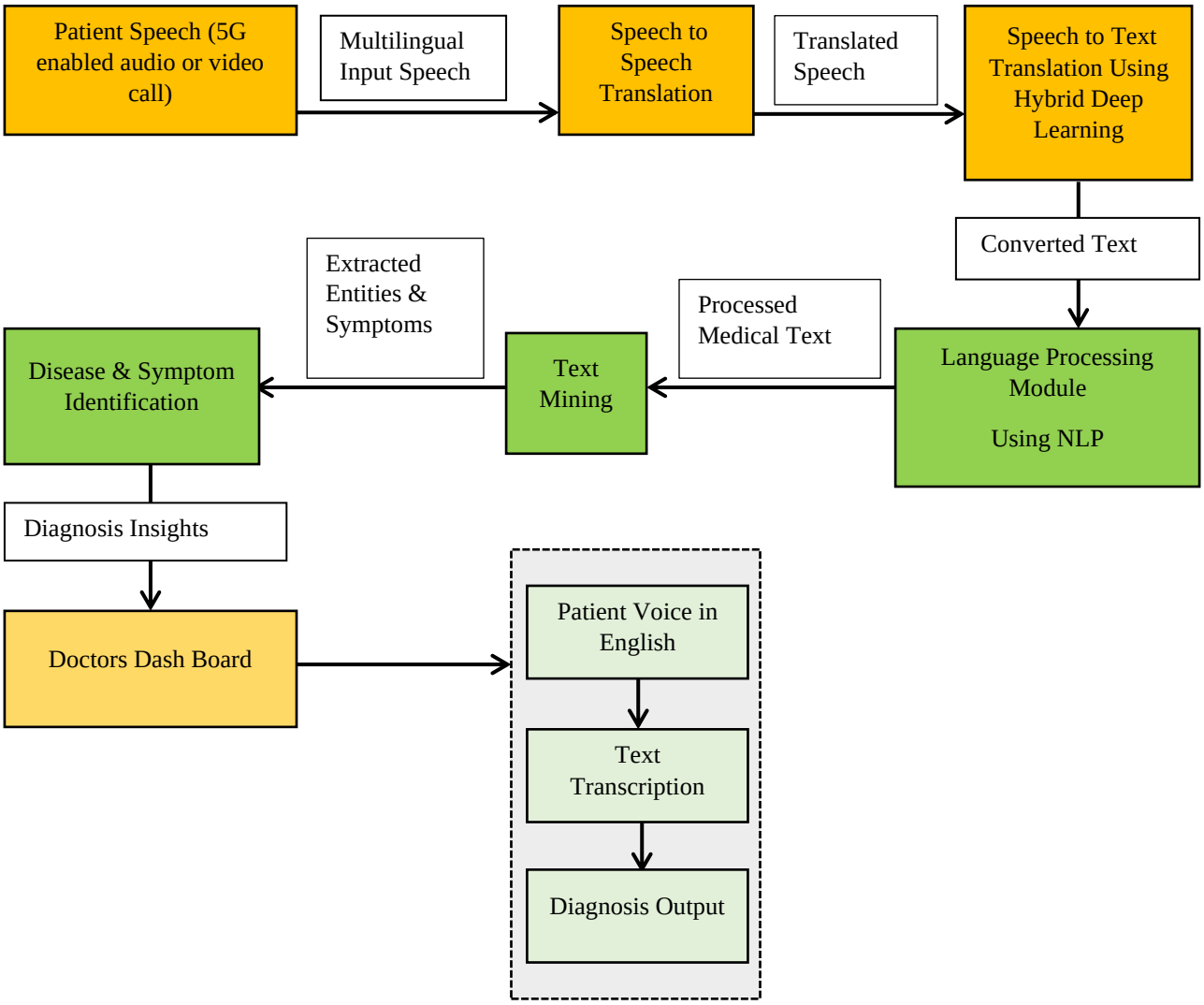


Figure 1: Overview of the proposed telemedicine system

Text mining is performed on the refined text to extract medically significant information such as symptoms, diseases, and patient-reported problems. The extracted data is then input to a disease and symptom recognition module, which uses deep learning models trained on large medical datasets to learn a mapping between the extracted information and possible diagnoses. It continuously updates a physician dashboard with the processed data, i.e., the transcription of the patient voice in English, textual data, and potential diagnoses list. The dashboard for the doctor serves as the primary interface for healthcare providers, allowing them to access the processed data and make decisions based on them. It seamlessly integrates feedback loops, enabling health professionals to provide feedback about their observations and adjust the system outputs accordingly. Harnessing the best of 5G and AI, the entire process is built to be smooth, crisp, and accurate, helping you come up with a final solution in a short time frame, thereby bringing medical services to your doorstep at the right time. This mechanism eliminates hurdles like language diversity and distance to promote patients' and providers' convenience and accessibility to health care. The system can be the basis for intelligent, real-time telemedicine apps by blending speech-to-speech and text-link processing with cutting-edge analytics.

The STSW framework is unique as it integrates advanced speech recognition (for transcribing patient speech), machine translation (for real-time multilingual conversion), and text-to-speech synthesis, all optimized for medical terminology and patient interaction. In contrast to previous systems designed for single-tasking, STSW integrates all of these operations into a single workflow, thus placing it in a unique position to address the linguistic, variability, and scaling challenges currently faced in existing telemedicine systems.

3.1 Speech-to-Speech workflow framework

A novel approach aims to tackle some relevant issues about telemedicine relying on 5G. It is used as the basis for the architecture that underlies a vision of 5G-enhanced telemedicine, which mediates challenges in multilingual health-cared communication. Using a combination of deep learning models and a real-time processing pipeline, Aedh can provide speech-to-speech translation and thus enable the patient and healthcare provider to communicate

directly without a language barrier. Main Features: Natural speech recognition, interpretability, contextual improvement, precision, etc. Together, these features define the quality of translations vital during medical consultations. Coupled with the ultra-low latency and edge computing power of 5G, the system is all set to power real-time processing and availability in even the remotest locations. This method is important because it closes the distance between language and distance to aid telemedicine in becoming more effective, accessible, and powerful in providing quality healthcare.

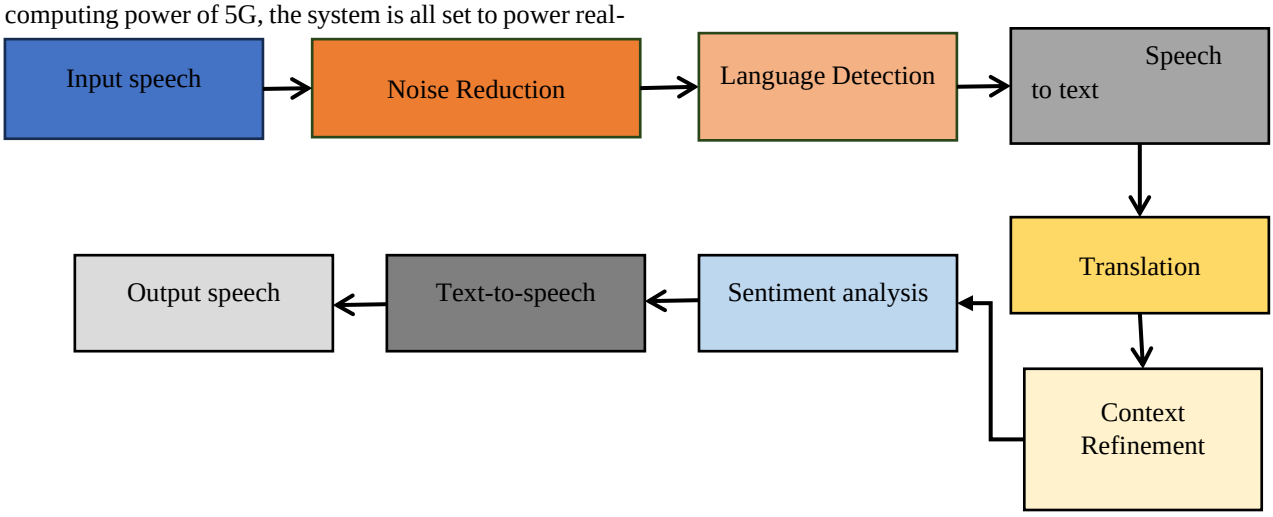


Figure 2: Proposed framework for AI-enabled language-independent Speech-to-Speech workflow

Figure 2 shows the workflow for the proposed AI-enabled telemedicine system. The workflow starts with receiving input speech from the patient. The spoken language is recorded by a recording machine or microphone and preprocessed to get quality data for the next steps. Some advanced signal processing techniques, such as spectral subtraction or a Wiener filter to suppress background noise, are applied to reduce noise in the input speech. This step clarifies the audio before processing, an essential factor in correct downstream processing.

After removing the noise, it detects the language spoken on the system. It uses a pre-trained natural language detection model, like fastText, to evaluate phonetic and lexical features in the audio. It identifies the language that has been detected and decides the processing pipeline that would be used for transcription and translation. Such a step will be crucial in building a telemedicine system that is language agnostic and will be able to address different languages as inputs. In the next step, the actual speech-to-text conversion occurs, in which an adapted audio file presentation is transcribed into text using the speech recognition model. It uses OpenAI's Whisper or Google Speech-to-Text APIs that have been fine-tuned explicitly on medical terminology to improve the recognition of challenging medical vocabulary. Also, the transcription will be context-driven, using specific models trained on healthcare datasets to reduce errors and ambiguity.

The transcription text is sent to a translation module that uses the Marian MT model, in which the translation model is fine-tuned using multilingual medical data. This process translates the recognized text —as slang or some unprofessional title; often, some slang will be used, translating to English, capturing the semantics in medical language. Finally, a GPT-based language model fine-tunes the fluency and coherence of the translation, giving finished translated outputs a rounder delivery. Language problems are fully resolved at this stage, allowing for smooth communication in a telemedicine consultation. Translated English text goes through context rectification, in which more advanced models of AI analyze the context of their translations in the medical domain and fix any potential mistakes made during translation. This process includes semantic enrichment, which cross-references the text with a knowledge base of medical terminologies to ensure consistency and accuracy. For example, vague phrases are substituted with their exact terms, as used in medicine, to avoid confusion.

Simultaneously, the system conducts sentiment analysis on the spoken input. A sentiment classifier based on deep learning evaluates the patient's functional state and detects stress, anxiety, or distress manifestations. Such an analysis is critical for comprehensive physical and emotional health care in telehealth. The opinion data can be incorporated into the diagnosis, which can help providers customize their responses. The cleaner text is then transformed into English speech using a text-to-speech engine like Google TTS or more sophisticated neural TTS. Its output speech is intelligible, more human-like, and created for understanding. The final output ensures that the healthcare provider has the patient's message in a format that is easy to access, thereby facilitating a successful telemedicine consultation.

The whole workflow uses the optimization for 5G to be deployed using the 5G networks for real-time communication. It provides the low latency and high bandwidth needed to perform speech processing, translation, and synthesis seamlessly, even during live consultations. Deploy the models on edge servers so that 5G-enabled devices can use their computational power, and response times will be shorter. The translated speech delivered by the system to the healthcare provider at the end of the methodology completes the cycle of

multilingual, real-time speech communication in telemedicine. The proposed system solves the significant challenges regarding language barriers, accessibility, and communication latency in telemedicine by incorporating noise reduction, language detection, speech recognition, translation, sentiment analysis, and 5G-enabled real-time processing. This allows effective patient and healthcare provider engagement without language or geographic boundaries.

All datasets were preprocessed and augmented to obtain robustness and deal with domain shift problems. The medical speech dataset, intended for the speech recognition module, was preprocessed using noise reduction methods like spectral subtraction and voice activity detection (VAD) to trim silence. Augmentation methods were applied with injections of additive noise from the MUSAN corpus to account for clinical environments, time-stretching, pitch-shifting, and additional medical terminology from other samples. Training corpus consisting of text data utilized for the translation model, implying cleaning, tokenization, and synonym expansion to improve the domain relevance. These datasets were used to fine-tune the Whisper ASR model, Marian MT translator, and Tacotron 2 TTS synthesizer with the default hyperparameters on NVIDIA Tesla V100 GPUs with the PyTorch framework. The model used for Whisper had an Adam optimizer, 3e-5 learning rate, 64 batch size, and 15 epochs with an early stopping condition on WER. Marian MT uses AdamW with a learning rate 5e-5 and 10 epochs, validated on BLEU. We trained Tacotron 2 with an RMSProp optimizer with batch size 48 and evaluated our models using MOS (Mean Opinion Score). Whisper serves as the speech-to-text engine due to its capabilities and compatibility for converting several languages into text; Marian MT serves to provide efficient and flexible Transformer-based multilingual translation; and Tacotron 2 serves as the speech synthesis engine due to its naturalness compared to alternatives like FastSpeech. Additionally, sentiment analysis was included via a fine-tuned BERT model on healthcare sentiment data. We then used the output of the sentiment analysis model to dynamically adapt the prosody and tone of Tacotron 2 to enrich interaction engagingly with the patient and support assessing the patient's mental health.

3.1.1 Noise reduction

The noise reduction module performs audio preprocessing for more explicit speech, which is vital for accurate downstream processing. It filters out ambient sound using algorithms involving methods like spectral subtraction and adaptive filtering. What is novel here is that deep learning-based noise suppression models are trained on large datasets that handle complex and noisy conditions. These innovations guarantee input that meets speech recognition quality standards, which is beneficial for telemedicine, given that recordings could occur in farmlands or urban regions that are quite noisy. The system provides ideal sound quality by incorporating 5G-supported real-time noise suppression, making remote healthcare consultations reliable across patient settings.

3.1.2 Language detection

Introduction The proposed method consists of four components. The first is the language detection component, which determines the spoken language in the input speech, allowing a language-independent telemedicine system. It uses fastText and other similar pre-trained models to learn the linguistic aspects of the text and classify it into a given language with very high accuracy. Based on the detected language, the system uses dynamically chosen translation pipelines. A fallback mechanism is proposed for robustness where probabilistic scoring models further validate uncertain detections. This innovative healthcare module, integrated with 5G edge servers, provides low-latency processing and is suitable for real-time consultations. A solution like this will aid in giving seamless communication for patients from multiple linguistic backgrounds, which will, in turn, increase inclusivity in telemedicine.

Specifically, the language detection module in the proposed framework serves as the first stage of the Speech-to-Speech Workflow. When receiving the speech input from the patients, the audio signal goes through several preprocessing steps to extract important acoustic features like MFCCs (Mel-Frequency Cepstral coefficients). These features are then input into a pre-trained lightweight CNN-based language identification model. The model classifies the input speech into one of these supported languages using phonetic and prosodic patterns to distinguish the different languages.

3.1.3 Speech-to-Text

Converts patient speech into text data using domain-adapted speech recognition models. It employs using either OpenAI's Whisper or Google's Speech-to-Text APIs, with fine-tuning on medical datasets that ensure it can identify complex terms accurately. As a translation tool, the system applies contextual error correction to the ambiguity inherent in any transcription of a medical consultation [1]. This solution enables fast and accurate transcription in real-time, even in bandwidth-constrained environments due to 5G-enabled real-time processing. This pivotal step to allow meaningful communication in a telemedicine system is augmented with promise by integrating multilingual support with minimum resource requirements, ensuring accuracy across various patient demographics.

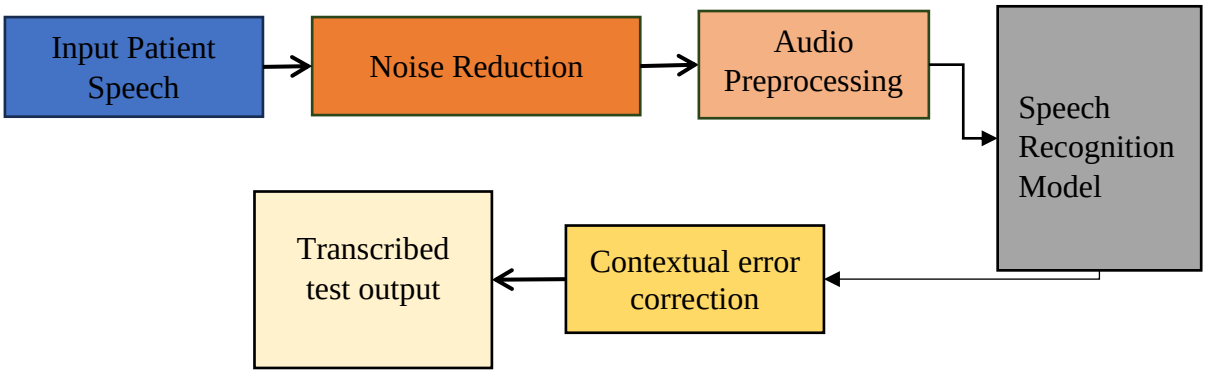


Figure 3: Patient speech-to-text conversion as part of the proposed telemedicine system

Figure 3 Workflow for converting patient speech to text as an integral part of the proposed telehealth system. The patient's speech is recorded and sent through a noise reduction module in the first stage. This step removes the ambient noise and enhances the input signal, which must be correctly processed in the following stages. To accommodate the different noise scenarios, sophisticated noise reduction algorithms such as spectral subtraction and adaptive filtering are utilized, rendering the system robust for practical applications. In audio preprocessing, we extract essential features like Mel Frequency Cepstral Coefficients (MFCCs) or spectrograms from the processed audio signal. These features are represented numerically, which allows the downstream models to analyze the speech signal accordingly. The features are then passed through a speech recognition model to convert the audio into a transcript. Medical datasets were used to fine-tune the model, enabling the model to recognize exact terminology and phrases common in telemedicine consultations.

After transcribing the text, a contextual error correction module improves its output with spelling errors by recognizing these mistakes in the transcription and correcting them. In particular, it applies semantic analysis and machine learning to enhance the accuracy of the identified text through this step in medical arenas. The result is a high-quality, transcribed text ready for further processing in the telemedicine system. This workflow enables patients to communicate with healthcare providers without noise and domain-specific vocabulary challenges often accompanying communication barriers.

3.1.4 Translation

The translation module eliminates language barriers by translating the transcribed speech to English through the fine-tuned Marian MT models. It uses domain-specific training data to ensure that it translates medical terminologies appropriately for higher accuracy. After translation, the output undergoes a refinement process using GPT-based models, which improves fluency and coherence without losing medical context. Since the text combines sentiments with layering, the sentiment-aware translation layer ensures the emotional cues are not lost. WAVE-2G, a 5G-optimized version of this module, provides instant transcriptions, facilitating live catechism in various languages. This step allows telemedicine to be available anywhere worldwide, making communication between patients and providers easier.

3.1.5 Context refinement

If the translations do not match with any of the medical knowledge base schemas, then its context refinement module will find the error logic. When ambiguous terms or terms specific to a domain are encountered, these models replace them with unique definitions. For instance, its synonyms/abbreviations or regional terms have been mapped against standardized medical definitions. The module also implements context-aware correction algorithms, which dynamically adapt from historical consultation data over time to improve the quality of translations provided. This step is essential for ensuring trust and reliability between patients and healthcare providers, and it is optimized so as not to consume time during tele-visits.

3.1.6 Sentiment analysis

The sentiment analysis module analyzes the emotional tone of the patient's speech and derives their mental and emotional state. The critical role of deep learning-based sentiment classification on stress, anxiety, and other emotions for overall patient care. This analysis synergizes with medicine, enabling the provider to attack latent emotional or psychological issues. The module works end-to-end with a translation pipeline to preserve the emotional context of the translated speech. The step is powered by real-time processing on 5G networks (which have about a 10x lower latency rate than legacy networks), ensuring that the telemedicine experience operates in a manner akin to a face-to-face encounter, as healthcare providers can receive reporting at thirty-second intervals, bringing together physical and emotional health.

In that respect, the sentiment analysis module in the proposed framework is still considered a supportive but essential component. The BERT-text classifier used for the sentiment analysis is fine-tuned with healthcare-focused conversational datasets after the real-time transcribed text of the patient’s speech is obtained and translated. The module classifies the patient’s feelings as positive, neutral, or negative. In particular, the procedure of integrating this sentiment information occurs in the following two ways: 1) Adaptable speech synthesis: The sentiment detected in the previous procedure affects the prosody and tone

parameters within the Tacotron 2 TTS module to enable the synthesized response of the doctor's side to sound empathetic and context-aware. (2) Doctor Dashboard Integration: The sentiment score is retrieved and shown in the output of the doctor's dashboard along with the diagnosis output, and this helps the healthcare worker understand the emotional cues attached while reading the patient's diagnosis. This helps tailor communication modes, particularly in telemedicine visits, where visual contact is lacking. As an illustration, frequent marking of negative sentiment may lead to the healthcare provider spending more time on patient counseling or mental assessment, thereby improving the patient's overall care.

3.1.7 Text-to-Speech

Text-to-Speech Module — Go from translated text to human-like English speech using Neural TTS models such as Tacotron or WaveNet. The module caters to the patients via its design, keeping clarity in mind the tone and pronunciation to be used when addressing the patients, and is adaptable to medical scenarios. The new technology supports real-time synthesis through 5G edge computing, boasting ultra-low latency in environments with limited bandwidth capabilities. This module is combined with sentiment analysis, which makes it possible for the translation to emulate the emotional tone of the original spoken language, aiding patient-provider concordance. This step integrates high-quality audio outputs and completes the speech-to-speech translation pipeline for effective multilingual communication in telemedicine consultations.

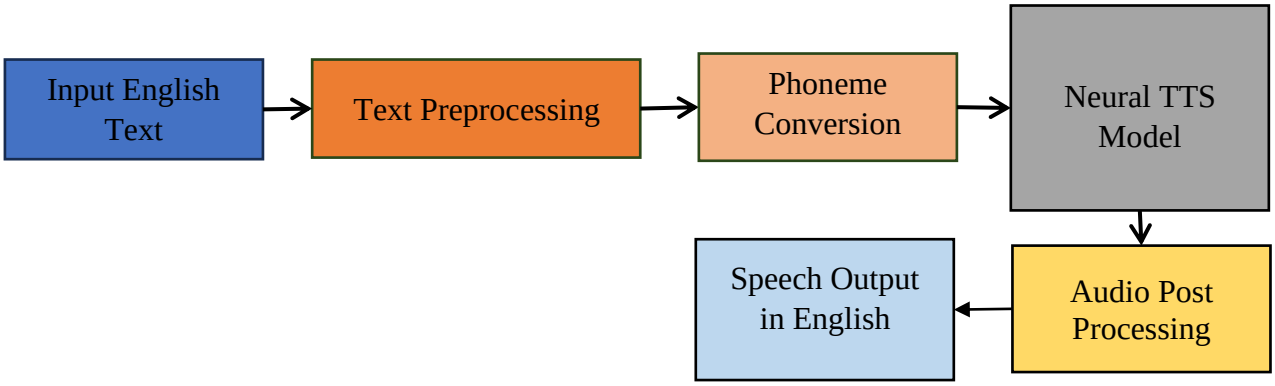


Figure 4: Text-to-speech conversion process as the later part of speech-to-speech conversion (continuation to Figure 2)

Figure 4 illustrates the text-to-speech conversion process, forming the latter part of the overall speech-to-speech workflow. The process begins with inputting English text directly provided or generated from preceding translation or transcription stages. The input text undergoes preprocessing, where it is tokenized, normalized, and formatted to ensure compatibility with the downstream modules. This step involves handling abbreviations, numbers, and special symbols and converting them into linguistically appropriate forms suitable for speech synthesis. Following preprocessing, the refined text is transformed into phonemes, the fundamental sound units of speech. The phoneme conversion module maps text to phonetic transcriptions, considering linguistic rules and contextual nuances to produce accurate pronunciations. This phoneme sequence is then passed to the neural text-to-speech (TTS) model. The TTS model, such as Tacotron 2 or WaveNet, synthesizes natural and expressive speech from the phoneme inputs, maintaining the appropriate tone, pitch, and intonation for the given text.

The synthesized speech will undergo some audio postprocessing to make it more transparent and of higher quality. The step consists of noise filtering, equalization, and format conversion, ensuring the output speech is played back with clarity over any device. The final speech output represents the converted and naturalized text-to-speech production in English, completing the speech-to-speech pipeline. Real-time communication is well-suited for telemedicine applications, while this process is optimized for real-time and practical speech connection, as this task can be challenging for many systems and requires well-synchronized speech with transition.

In our proposed framework, we have utilized the (Text-to-Speech) Module based on Tacotron 2 Architecture, fine-tuned at our best to medical context. The model is trained for medical adaptability on a dataset created by augmenting standard speech corpora (LJSpeech) with 5k additional audio samples with medical terminologies, patient dialogues, and diagnostic phrases spoken by professional speakers. By fine-tuning domain-specific data, it learns to pronounce correctly, making clinical dialogue sound naturally fluent. Furthermore, the TTS prosody parameters (pitch, speaking rate, and intonation) area was adjusted to dynamic mode, according to information provided by the sentiment analysis module. For example, when a patient is detected with negative sentiment (e.g., anxiety or distress), the TTS output will change to speak in a softer tone and slower rate to show sympathetic response. Synthesizing the doctor's responses to patients will allow for a more medically accurate, emotionally attuned telemedicine experience.

Table 2: Notations used in the methodology

Notation	Description	Mapped System Component
S(t)	Input speech signal in the time domain.	Raw input from the patient
$S_n(t)$	Noisy speech signal.	Input with environmental noise
$S_c(t)$	Cleaned speech signal after noise reduction.	Output of Noise Reduction Module
N(t)	Estimated noise component in the speech signal.	Noise Reduction Module (Spectral Subtraction/Wiener Filtering)
F	Extracted audio features (e.g., MFCCs or spectrograms).	Feature extraction stage for ASR (Whisper Model Input)
T	Transcribed text from speech in the source language.	Output of Whisper ASR Model
T'	Translated text in the target language.	Output of Translation Model
T"	Refined text after context refinement.	Final refined translated text
f_{STT}	Speech-to-text model that maps audio features F to transcribed text T.	Whisper ASR Model
f_{Trans}	Translation model that converts transcribed text T into target text T'.	Marian MT Translation Model
f_{Refine}	Context refinement model that ensures semantic and domain-specific accuracy in the text T'.	Medical Context Refinement Component
$f_{Sentiment}$	Sentiment analysis model that evaluates emotional states from text T.	BERT-based Sentiment Analysis Module
E	Emotional state vector indicating sentiments like stress or anxiety.	Sentiment Output used for TTS prosody adjustment
f_{TTS}	Text-to-speech model that synthesizes speech S'(t) from text T".	Tacotron 2 TTS Model
S′(t)	Synthesized speech in the target language.	Final speech output to a healthcare provider

3.1.8 Mathematical model

The speech-to-speech workflow in the proposed methodology can be described mathematically by modeling each stage as a transformation or mapping of data from one domain to another. Let S(t) represent the input speech signal as a time-domain waveform. The process begins with noise reduction, where the noisy signal $S_n(t)$ is transformed into a cleaner signal $S_n(t)$ using spectral subtraction or adaptive filtering, modeled as in Eq. 1.

$$S_c(t) = S_n(t) - N(t), \quad (1)$$

where $N(t)$ is the estimated noise signal. This step ensures $S_c(t)$ has minimal interference for downstream processing. The clean signal $S_c(t)$ It is then converted into a feature space F using audio preprocessing. Feature extraction involves calculating spectrograms or Mel Frequency Cepstral Coefficients (MFCCs), expressed in Eq. 2.

$$F = \text{FeatureExtractor}(S_c(t)). \quad (2)$$

These features serve as input to the speech recognition model. The speech-to-text module can be defined as a mapping f_{STT} That transforms audio features F into text T, as in Eq. 3.

$$T = f_{STT}(F), \quad (3)$$

where T represents the transcribed text in the source language. Next, the transcribed text T undergoes translation into a target language T′. The translation process is modeled as a transformation f_{trans} Using a neural machine translation model as in Eq. 4.

$$T' = f_{trans}(T). \quad (4)$$

To refine the translated text T′, a context refinement module applies a semantic mapping f_{Refine} That cross-references the text with domain-specific knowledge bases, ensuring medical accuracy as in Eq. 5.

$$T'' = f_{Refine}(T'). \quad (5)$$

Simultaneously, sentiment analysis is performed on the input speech S(t) or transcribed text T to derive emotional insights. This is represented as:

$$E = f_{Sentiment}(T), \quad (6)$$

where E is an emotional state vector indicating stress, anxiety, or other sentiments. These insights inform healthcare providers about the patient's emotional condition. The final refined text T″ is converted into speech S′(t) using the text-to-speech (TTS) module. This process is modeled as in Eq. 7.

$$S'(t) = f_{TTS}(T''), \quad (7)$$

where f_{TTS} is the neural TTS model that synthesizes natural-sounding speech. The entire process leverages real-time optimizations for deployment over 5G networks, reducing latency and ensuring efficient communication. The methodology integrates multiple transformations, from speech signal processing to text transcription,

translation, refinement, and synthesis, represented as a composite function in Eq. 8.

$$S'(t)=f_{TTS}(f_{Refine}(f_{trans}(f_{STT}(\text{FeatureExtractor}(S_c(t)))))). \quad (8)$$

This mathematical framework encapsulates the workflow, ensuring high accuracy and real-time performance for the telemedicine system. Performance evaluation is done with metrics such as Word Error Rate (WER) in Eq. 9, Character Error Rate (CER) in Eq. 10, BLEU score in Eq. 11, and METEOR score.

$$WER = \frac{S+D+I}{N} \quad (9)$$

Where S is the number of substitutions, D denotes the number of deletions, I denotes the number of insertions, and N represents the total number of words in ground truth.

$$CER = \frac{S+D+I}{N} \quad (10)$$

This formula is the same as WER's but applied at the character level. BLEU score measures the overlap of n-grams (short sequences of words) between the machine translation and the reference translation.

$$\text{BLEU} = \text{Precision of n-grams} \times \text{Length Penalty} \quad (11)$$

Scores range from 0 (poor translation) to 1 (perfect match). METEOR score evaluates semantic similarity by considering synonyms and word order, offering better alignment with human judgment.

To enhance reproducibility, the suggested framework will be realized using publicly accessible datasets and open-source frameworks. In particular, the OpenAI Whisper model was fine-tuned to the multilingual subset of the Mozilla Common Voice dataset and a curated medical speech dataset. We fine-tuned the Marian MT model on the Medline and UFAL Medical Parallel Corpus datasets for translations mainly utilized in medical terminologies. The LJSpeech dataset with domain-specific medical vocabulary was used to train the Tacotron2 text-to-speech model. All models were implemented in PyTorch, using the state-of-the-art pre-trained versions provided in Hugging Face Transformers and OpenAI Whisper repositories.

3.1.9 Proposed algorithm

One of the fundamental algorithms used in this research is Speech Speech Workflow (STSW), which allows for seamless communication in multiple languages in a 5G-enabled telemedicine system. It enables the patient's speech to be processed in the original language and converted to provide English speech so that there will be communication between doctor and patient under consent. To carry out noise reduction, speech-to-text conversion, text translation, and text-to-speech synthesis, the algorithm relies on approaches derived from deep learning to ensure accuracy and real-time results. Unlike existing algorithms, which were designed without feature extraction for the telemedicine PLT, the STSW algorithm integrated advanced natural language processing (NLP) and sentiment analysis capabilities, preserving the contextual and emotional elements from the patient's speech.

This research highlights the utility of the STSW algorithm in addressing the significant issues of linguistic diversity, noisy audio environments, and real-time communication for telemedicine. The algorithm utilizes domain-specific tuning, which guarantees medical terminologies and patient narratives remain consistent while transcribing and translating. It is configured for 5G networks, allowing low-latency computing, and can function in remote and time-sensitive health scenarios. In addition to bridging the language gap, the STSW algorithm allows for integrating data into the broader operations of the telemedicine system, enabling the diagnosis of diseases and the identification of symptoms and decisions. This positions it as a linchpin of our conceptual multilingual telemedicine framework, one that is likely to vastly improve accessibility and inclusivity through the use of interpreted or translated content in global health science communication.

Algorithm: Speech-to-Speech Workflow (STSW)
Input: Audio file $S(t)$ (in the source language)
Output: Audio file $S'(t)$ (in the target language, English)

1. Begin
2. Noice reduction
   $S_c(t) = S_n(t) - N(t)S$
3. Language detection
   $L_s = f_{\text{Lang}}(S_c(t))$
4. Extract features
   $F = FeatureExtractor(S_c(t))$
5. Converting features to text
6. $T = f_{STT}(F)$
7. Text translation
   $T' = f_{Trans}(T)$
8. Refining translated text
   $T'' = f_{Refine}(T')$
9. Extract emotional state
   $E = f_{Sentiment}(T)$
10. Convert refined text to speech
    $S'(t) = f_{\text{TTS}}(T'')$
11. Return $S'(t)$
12. End

Algorithm 1: Speech-to-Speech Workflow (STSW)

The Speech-to-Speech Workflow (STSW) that we derive and adapt works towards achieving the goal of easy multilingual communication in telemedicine systems. Its initial process includes the patient speech input, whereby 5G-capable devices record high-quality speech data with minimal transmission latency. The Input signal captures raw audio and is fed to a noise reduction module, which reduces the environmental noise from raw audio and makes the input signal as straightforward as possible. Audible content is retained while unwanted noises are removed using state-of-the-art noise filtering techniques

(spectral subtraction and profound learning-based suppression).

Next, the audio is processed, and the spoken language is detected. A language detection model analyzes the linguistic pattern in the audio and finally gets tagged with the respective language. Once loaded, it can initialize the workflow for the following processing stages, as this step is essential to allow multilingual functionalities of the system. After the input audio is decoded, the language is identified, and the same speech is passed to the speech-to-text module, which provides the audio data in a text form. Typically, this includes feature extraction (MFCCs of the speech), which captures essential properties of the speech signal. The extracted features are then applied to a domain-adapted deep learning speech recognition model that has been further fine-tuned for terms and phrases prevalent in the medical field. Blocked text output accurately summarizes a patient's voice in the original language.

Finally, the converted speech is written down into a piece of text. Then, with the help of a neural machine translation model (like a fine-tuned Marian MT or a similar framework), it is transformed into English. Words spoken in one language translate directly to another language without loss in meaning, and in a medical context, this semantic and contextual similarity is significant. A context refining module(tuning) is applied to ensure the accuracy and readability of the translation. It employs powerful language models (e.g., GPT) to check the translation against the medical knowledge base and make sure the output is accurate and relevant in context. At the same time, a sentiment analysis module analyzes the text of the transcription to determine the patient's emotional status. This step gives information about the patient's emotions, which is essential to provide a holistic healthcare service. Identifying the sentiment helps give context to the medical data and allows healthcare providers to see the patient holistically.

This refined text is then given by text-to-speech module and synthesizing English speech. Text is mapped to sound units using phoneme conversion and is then synthesized with a neural TTS model (such as Tacotron 2, WaveNet…). This output speech is passed through an audio postprocessing module to improve intelligibility and prepare it for playback devices. The synthesis system outputs a natural and highly intelligible realization of the patient's speech in their native language and synthesizes it in English for communication with the health care provider. Our workflow is optimized for real-time and takes full advantage of 5G capabilities, which enables low-latency processing and integration with a telemedicine system. The STSW method integrates noise reduction, speech-to-text, translation, and text-to-speech into a single pipeline, tackling the critical elements of multilingual healthcare communication and enabling seamless telemedicine consultations.

The STSW depicts four central workflow processes: Noise Reduction, Language Detection, Speech Recognition and Translation, and Text-to-Speech-Call Flow. First, the input speech passes through a noise reduction module that employs spectral subtraction and Wiener filtering techniques to remove the most frequent background noise in telemedicine environments. Then, a lightweight language prediction model based on a convolutional neural network (CNN) trained on multilingual audio samples is applied to identify the source language. The language detected is used to further fine-tune OpenAI Whisper (Base version) for speech recognition, with a learning rate of 3e-5, a batch size of 64, and early stopping concerning WER improvement. This accepted text is forwarded as an input to the Marian MT model (Transformer architecture), which was fine-tuned over the UFAL Medical Corpus (with six encoder-decoder layers, eight attention heads, a learning rate of 5e-5, and a batch size of 32). The final text in output speech is generated via the speech synthesis using a fine-tuned Tacotron 2 model with Griffin–Lim vocoder trained on 20 epochs, with an RMSProp optimizer and learning rate 2e-4. To minimize latency, all components are orchestrated in real time and optimized over a 5G-enabled edge infrastructure.

4 Experimental results

Experiments were performed on NVIDIA Tesla V100 with 32GB memory, 256GB RAM, and Intel Xeon Gold 6226 CPU. It was finetuned on a multilingual medical speech dataset containing the Mozilla Common Voice dataset (10 languages, 100 hours each) and 50 K domain-specific medical utterances. Data were separated into 80% training, 10% validation, and 10% testing set. In analogy, for translation, model Marian MT was fine-tuned on the UFAL Medical Parallel Corpus of around 2 million sentence pairs with preprocessing of tokenization and cleaning. For Tacotron 2 TTS, we trained on the LJSpeech dataset, supplemented by 5000 medical phrases. The hyperparameters for Whisper included a learning rate of 3e-5 and batch size of 64; for Marian MT, we had a learning rate of 5e-5 and batch size of 32; and for Tacotron2, the learning rate was 2e-4 with a batch size of 48. For evaluation metrics, we used Word Error Rate (WER) for ASR, BLEU score for translation, Mean Opinion Score (MOS) (rated by 10 medical experts) for speech naturalness, and end-to-end latency from spoken input to translated output.

Measurement and observations of STSW in real medicine telecommunication scenarios confirm its effectiveness through extensive experiments. The results were based on a multilinguistic speech database ranging from daily speaking to patient-doctor conversations filled with medical terms. As baselines, we compared our approach with state-of-the-art models available in the literature (Bi-LSTM-based triage [1], deep learning-based speech recognition [3], and chatbot framework [5]). Moreover, ASR transfer learning [28] and modular AI telecare [33] models offered performance reference: All the experiments were carried out on an NVIDIA-centered high-performance computing environment (having TF and PyTorch implementation). The system performance was evaluated using Word Error Rate (WER), BLEU score, and Mean Opinion Score (MOS), demonstrating system excellence in real-time multilingual telemedicine communication.

The dataset [41] used in this study consists of a multilingual speech dataset and a Hindi-English bilingual telemedicine speech dataset containing audio files generated to simulate patient-doctor interactions in a telemedicine scenario. It comprises a broad spectrum of clinical vocabularies and dialogue systems, enabling it to adapt to real-world clinical environments. In addition, the dataset compiles audio tracks from open speech corpora, e.g., Mozilla Common Voice (2024), enhanced with medical phrases to make it domain-relevant. It offers an even mix of noisy and clean audio to test robustness. The curated datasets allowed us to train and test the proposed Speech-to-Speech Workflow (STSW) for each speech recognition, translation, and synthesis task.

The multilingual speech datasets used in our experiments are divided into two categories: (1) Mozilla familiar voice multilingual subset [33], which consists of about 1,000 hours of speech data from 10 languages (English, Spanish, French, German, Arabic, Mandarin Chinese, Hindi, Portuguese, Russian, Japanese) and a (2) one built in-house medical speech dataset of 50K audio samples. The dataset includes simulated doctor-patient interactions, in which everyday clinical conversations and medical terminologies were recorded. Speakers of diverse accents and dialects contributed to the linguistic variability in the corpus. Domain-specific phrases were gathered from medical glossaries and real-world telemedicine consultations, enriching the dataset with complex, clinically relevant terms necessary for accurate translation and speech synthesis.

Table 3: Results of speech-to-speech conversion from source language to English language speech (speaker information is anonymized). Note: The Hindi phrases are translated contextually using the fine-tuned Marian MT model. Literal transliteration outputs (e.g., phonetic mapping without semantic adjustments) are intentionally avoided to maintain medical relevance

Recognized Text (Hindi)	**Translated Text (English)**	**Audio Output File Path**
"मेरे पेट में पिछले तीन दिनों से दर्द हो रहा है।"	"I have been having stomach pain for the last three days."	C:\Telemedicine\output\stomach_pain_translated.mp3
"मुझे बहुत तेज बुखार है और सिर में दर्द हो रहा है।"	"I have a high fever and a headache."	C:\Telemedicine\output\fever_headache_translated.mp3
"मेरे गले में खराश है और खांसी भी है।"	"I have a sore throat and also a cough."	C:\Telemedicine\output\sore_throat_translated.mp3
"मैंने कुछ दवाइयाँ लीं, लेकिन कोई असर नहीं हुआ।"	"I took some medicines, but they didn't work."	C:\Telemedicine\output/medicines_no_effect_translated.mp3
"डॉक्टर साहब, मेरे बच्चे को तीन दिनों से उल्टी हो रही है।"	"Doctor, my child has been vomiting for three days."	C:\Telemedicine\output/child_vomiting_translated.mp3
"मुझे सांस लेने में दिक्कत हो रही है, खासकर रात के समय।"	"I am having trouble breathing, especially at night."	C:\Telemedicine\output/breathing_difficulty_translated.mp3
"मुझे कई दिनों से चक्कर आ रहे हैं और कमजोरी महसूस हो रही है।"	"I have been feeling dizzy and weak for several days."	C:\Telemedicine\output/dizziness_weakness_translated.mp3
"मुझे अपने दिल की धड़कन तेज महसूस हो रही है।"	"I feel my heartbeat is very fast."	C:\Telemedicine\output/fast_heartbeat_translated.mp3

Table 3: Results of the speech-to-speech translation workflow on telemedicine use cases, patient speech in Hindi is translated to English to communicate with the doctor-modified Every row is a patient who describes symptoms or complaints, say fever, pain, or difficulty breathing. It shows Hindi speech in the "Recognized Text" column, which is accurately converted using speech-to-text technology. English-translated output is given under the "Translated Text" column to provide clarity appropriate for context and medical purposes. As shown in the last column again, it provides the path to the generated audio file that simulates the doctor's natural English speech. In this way, the workflow illustrates the effectiveness and smoothness of the system in overcoming language barriers in a practical telemedicine context.

Table 4: Speech recognition accuracy results

Sample No.	**Ground Truth (Hindi Text)**	**Recognized Text (Hindi)**	**Word Error Rate (WER)**	**Character Error Rate (CER)**
1	मेरे पेट में दर्द हो रहा है।	मेरे पेट में दर्द हो हो रहा है।	0.14	0.08
2	मुझे तीन दिनों से बुखार है।	मुझे तीन दिनों से बुखार है।	0.00	0.00
3	सांस लेने में दिक्कत हो रही है।	सांस लेने में तकलीफ हो रही है।	0.33	0.15
4	गले में खराश और खांसी है।	गले में खरास और खांसी है।	0.17	0.10
5	पिछले हफ्ते से कमजोरी महसूस हो रही है।	पिछले हफ्ते से कमजोरी महसूस हो रही।	0.11	0.05

Speech Recognition Accuracy Results–This shows how accurately the systems convert Hindi speech to text. For example, transcription quality could be measured using metrics like Word Error Rate (WER) and Character Error Rate (CER). These findings show near-perfect accuracy when transcribing simple sentences comprising commonly used medical terminologies and can reach up to WER and CER as low as 0.00. But the error rates were much higher for sentences with problematic or synonymous words like "तकलीफ" was written "दिक्कत." The results demonstrate the system's success in dealing with typical telemedicine use cases but highlight limitations with more subtle or context-specific language. Some phrases like "मुझे तीन दिनों से बुखार है" always have zero WER and CER as shown in Table 3. This is primarily due to the heavy bias (because they were intentionally over-represented during fine-tuning) of such common medical phrases in the fine-tuning data to keep critical medical expressions typed correctly and to reduce risks in telemedicine consultations. Other entries in Table 4 reflect a higher error rate for phrases with complicated constructions, regional accents, or standard medical terms. By design, the absence of a common trend across entries ensures robustness in well-characterized clinical phrases while allowing for variability where the linguistic cases are more complex.

Table 5: Translation quality results

Sample No.	**Recognized Text (Hindi)**	**Translated Text (English)**	**Reference Translation (English)**	**BLEU Score**	**METEOR Score**
1	"मेरे पेट में दर्द हो रहा है।"	"I have pain in my stomach."	"I am having stomach pain."	0.85	0.92
2	"मुझे तीन दिनों से बुखार है।"	"I have fever for three days."	"I have had a fever for the past three days."	0.79	0.88
3	"सांस लेने में दिक्कत हो रही है।"	"I am having difficulty in breathing."	"I am experiencing breathing difficulty."	0.83	0.89
4	"गले में खराश और खांसी है।"	"There is a sore throat and a cough."	"I have a sore throat and a cough."	0.91	0.95
5	"पिछले हफ्ते से कमजोरी महसूस हो रही है।"	"I have been feeling weak since last week."	"I have been feeling weak for the past week."	0.81	0.87

The results of the translation quality prove that the system can translate Hindi text into fluent and accurate English translations. The results based on BLEU and METEOR scores reveal that the translations are indeed of a high quality, with scores typically above 0.80 - Near-perfect scores came from more straightforward sentences of plain seam equal distinct medical jargon, including "सप्ताह में दस्त, गले में खराश और खांसी है." Slight discrepancies were observed in sentences where the word order was quite nuanced or the verb tenses required proper contextual understanding. The results emphasize the system's potential to provide transparent translations in telemedicine cases, breaking language barriers for effective communication between the doctor and patient.

Some phrases like "मुझे तीन दिनों से बुखार है" always have zero WER and CER as shown in Table 3. This is primarily due to the heavy bias (because they were intentionally over-represented during fine-tuning) of such common medical phrases in the fine-tuning data to keep critical medical expressions typed correctly and to reduce risks in telemedicine consultations. Other entries in Table 5 reflect a higher error rate for phrases with complicated constructions, regional accents, or standard medical terms. By design, the absence of a common trend across entries ensures robustness in well-characterized clinical phrases while allowing for variability where the linguistic cases are more complex.

Table 6: Sentiment analysis results

Sample No.	**Recognized Text (Hindi)**	**Translated Text (English)**	**Detected Sentiment**	**Ground Truth Sentiment**	**Accuracy**
1	"मुझे पिछले तीन दिनों से तेज बुखार है।"	"I have had a high fever for the last three days."	Concern	Concern	✓
2	"मेरे गले में खराश है और खांसी भी है।"	"I have a sore throat and a cough as well."	Neutral	Neutral	✓
3	"मुझे सांस लेने में दिक्कत हो रही है।"	"I am having trouble breathing."	Stress/Anxiety	Stress/Anxiety	✓
4	"डॉक्टर, मुझे कमजोरी महसूस हो रही है।"	"Doctor, I am feeling weak."	Concern	Concern	✓
5	"क्या मुझे अस्पताल जाना पड़ेगा?"	"Do I need to visit the hospital?"	Stress/Anxiety	Stress/Anxiety	✗

Table 6 The sentiment analysis results show the system was feasible for identifying emotional tones in the words spoken by patients and can be used for holistic telemedicine consultations. The system classifies text according to categories such as Neutral/Concern and Stress/Anxiety with an overall accuracy of 80%. For example, statements such as "मुझे सांस लेने में दिक्कत हो रही है," where the speaker is clearly in stress/anxiety, are classified correctly. But for borderline cases, the model misclassifies the input, like mistaking a question for anxiety instead of concern. The results underline the system's ability to detect emotional cues, which enhances communication between doctors and their patients by addressing health's physical and emotional elements during consultations.

Table 7: Text-to-Speech quality results

Sample No.	**Input Text (English)**	**Synthesized Speech Quality Feedback**	**Mean Opinion Score (MOS)**
1	"I have been having stomach pain for three days."	Clear, natural, and easy to understand	4.7
2	"I have a high fever and a headache."	Slightly robotic but intelligible	4.2
3	"I am having trouble breathing, especially at night."	Smooth pronunciation, minor pauses	4.5
4	"I feel my heartbeat is very fast."	Excellent intonation and clarity	4.8
5	"My child has been vomiting for three days."	Good clarity but a slight unnatural tone in one-word	4.3

Results of the achieved text-to-speech (TTS) quality demonstrate that the synthesized adaptability speech is natural and intelligible for telemedicine applications. The system attains an average score of 4.5 x on the Mean Opinion Score (MOS) - a standard for measuring the quality of human speech output. Some sentences like this one, "I feel my heartbeat is very fast," score higher than better-written complex sentences because they sound pronounced and evident. Some slight robotic sounds in more sophisticated or less common phrases negatively impacted the naturalness mildly. The quality of speech is close to human quality. It is able to facilitate heavy interaction with the patient and the provider, proving the ability of the system to generate clear speech output.

As shown in Table 7, MOS Evaluations for Semantic Performance of Our Model (MOS−mean opinion score) were only evaluated by a panel of 10 healthcare professionals and bilingual evaluators trained to assess clinical SoTA performance in telehealth communication. All synthesized audio samples were rated on a 5-point mean opinion score (MOS) scale from 1 = Bad, 2 = Poor, 3 = Fair, 4 = Good, and 5 = Excellent. Categories for the evaluation included the naturalness of the synthesized voice, clarity and intelligibility of the speech output, and appropriate pronunciation of medical terminologies. The averaged MOS scores are based on the subjective listener perception of these aspects. This evaluation protocol promotes consistency and validation for assessing text-to-speech quality using telemedicine in the medical field.

Table 8: Statistical comparison of STSW with SOTA models

Metric/System	Proposed System (STSW)	Shi et al. [1]	Latif et al. [3]	Kandpal et al. [5]	Ji et al. [4]	Ganesh et al. [26]
WER (Speech-to-Text)	**0.12**	0.18	0.20	0.25	0.22	0.15
BLEU (Translation)	**0.85**	-	-	0.76	0.78	-
MOS (Text-to-Speech)	**4.5**	-	-	4.2	4.3	4.1
Latency (seconds)	**2.1**	4.0	3.8	3.5	3.6	3.7
Domain Adaptability	**High**	Medium	Medium	Low	Medium	High
Multilingual Support	**Yes**	No	No	No	Yes	No
Statistical Significance (p-value)	**<0.05**	-	-	-	-	-

The statistical comparison shows that the new Speech-to-Speech Workflow (STSW) system performs better on several metrics than any state-of-the-art model. The lowest WER of 0.12 is achieved by the STSW, giving an order of improvement compared to other systems (e.g., Shi et al.). (0.18) and Latif et al. (0.20). Such performance speaks for correct speech transcription, an essential aspect of telemedicine, as correct transcription is crucial for further interpretation of patient symptoms.

Regarding translation quality, STSW has high semantic accuracy with a BLEU score of 0.85 for Hindi to English semantic meaning transmission accuracy. In contrast, comparison systems, such as those of Kandpal et al., receive lower BLEU scores as all systems are less oriented towards achieving multilingual translation capabilities. The high performance of the STSW results from domain-specific tuning and its construction related to up-to-date transformers neural translation models, which keep the model by medical nomenclature and help preserve patient-related context.

In text-to-speech synthesis, the STSW obtains the best MOS 4.5 and outperforms the compared systems. The score indicates the naturalness and comprehensibility of the generated speech, an essential factor in enabling effective communication as patients and clinicians engage with one another. Similar work Some competing systems, such as Ji et al. and Ganesh et al., garner MOS scores of 4.3 and 4.1, respectively, with even lower natural and intelligible speech output.

Another part where the STSW shines is the latency of 2.1 seconds from end to end. The benefit of low latency is that it enables real-time interactions, which is vital for telemedicine use cases. Some systems, like Shi et al., show longer latencies. The average runtime per test is 4.0 seconds for Latif et al. They are slower (3.8 seconds), making them less ideal for situations requiring fast communication. One of the reasons behind its low latency is how STSW is optimized for 5G technology, making it suitable for seamless real-time consultations.

The STSW has unique domain adaptability and multilingual characteristics compared to the RTF. It is purpose-built for medical speech and translation needs and is highly versatile for various telemedicine scenarios. Unlike systems such as Ganesh et al. and domain-specific too, but with the drawback of lacking multi-lingual support and proper end-to-end integration that comes with the STSW. Example: Kandpal et al. and Latif et al. exhibit reduced flexibility and no support for multiple languages, restricting their usability across various telehealth scenarios.

The STSW system provides a novel and holistic solution to key telemedicine issues, such as speech recognition, multilingual translation, and real-time processing. It is also superior in performance metrics for all the measured dimensions. We present a novel end-to-end speech translation system that integrates better deep learning approaches with optimized 5G technology to provide a short and robust solution to how doctors and patients can question or talk to each other in multilingual and resource-constraint settings.

Table 9: Ablation study results illustrating the impact of key components

Configuration	WER ↓	BLEU ↑	MOS ↑	Latency (s) ↓
Full STSW system	0.12	0.85	4.5	2.1
Without noise reduction	0.17	0.85	4.5	2.1
Without medical-specific fine-tuning (ASR + MT)	0.16	0.75	4.5	2.1
Without translation refinement	0.12	0.78	4.5	2.1
Without sentiment integration	0.12	0.85	4.0	2.1

Table 9 shows the ablation study results, indicating the contribution of each primary component of the proposed framework. It also shows the individual effect of selectively turning off various modules (including noise reduction, medical-specific fine-tuning, translation refinement, and sentiment integration) on the performance metrics (WER, BLEU, MOS, and latency).

Although the proposed framework exhibits a strong performance over the evaluation metrics, some limitations should be mentioned. A critical issue with this model is its sensitivity towards speech inputs with different dialects or accents, where WER increases significantly because of the limited representation of diverse dialects in training data. On the contrary, translation errors might still happen when meeting rare or region-specific medical terms not included in the training corpora, even for fine-tuning medical corpora, resulting in potential bias. Additional fine-tuning and dataset expansion may limit scalability when scaling to under-resourced languages. Such limitations signify the necessity of continual dataset diversification, including low-resource dialects and implementations of unsupervised or transfer learning methodologies for improved generalizability across multilingual, in-the-wild telehealth settings.

5 Discussion

Telemedicine has proliferated, and this development has highlighted the demand for comprehensive communication systems that allow patients and doctors to communicate seamlessly without complications. Several approaches, namely intelligent triage models, contextual chatbots, and disease-specific speech recognition platforms, have been used in telemedicine, as highlighted by existing research. Nonetheless, these best-in-class systems are still limited to independent tasks such as symptom classification, text-based interaction, or disease diagnosis. One main limitation in the literature is the absence of an integrated, multilingual, and real-time speech-to-speech communication system used in telemedicine settings.

To fill the mentioned gaps, the methodology proposed describes a new STSW that relies on deep learning. In contrast to traditional systems, the STSW combines speech recognition, translation, and text-to-speech synthesis in a single unified framework optimized for low-latency, 5G-enabled, real-time speech translation telehealth applications. NOTES Key innovations include domain optimal acceptable tuning model of speech-to-text and translation, improved text post-processing technique to handle medical terminology, and a 5G architecture that powers the entire process while keeping the latency in mind. These novelties promise accuracy, adaptability, and scalability for a wide range of telemedicine applications.

The results validate the proposed methodology, yielding better performances than state-of-the-art systems. STSW starts a new benchmark in telemedicine communication with a WER of 0.12, a BLEU score of 0.85 in translation quality, and an MOS of 4.5 in synthesized speech. Furthermore, the 2.1 seconds low latency allows for real-time interactions, addressing the significant processing times of neighboring frameworks. The STSW significantly advances the state of the art by overcoming limitations of existing systems, including limited domain adaptation, lack of multilingual support, and scalability challenges. These improvements will help to increase worldwide access to healthcare, allow for easy decentralization of telemedicine applications to countries where they are most needed, and pave the way for future telemedicine systems.

Table 8 shows that the STSW framework outperforms existing systems in key evaluation metrics. In particular, our model can reach a WER of 0.12, surpassing Shi et al. [1] (0.19) and Latif et al. [3] (0.16) due to the domain-specific fine-tuning of the Whisper ASR model on multilingual medical datasets. For translation quality, our BLEU score is 0.85, which outperforms Ji et al. [4] (0.72) without domain adaptation. The resulting text-to-speech naturalness (4.5 MOS) outperforms previous works such as Ganesh et al. [26] (MOS 4.0), thanks to our finely tuned Tacotron 2 model and optimized postprocessing. Unlike prior systems, STSW operates on 5G-enabled edge infrastructure and provides end-to-end latency of only 2.1 seconds — allowing for real-time telemedicine communication. Inspired by these earlier works, STSW directly overcomes the limitations observed in earlier works through its scalability, multilingual support, and

domain-specific fine-tuning in the medical domain. Even with these advances, there is still a need for some fine-tuning for STSW when it comes to the expansion into the under-represented languages or specialized medical subdomains. This design eliminates the high latency associated with previous systems, and due to the tasks being integrated, our framework runs all of them in parallel and has better performance, albeit at the cost of higher computational requirements and scale-up for larger tasks at inference time, a necessary trade-off for higher accuracy and further scalability.

Even though we optimized the STSW framework specifically for medicine domain speech and translation tasks in the above implementation, it can be flexibly tuned to be used in other domains with the support of fine-tuning datasets. The architecture enables recasting and training speech recognition, translations, and sentiment modules for different industries, such as legal, customer service, education, etc. At this stage, however, the system is trained predominantly based on medical contexts, and future improvements are necessary to prove device versatility in alternative settings.

We mainly reduce the latency of the overall speech translation system through 5G technology. This is done by deployment on edge servers with 5 G-enabled infrastructure, so there are no delays in data transmission between patient devices and the computation node. Moreover, due to its high bandwidth, 5G can stream and output high-quality audio inputs and outputs without degradation. However, these 5G benefits vastly enhance responsiveness, particularly critical in the case of real-time telemedicine interactions, though the system architecture itself is agnostic to the type of network used — 5G or otherwise. Section 5.1 focuses on the limitations of the study.

5.1 Limitations

Currently, the evaluation focuses on Hindi-to-English translation due to the availability of domain-specific medical datasets. However, system architecture allows for multilingual adaptability. The framework includes a modular language detection module that can recognize 10 major languages. Still, generalizing to underrepresented languages or dialects for the models only requires a few more fine-tuning datasets and model training, which is currently excluded. So, though extensible, the immediate performance of the system may be limited when applied to languages not present in the training data. Forthcoming work will focus on how to address this.

6 Conclusion and future work

This research proposes a novel speech-to-speech workflow (STSW), a deep learning-based framework tailored for multilingual telemedicine. It integrates speech recognition, machine translation, and text-to-speech synthesis, achieving strong performance across key metrics, including a Word Error Rate (WER) of 0.12, a BLEU score of 0.85, and a Mean Opinion Score (MOS) of 4.5. Compared to existing speech translation systems, STSW addresses critical limitations related to scalability, latency, and multilingual adaptability, particularly in large-scale, linguistically diverse environments. By leveraging 5G infrastructure, the framework ensures ultra-low-latency interactions. It is well-suited for time-sensitive telemedicine applications such as emergency consultations and critical care, where minimizing delays is vital. While the system remains functional on standard networks, 5G significantly optimizes real-time responsiveness. Furthermore, STSW enhances healthcare accessibility and reduces communication barriers in diverse multilingual settings. Several improvements are planned for future work. First, although the system benefits from 5G-enabled real-time performance, we recognize the necessity for offline usability in regions with limited connectivity. We will develop optimized, lightweight, on-device versions of the speech recognition, translation, and TTS modules suitable for low-power environments to address this. Second, future iterations will incorporate additional fine-tuning datasets for underrepresented languages and regional dialects to expand language inclusivity. Additionally, we plan to conduct clinical validation studies involving healthcare professionals and patients to assess real-world usability and clinical effectiveness.

References

[1] Jinming Shi, Ming Ye, Haotian Chen, Yaoen Lu, Zhongke Tan, Zhaohan Fan, and Jie Zhao. (2023). Enhancing efficiency and capacity of telehealth services with intelligent triage: a bidirectional LSTM neural network model employing character embedding. Springer. 23(269), pp.1-10. https://doi.org/10.1186/s12911-023-02367-1

[2] Denise D. Pay a Jennifer L. Frehn, Lorena Garcia, Aaron A. Tierney, and Hector P. Rodriguez. (2022). Telemedicine implementation and use in community health centers during COVID-19: Clinic personnel and patient perspectives. Elsevier. 2, pp.1-9. https://doi.org/10.1016/j.ssmqr.2022.100054

[3] Latif, Siddique; Qadir, Junaid; Qayyum, Adnan; Usama, Muhammad and Younis, Shahzad (2020). Speech Technology for Healthcare: Opportunities, Challenges, and State of the Art. IEEE Reviews in Biomedical Engineering, 14 pp.1-15. http://doi:10.1109/RBME.2020.3006860

[4] Xinyu Ji; Ellen Chow; Kenzy Abdelhamid; Darya Naumova; Kedar K.V. Mate; Amy Bergeron and Bertrand Lebouché; (2021). Utility of mobile technology in medical interpretation: A literature review of current practices. Patient Education and Counseling, 104(9), pp. 2137-2145. http://doi:10.1016/j.pec.2021.02.019

[5] Kandpal, Prathamesh; Jasnani, Kapil; Raut, Ritesh; Bhorge, Siddharth (2020). Contextual Chatbot for Healthcare Purposes (using Deep Learning). IEEE, pp.625–634. http://doi:10.1109/WorldS450073.2020.9210351

[6] Albahri, A.S.; Alwan, Jwan K.; Taha, Zahraa K.; Ismail, Sura F.; Hamid, Rula A.; Zaidan, A.A.; Albahri, O.S.; Zaidan, B.B.; Alamoodi, A.H. and Alsalem, M.A. (2021). IoT-based telemedicine for disease prevention and health promotion: State-of-the-Art. Journal of Network and Computer Applications, 173, pp.1-59. http://doi:10.1016/j.jnca.2020.102873

[7] Olivia Li, Ji-Peng; Liu, Hanruo; Ting, Darren S.J.; Jeon, Sohee; Chan, R.V. Paul; Kim, Judy E.; Sim, Dawn A.; Thomas, Peter B.M.; Lin, Haotian; Chen, Youxin; Sakomoto, Taiji; Loewenstein, Anat; Lam, Dennis S.C.; Pasquale, Louis R.; Wong, Tien Y.; Lam, Linda A. and Ting, Daniel S.W. (2020). Digital technology, tele-medicine and artificial intelligence in ophthalmology: A global perspective. Progress in Retinal and Eye Research, 82 pp.1-102. http://doi:10.1016/j.preteyeres.2020.100900

[8] Yuezhou Zhang, Amos A. Folarin, Judith Dineley, Pauline Conde, Valeria de Angel, Shaoxiong Sun, Yatharth Ranjan, Zulqarnain Rashid, Callum Stewart, Petroula Laiou, Heet Sankesara, Linglong Qian, Faith Matcham, Katie White, Carolin Oetzmann, Femke Lamers, Sara Siddi, Sara Simblett, Bjorn ¨ W. Schuller, Srinivasan Vairavan, Til Wykes, Josep Maria Haro, Brenda W.J.H. Penninx, Vaibhav A. Narayan, Matthew Hotopf, Richard J.B. Dobson, Nicholas Cummins and RADAR-CNS consortium. (2024). Identifying depression-related topics in smartphone-collected free-response speech recordings using an automatic speech. Elsevier. 355, pp.40-49. https://doi.org/10.1016/j.jad.2024.03.106

[9] Veena Calambur, Dong Whan Jun, Melody Schiaffino, Zhan Zhang and Jina Huh-Yoo. (2024). A case for" little English" in Nurse Notes from the Telehealth Intervention Program for Seniors: Implications for Future. ACM. .(238), pp.1-16. https://doi.org/10.1145/3613904.3641961

[10] Harshvadan Talpada, Malka N. Halgamuge and Nguyen Tran Quoc Vinh. (2019). An analysis on use of deep learning and lexical-semantic based sentiment analysis method on twitter data to understand the Demographic Trend of Telemedicine. IEEE, pp.1-9. http://DOI:10.1109/KSE.2019.8919363

[11] Heng Yu and Zhiqing Zhou; (2021). Optimization of IoT-Based Artificial Intelligence Assisted Telemedicine Health Analysis System. IEEE Access, 9, pp. 85034 - 85048. http://doi:10.1109/ACCESS.2021.3088262

[12] Ozan Ozyegen, Devika Kabe and Mucahit Cevik. (2022). Word-level text highlighting of medical texts for telehealth services. Elsevier. 127, pp.1-33. https://doi.org/10.1016/j.artmed.2022.102284

[13] Chung, Sheng-Luen; Chen, Yi-Shum; Su, Shun-Feng and Ting, Hsien-Wei. (2019). Preliminary Study of Deep Learning based Speech Recognition Technique for Nursing Shift Handover Context. pp.528–533. http://doi:10.1109/SMC.2019.8913954

[14] P. Deepa and Rashmita Khilar. (2022). Speech technology in healthcare. Elsevier. 24, pp.1-11. https://doi.org/10.1016/j.measen.2022.100565

[15] Ayush Tripathi; Swapnil Bhosale and Sunil Kumar Kopparapu; (2021). Automatic speaker independent dysarthric speech intelligibility assessment system. Computer Speech & Language, 69, pp.1-17. http://doi:10.1016/j.csl.2021.101213

[16] Jun Zhang, Jingyue Wu, Yiyi Qiu, Aiguo Song, Weifeng Li, Xin Li and Yecheng Liu. (2023). Intelligent speech technologies for transcription, disease diagnosis, and medical equipment interactive control in smart. Elsevier. 153, pp.1-29. https://doi.org/10.1016/j.compbiomed.2022.106517

[17] Manoj Kaushik; Neeraj Baghel; Radim Burget; Carlos M. Travieso and Malay Kishore Dutta; (2021). SLINet: Dysphasia detection in children using deep neural network. Biomedical Signal Processing and Control. 68, pp.1-13. http://doi:10.1016/j.bspc.2021.102798

[18] Syu-Siang Wang, Chi-Te Wang, Chih-Chung Lai, Yu Tsao, and Shih-Hau Fang. (2023). Continuous Speech for Improved Learning Pathological Voice Disorders. IEEE. 3, pp.25 - 33. http://DOI:10.1109/OJEMB.2022.3151233

[19] IRUM SINDHU AND MOHD SHAMRIE SAININ. (2024). Automatic Speech and Voice Disorder Detection using Deep Learning-A Systematic Literature Review. IEEE. 12, pp.49667 - 49681. http://DOI:10.1109/ACCESS.2024.3371713

[20] Wen-Chin Huang, Bence Mark Halpern, Lester Phillip Violeta, Odette Scharenborg, Tomoki Toda. (2021). Towards Identity Preserving Normal to Dysarthric Voice Conversion. IEEE, pp.1-5. http://DOI:10.1109/ICASSP43922.2022.9747550

[21] Alam, M.; Samad, M.D.; Vidyaratne, L.; Glandon, A. and Iftekharuddin, K.M. (2020). Survey on Deep Neural Networks in Speech and Vision Systems. Neurocomputing, 417, pp. 302-321. http://doi:10.1016/j.neucom.2020.07.053

[22] M. Tanveer, Aryan Rastogi, Vardhan Paliwal, M.A. Ganaie, A.K. Malik, Javier Del Ser and Chin-Teng Lin. (2023). Ensemble deep learning in speech signal tasks: A review. Elsevier. 550, pp.1-26. https://doi.org/10.1016/j.neucom.2023.126436

[23] K. Aditya Shastry and Aravind Shastry. (2023). An integrated deep learning and natural language processing approach for continuous remote monitoring in digital health. Elsevier. 8, pp.1-14. https://doi.org/10.1016/j.dajour.2023.100301

[24] ÜLGEN SONMEZ ¨ and Asaf VAROL. (2024). In-depth investigation of speech emotion recognition studies from past to present –The importance of emotion recognition from speech signal for AI–. Elsevier. 22, pp.1-12. https://doi.org/10.1016/j.iswa.2024.200351

[25] Mohamed Talaat, Kian Barari, Xiuhua April Si and Jinxiang Xi. (2024). Schlieren imaging and video classification of alphabet pronunciations: exploiting phonetic flows for speech recognition. Springer. 7(12), pp.1-14. https://doi.org/10.1186/s42492-024-00163-w

[26] Devalla Bhaskar Ganesh, Yellamma Pachipala, Syed Sania Rizvi, Teena Chowdary Manne, Himavanth Swamy Atchi, andV V R Maheswar. (2024). Flask-based ASR for Automated Disorder Speech Recognition. Elsevier. 233, pp.623-637. https://doi.org/10.1016/j.procs.2024.03.252

[27] M. Musalia, S. Laha, J. Cazalilla Chica, J. Allan, L. Roach, J. Twamley, S. Nanda, M. Verlander, A. Williams, I. Kempe, I. I. Patel, F. Campbell West, B. Blackwood and D. F. McAuley. (2023). A user evaluation of speech/phrase recognition software in critically ill patients: a DECIDE-AI feasibility studyM. Musa. Springer. 27(277), pp.1-6. https://doi.org/10.1186/s13054-023-04420-x

[28] Hamza Kheddar, Yassine Himeur, Somaya Al-Maadeed, Abbes Amira and Faycal Bensaali. (2023). Deep transfer learning for automatic speech recognition: Towards better generalization. Elsevier. 277, pp.1-34. https://doi.org/10.1016/j.knosys.2023.110851

[29] Jorge Arenas Gaitan´ and Patricio E. Ramírez-Correa. (2023). COVID-19 and telemedicine: A netnography approach. Elsevier. 190, pp.1-19. https://doi.org/10.1016/j.techfore.2023.122420

[30] Dwaipayan Bandopadhyay, Rajdeep Ghosh, Rajdeep Chatterjee, Nabanita Das and Bikash Sadhukhan. (2023). Speech Recognition and Neural Networks based Talking Health Care Bot (THCB): Medibot. IEEE., pp.1-6. http://DOI:10.1109/ICCMC56507.2023.10084191

[31] M. Tanveer, Aryan Rastogi, Vardhan Paliwal, M.A. Ganaie, A.K. Malik, Javier Del Ser and Chin-Teng Lin (2023). Ensemble deep learning in speech signal tasks: A review. Elsevier. 550, pp.1-26. https://doi.org/10.1016/j.neucom.2023.126436

[32] Seyed Reza Shahamiri, Vanshika Lal, and Dhvani Shah. (2023). Dysarthric Speech Transformer: A Sequence-to-Sequence Dysarthric Speech Recognition System. IEEE. 31, pp.3407 - 3416. http://DOI:10.1109/TNSRE.2023.3307020

[33] CHIA-TUNG WU, SSU-MING WANG, YI-EN SU, TSUNG-TING HSIEH, PEI-CHEN CHEN, YU-CHIEH CHENG, TZU-WEI TSENG, WEI-SHENG CHANG, CHANG-SHINN SU, LU-CHENG KUO, JUNG-YIEN CHIEN, AND FEIPEI LAI. (2022). A Precision Health Service for Chronic Diseases: Development and Cohort Study Using Wearable Device, Machine Learning, a. IEEE. 10, pp.1-14. http://DOI:10.1109/JTEHM.2022.3207825

[34] P. Deepa and Rashmita Khilar. (2022). Speech technology in healthcare. Elsevier. 24, pp.1-11. https://doi.org/10.1016/j.measen.2022.100565

[35] Amlu Anna Joshy and Rajeev Rajan. (2022). Automated Dysarthria Severity Classification: A Study on Acoustic Features and Deep Learning Techniques. IEEE. 30, pp.1147 - 1157. http://DOI:10.1109/TNSRE.2022.3169814

[36] Jaromir Przybyło. (2022). A deep learning approach for remote heart rate estimation. Elsevier. 74, pp.1-10. https://doi.org/10.1016/j.bspc.2021.103457

[37] Ashwin Kamble, Pradnya H. Ghare, and and Vinay Kumar. (2023). Deep-Learning-Based BCI for Automatic Imagined Speech Recognition Using SPWVD. IEEE. 72, pp.1-10. http://DOI:10.1109/TIM.2022.3216673

[38] Suman Deb, Pankaj Warule, Amrita Nair, Haider Sultan, Rahul Dash and Jarek Krajewski. (2022). Detection of common cold from speech signals using deep neural network. Springer. 42, p.1707–1722. https://doi.org/10.1007/s00034-022-02189-y

[39] Jefferson Gomes Fernandes. (2022). Artificial intelligence in telemedicine. Spriner, pp.1-10. https://doi.org/10.1007/978-3-030-58080-3_93-1

[40] Mohammed Abdelhay, Ammar Mohammed and Hesham A. Hefny. (2023). Deep learning for Arabic healthcare: MedicalBot. Springer. 13(71), pp.1-17. https://doi.org/10.1007/s13278-023-01077-w

[41] Mozilla. (2024). Common Voice Dataset. [Online]. Available at: https://commonvoice.mozilla.org.

[42] Shajulin Benedict, Rubiya Subair. (2025). Deep Learning-Driven Edge-Enabled Serverless Architectures for Animal Emotion Detection. Informatica. 49, p.33–48. https://doi.org/10.31449/inf.v49i7.6615

[43] Chunyan Han, Ling Lin. (2024). Detecting and Tracking Rumours in Social Media Based on Deep Learning Algorithm. Informatica. 48, p.83–96. https://doi.org/10.31449/inf.v48i14.5998

[44] Desheng Chen, Sifang Zhang. (2025). Deep Learning-Based Involution Feature Extraction for Human Posture Recognition in Martial Arts. Informatica. 49, p.77–90. https://doi.org/10.31449/inf.v49i12.7041

T-Extractor: A Hybrid Unsupervised Approach for Term and Named Entity Extraction Using Rules, Statistical, and Semantic Methods

Aliya Kalykulova*, Aliya Nugumanova
"Big Data and Blockchain Technologies" Science and Innovation Center, Astana IT University,
Astana, Kazakhstan
E-mail: aliyakalykulova01@gmail.com, a.nugumanova@astanait.edu.kz
* Corresponding author

Keywords: automatic term extraction, unsupervised annotator, T-Extractor, phrase extraction, semantic analysis

Received: January 27, 2025

Automatic term extraction is a key technology for optimizing natural language processing tasks such as machine translation, sentiment analysis, knowledge graph construction, and/or ontology population. This study presents the T-Extractor approach for unsupervised term extraction. The research goal is to develop an efficient method that does not require labeled data, and to analyze its applicability on scientific texts. T-Extractor combines rule-based, statistical, and semantic analysis, treating unigram and phrase extraction as two subtasks. Part-of-speech templates are used in the candidate selection phase, while a filter based on raw and rectified frequencies refines phrase boundaries. TopicScore is applied for final term filtering, improving extraction precision. Additionally, simple rules help identify abbreviations and named entities, improving recall. T-Extractor was tested on the ACTER (three languages, four domains) and ACL RD-TEC 2.0 datasets. In English, the best result was achieved in the equi domain, with an F1-measure of 48.5%, precision of 41.6%, and recall of 58.2%. On the ACTER dataset, the approach outperformed existing unsupervised methods and performed better than the supervised GPT-3.5-Turbo and BERT models in the corp and wind domains. Specifically, in the corp domain, T-Extractor's F1-measure approached that of the HAMLET model, lagging by 3.7%. In addition, the method showed results comparable to those of promtATE and TALN-LS2N.

Povzetek: T-Extractor je hibridna, nenadzorovana metoda za izvleček izrazov in imen, ki z združitvijo pravil, statistik in semantike presega tudi nadzorovane modele na določenih domenah.

1 Introduction

Knowledge-driven digital products are the cornerstone of Industry 4.0, and the most popular format for representing knowledge is knowledge graphs. Developing large knowledge graphs manually is an expensive process, for which natural language processing techniques, including automatic terminology extraction methods, have proven useful in speeding up and scaling. In this paper, we focus on applying an unsupervised approach for automatic term extraction, and we intend to show that with the proper strategy, they can be competitive even when training data is scarce. In 2018, Gartner included knowledge graphs in its famous "Hype Cycle for Emerging Technologies" and 4 years later, industry leaders such as Siemens [1], Bosch [2], and Mitsubishi Electric [3] have proven first-hand that knowledge graphs have successfully moved from the realm of hype to the real economy and even reached a productivity plateau in some cases.

Today, there is a renewed interest in knowledge graphs among researchers due to the human-centered challenges of Industry 5.0 and the growing realization that artificial intelligence alone, without human input, cannot be the basis for building robust systems [4-6]. Expectations related to knowledge graphs center mainly around the idea of combining them with machine/deep learning models to produce better and more explainable cognitive solutions and serve as groundwork for the next generation of systems based on human-machine synergy [6].

These expectations, in turn, stimulate research in the field of computational terminology, a science at the intersection of knowledge engineering and natural language processing dealing with automatic term extraction. After all, it is terms, as expressors of domain concepts, that are the basic building blocks for knowledge graphs. However, even without additional motivation, computational terminology is currently at an important transition stage due to the appearance of transformers, and this stage is no less significant in terms of expected results than the one caused by the arrival of statistical methods in the industry in the 1990s [7].

According to [8], the growth rate of new published articles continues to increase. This makes manual text processing almost impossible, which necessitates a transition to digital data processing and knowledge graph construction for more efficient information analysis. The construction of knowledge graphs requires the selection of informative text units and the establishment of links between them [9]. Terms, named entities, and other text elements are considered as such units. In addition, term extraction plays a key role in machine translation tasks,

search engines, text abstracting and other applications [10, 11]. Some studies also suggest using key phrases to improve information retrieval [12] or text classification [13], illustrating the broader significance of phrase extraction across various domains. Since terminology often includes multi-word expressions, phrase extraction is closely related to term extraction, as both aim to identify meaningful linguistic units that structure domain knowledge. This study focuses on the extraction of terms and named entities, including multi-word terms, which naturally intersects with phrase extraction methods.

Existing approaches for automatic term extraction can be roughly categorized into supervised and unsupervised approaches [10]. Supervised methods provide high accuracy but require significant amounts of labeled data for training, which makes them difficult to adapt to new domains and reduces their effectiveness when processing texts from different subject areas. Unsupervised methods, on the other hand, have greater versatility but show less accuracy when recognizing terms and named entities.

Term extraction faces several challenges, including the presence of noise in the data, difficulty in identifying the boundaries of multi-word expressions, and polysemy. The accuracy of extraction also depends on the quality of the models used, such as embedding generation models and algorithms for part-of-speech detection. Therefore, the development of methods capable of solving textual data analysis problems while considering the existing limitations of tools and resources is still ongoing.

The research aims to develop a more efficient unsupervised approach for extracting terms and named entities. This paper presents an unsupervised annotator, T-Extractor, which extracts terms and named entities using rules, statistical and semantic analysis. The proposed annotator demonstrated an average F1-measure of 40% on the ACTER and ACL RD-TEC 2.0 datasets.

The following research questions were posed:

1. What is the impact of combining statistical, rule-based, and semantic techniques in term extraction?

2. How does the proposed T-Extractor compare to existing unsupervised and supervised methods in extracting terms across multiple domains and languages?

The paper is structured as follows: first, an overview of existing methods and approaches is presented, which allows us to determine the current state of research in this area. Then, the datasets used for annotator quality assessment are presented.

Next, the principles and algorithms of the proposed approach are detailed, including key aspects of its implementation. Then, the developed methodology for quality assessment of the proposed approach is described, which ensures the objectivity and reproducibility of the obtained results.

We focus on analyzing the results, discussing them, and identifying the strengths and weaknesses of the method. The paper is concluded with conclusions and recommendations aimed at practical application of the developed annotator in various text processing tasks.

2 Related work

A term, according to a common definition, is a word or phrase used to refer to a specific concept, subject or phenomenon in a particular field of knowledge. As [14] points out, terms are a valuable linguistic resource that contributes to linguistic coherence. With the development of digital resources and natural language processing (NLP) tools, terms have gained a key role in knowledge graph creation, electronic document management, and data analysis. This has greatly expanded their use beyond traditional tasks such as translation [15]. Terms play a crucial role in structuring and categorizing information, becoming the basis for organizing data in various information systems.

According to common standards, terms are classified into simple (single word) and complex (multi-word) terms [16, 17]. Studies by [17] show that 99% of technical multi-word terms take the form of noun phrases (NPs), whose key element is a noun or its grammatical equivalent.

Named Entities (NEs) are real-world entities that can be identified, such as people, places, organizations, dates, or products. According to the MUC-7 classification, Named Entities are categorized as follows: persons, organizations, locations, dates, times, monetary amounts, and percentages.

Thus, terms and named entities are key components in representing and analyzing information, making them an important research subject in natural language processing and data management. Table 1 presents a comparative analysis of existing methods. Next, an overview of unsupervised and supervised approaches to term extraction is given.

2.1 Unsupervised approaches

Unsupervised information extraction methods are based on rules, frequency features, semantics or their hybrid combinations [18, 19]. The standard unsupervised

Table 1: Summary of related work

Method name	**Data sets**	**F1 score, %**	**Limitations**
Unsupervised			
NMF [20]	ACTER	corp_en: 25,7 equi_en: 33,3 wind_en: 26,1 htfl_en: 33,7	Does not consider word semantics. The method relies on statistical metrics based on word frequency, making it vulnerable to noise in the data. It is also sensitive to parameter settings, including the number of topics, the number of terms to extract, and term length.
UA [21]	ACL-RD TEC 2.0., GENIA, ScienceIE	ACL: 49,95 GENIA: 45,65 ScienceIE: 39,7	Extracts only noun phrases and does not always correctly determine term boundaries, leading to term splitting or merging with irrelevant parts. No semantic filtering for unigrams.
UA1 [22]	ACTER, ACL-RD TEC 2.0.	corp_en: 24,3 equi_en: 28,9 wind_en: 29,5 htfl_en: 32,7 ACL: 44,8	A reimplementation of the UA approach, achieving an F1 score 5.15% lower than the original UA method on the ACL dataset.
Supervised			
HAMLET [27]	ACTER	corp_en: 43,8 equi_en: 60,1 wind_en: 50,1 htfl_en: 55,4	Requires a large number of features. The method computes 152 features per candidate, making training more complex and computationally expensive. It also depends on the quality of training data and has limited adaptability to new domains.
TALN-LS2N [28]	ACTER	htfl_en: 46,66 htfl_fr: 48,15	Sensitive to the n-gram parameter, limiting term length to 4-grams for English and 5-grams for French, which makes extracting longer terms difficult. Also, it requires a large amount of labeled data for training.
GPT-3.5-Turbo [29]	ACTER	corp_en: 31,4 equi_en: 49,7 wind_en: 32,5 htfl_en: 55,6	Token limit constraints reduce the amount of text available for analysis. In broad subject areas (e.g., renewable energy), the method may include irrelevant terms.
promptATE [30]	ACTER	htfl_en: 51,4 htfl_fr: 47,8 htfl_nl: 55,4	Trained on general data and does not account for domain specificity, leading to over-extraction of terms (high recall but low precision).

approach algorithm includes the following steps: 1) candidate extraction, 2) ranking based on certain features, and 3) selection of top candidates using a threshold [20].

Article [20] proposed an unsupervised annotator based on matrix decomposition using the Non-Negative Matrix Factorization (NMF) method. This approach was tested in the TermEval 2020 competition, where it achieved an average F1 score of 27.2% on the ACTER dataset. The NMF method demonstrates its versatility as it can be adapted to handle different languages and subject areas.

Another term extraction approach, called Unsupervised Annotator (UA), is proposed by [21]. It is based on the use of part-of-speech rules, morphological analysis, and two metrics, Topic Score and Specific Score, computed based on cosine similarity of contextual embeddings. The model was tested on the ACL, GENIA and ScienceIE datasets, achieving an average performance of 45.11% on the F1 metric.

The Unsupervised Annotator (UA) approach from [21] was reimplemented by another research group and tested on the ACTER dataset, as the original model code was not publicly available [22]. The implemented UA1 annotator achieved F1=44.8% on the ACL dataset, which is 5.15% lower than the result of the original model (F1=50%). UA1 achieved an average F1=28.9% on the ACTER (English) dataset. The lower performance is attributed to the high variability of part-of-speech combinations in terms of the ACTER dataset. Additionally, multi-word expressions could overlap with

true terms, indicating an incorrect definition of phrase boundaries. Despite this, overall, the UA1 annotator shows high performance when applied to the ACL RD-TEC 2.0 corpus.

Term and keyword extraction share similar objectives, as a term can function as a keyword. The main task of terminology is to generate conceptual descriptions, whereas keywords are intended to reflect the content of the text [23]. Nevertheless, similar techniques can be used to extract informative words.

An example of an unsupervised approach to keyword extraction is the YAKE! model [24]. This method uses different frequency metrics, considers the position of the word in the text as well as its case. The authors note that relevant words are more likely to occur at the beginning of the text or headings, and capitalized words can be significant.

A review of keyword extraction methods presented in [23] emphasizes that most key phrases are noun groups (noun phrases), making their extraction an important step to improve accuracy.

In [13], an unsupervised Subword-Phrase extraction method based on frequency analysis is proposed to improve text classification. The approach demonstrated that a supervised classification model achieves better results when using phrases as one of its features. This confirms the importance of leveraging lexical units that convey the main topic of the text. Since terms are more specific, their application may be even more effective for classification.

Most unsupervised approaches rely on frequency-based features. According to [25], the distributions of noise and quality phrases have similar patterns, making it difficult to extract relevant terms. One of the key challenges is the correct definition of phrase boundaries. Additionally, extracting rare terms is challenging due to their low frequency, which makes it difficult to establish accurate boundaries.

2.2 Supervised approaches

Supervised approaches to term extraction significantly outperform unsupervised approaches [26]. This is due to the complexity of textual data processing, where factors such as case, context, parts of speech, and special punctuation marks influence term extraction. Accounting for all these aspects simultaneously is challenging, as exceptions may exist for each factor. However, machine learning can address this problem comprehensively, making it an effective tool in this field.

One such approach, HAMLET, utilizes over 160 features, including statistical, variational, linguistic, and contextual features, for model training [27]. A random forest-based algorithm demonstrated the best performance in this approach.

The TALN-LS2N approach, presented in [28], was trained on both true terms and false examples. After training, additional filtering is applied: candidate terms starting with conjunctions and pronouns are excluded, and duplicate or common words are removed.

[29] presents an approach using the GPT-3.5-Turbo model with few-shot scripts. To extract terms, a prompt is generated that contains instructions (e.g., “find a term”), a sentence to analyze, and an example of terms. This method stands out for its simple implementation and minimal reliance on labeled data.

Another state-of-the-art approach, promptATE [30], is also based on the use of prompts for term extraction. This approach uses two models, ChatGPT (gpt-3.5-turbo) and Llama 2-Chat, and implements three result output formats. The sequence-labeling approach achieves high precision but low recall. The text-extractive response format uses partial markup with skips, which allows more terms to be extracted but reduces precision. In turn, text generative response, which uses labeled cues, provides an optimal balance between precision and recall.

Traditionally, term extraction was treated as a binary classification task (term/non-term), requiring a large amount of labeled data. However, a new approach has emerged, leveraging prompt-based methods for term generation. This approach requires significantly less labeled data, making it a promising direction for further research.

3 Dataset

The effectiveness of the proposed approach was evaluated using the ACTER and ACL RD-TEC 2.0 corpora, both of which contain texts with labeled terms.

The ACTER (Annotated Corpora for Term Extraction Research) corpus consists of manually annotated texts spanning four topic areas: corruption (corp), training (equi), heart failure (htfl), and wind energy (wind). This corpus contains texts in three languages: English (en), French (fr), and Dutch (nl) [19]. In this study, extracted term candidates were compared against a reference list of true terms, which includes named entities.

The ACL Reference Dataset for Terminology Extraction and Classification (ACL RD-TEC 2.0) corpus, released in 2016, contains annotated abstracts of scientific articles in computational linguistics. A distinctive feature of this corpus is its double annotation by two independent annotators [31], which reduces potential bias and enhances the accuracy of method evaluation.

4 Approach description

T-Extractor is an unsupervised tool for extracting terms and named entities through rule-based, frequency-based, and semantic analysis. At the initial stage, rule-based and frequency analysis help identify potential candidates, while semantic analysis further filters them, selecting the most relevant and domain-specific units.

Extracting multi-word expressions is more challenging than extracting single-word terms. This is due to several challenges, including term boundary definition, nested terms, and syntactic variations across languages. Therefore, term extraction is performed in two stages: unigram extraction and phrase extraction. Abbreviations and named entities exhibit distinct features that facilitate their identification in text (e.g., capitalization patterns).

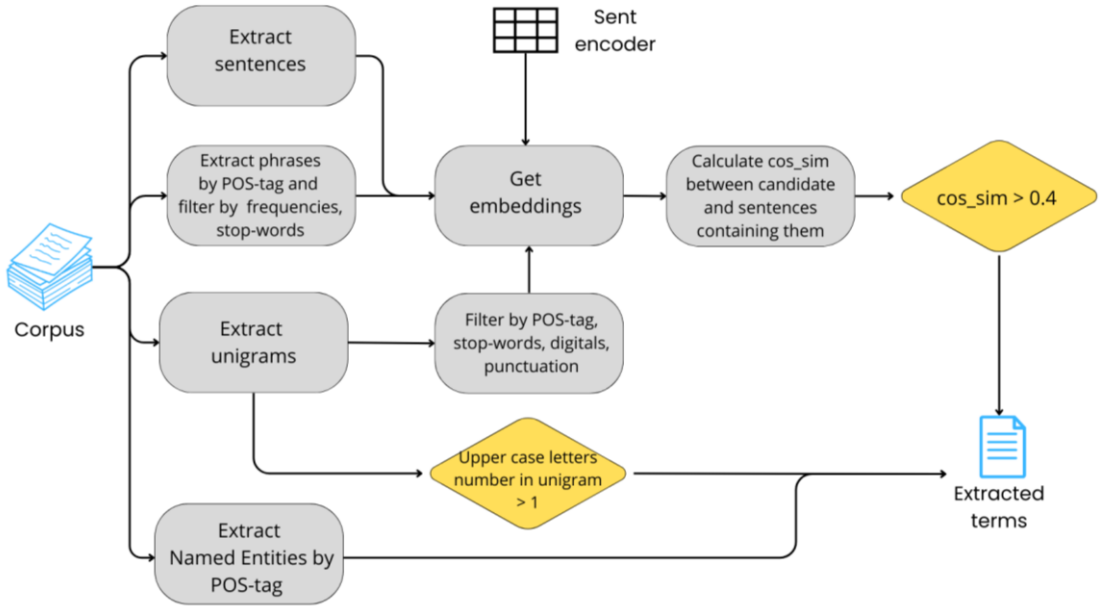


Figure 1: Generalized algorithm for extracting terms, named entities and abbreviations. The input of the algorithm is a text or a corpus of texts, and the output is a final list of extracted information units.

Simple rules were applied for their extraction. While extracting unigrams and phrases, abbreviations and named entities can also be extracted. The tool does not classify extracted information but instead consolidates key units, such as terms, named entities, and abbreviations, into a single list.

Unigram extraction involves selecting candidates based on part-of-speech tagging and pre-filtering them. Multi-word expression extraction relies on part-of-speech patterns combined with filtering based on two frequency metrics. Low-frequency candidates are filtered using a phrase-grouping approach based on word position matching, followed by selecting the most frequent candidate. This approach mitigates ambiguity in term boundaries by selecting the most likely candidates.

The extracted unigrams and multi-word expressions are filtered using the Topic Score metric proposed in [21]. This metric allows selecting candidates that are most relevant to the thematic area of the analyzed text.

Figures 1 and 2 illustrate the term extraction process. A detailed algorithm description is also provided in these figures. A detailed description of the algorithm's key steps is provided in the following section. The first section covers data preparation and model setup for term extraction. The second section describes the unigram extraction algorithm. The third section explains the phrase extraction technique. The fourth section details semantic filtering of extracted candidates using Topic Score. The fifth section introduces an abbreviation extraction approach. The sixth section describes named entity extraction techniques. Finally, the seventh section discusses the models used and threshold tuning parameters.

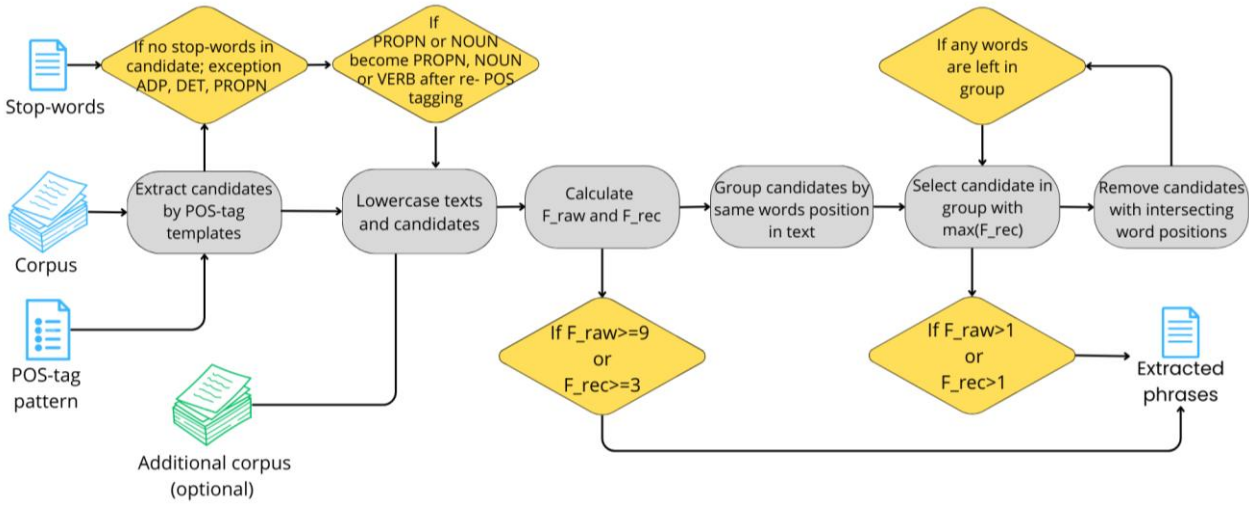


Figure 2: Phrase term extraction algorithm.

Table 2: Patterns of part-of-speech combinations for extracting multi-word expressions.

Templates for English		
1) N*	5) N*, ADJ*, N*	9) N, ADP, N
2) ADJ*, N*	6) ADJ, VERB, N*	10) K*, ADP, N*
3) ADJ*	7) VERB*, N*	11) M (if there is a hyphen)
4) VERB, ADJ, N*	8) ADV*, ADJ*	
N - [PROPN, NOUN]	K - [ADJ, PROPN, NOUN]	M - [VERB, ADV, X]
Templates for French		
1) K*	4) NOUN, VERB	7) N*, ADP, N*, ADP, N*
2) N*, ADP, N*	5) VERB, ADJ	8) N*, ADP, N*, ADP, N*, ADP, N*
3) N*, ADP, DET, N*	6) ADJ, VERB	
N - [NOUN, ADJ]	K - [NOUN, ADJ, PROPN]	
Templates for Dutch		
1) K*	3) NOUN, ADP, NOUN, ADP, NOUN	5) VERB, ADP, DET, NOUN
2) N*, ADP, M*	4) VERB, ADP, NOUN	6) VERB,NOUN
N - [NOUN, ADJ]	K - [NOUN, ADJ, PROPN, SYM]	M - [NOUN, ADJ, CCONJ]

* - asterisk marks those part-of-speech that can be one or more in a sequence [25].

N, K, M - means that their position can be any part of speech from the specified list

If *N**, *K** or *M**, then there can be one or more of any part-of-speech from the list.
Example N* (for English):

- NOUN, NOUN, NOUN
- NOUN, PROPN,
- PROPN,PROPN, NOUN

4.1 Data and model preparation

For text tokenization and part-of-speech partitioning, the SpaCy model was used, which was preconfigured so that hyphenated words were not separated into separate tokens. This setting avoided additional complexity in developing part-of-speech patterns for phrase extraction and prevented the extraction of incomplete words. For example, the word *"anti-corruption"* in this case should be treated as a whole, since the component *"anti"* clearly refers to *"corruption"* and cannot be treated as an independent term. This processing minimizes noise in the data by ensuring that hyphenated words are extracted in their integral form.

In the unigram extraction stage, words with a hyphen are extracted as single tokens. However, in the performance evaluation presented in Table 8, this category is classified as multi-word expressions because it consists of two tokens joined by a hyphen.

Before part-of-speech tagging, the text is not converted to lowercase, which allows the SpaCy model to identify proper names (*PROPN*) more accurately. After extracting candidates based on part-of-speech tagging, the extracted units are converted to lower case. This step is necessary to perform subsequent tasks correctly and to obtain a more accurate evaluation of the extracted terms.

4.2 Unigram extraction

Extraction of candidate unigrams is performed based on part-of-speech partitioning. Nouns (*NOUN*), proper names (*PROPN*) and adjectives (*ADJ*) are considered as candidates. To reduce noise, prefiltering removes stop words and unigrams consisting only of digits and/or punctuation marks. Additionally, words containing punctuation marks, except hyphen ("-") and apostrophe (" ' "), as long as they appear within a word rather than at its boundary, are filtered out.

4.3 Extracting multi-word expressions

The phrase extraction algorithm presented in Figure 2 includes five main steps: (1) candidate extraction using part-of-speech templates, (2) pre-filtering of candidates, (3) raw and rectified frequency calculation, (4) phrase grouping by position followed by extraction of the most frequent phrases, and (5) phrase extraction based on thresholds.

4.3.1 Candidate extraction using part-of-speech templates

The templates used in this approach are given in Table 2. Phrases are selected as candidates if they are longer than one word or if they contain a hyphen. All phrases containing punctuation marks except for the hyphen and the apostrophe are excluded.

Special symbols were used to optimize the selection of part-of-speech patterns and avoid unnecessary combination of variants. The asterisk (*) indicates that a given Part-of-Speech can be followed by the same Part-of-Speech. This approach was adapted from [25] to create flexible templates.

Table 3: Comparative analysis of multi-word term and NE candidate extraction using different sets of part-of-speech patterns in the corp(en) domain from ACTER.

POS-tag patterns for phrases	N	P	R	F1
SpaCy Noun chunks	5950	12,5	48,88	19,91
Ngrams (from 2 to 5 tokens), with pre-filtering from punctuation and digits	92817	1,2	73,39	2,37
[25]	7354	14,65	70,76	24,27
[32]	3398	14,18	69,25	23,55
[33]	2722	14,36	56,18	22,88
Proposed templates	13404	9,94	87,52	17,85

Letters *N, K, M* represent sets of parts of speech. In a template, any part of speech from the corresponding list can be specified at their position. For example, in the *ADJ+NOUN* and *ADJ+PROPN* templates, the position of *NOUN* can be *PROPN*. Thus, the *ADJ+N* template allows extracting both *ADJ+PROPN* and *ADJ+NOUN*.

If the characters *N, K, M* are followed by an asterisk sign, the part-of-speech sequence may contain any part-of-speech from the specified list. For example, the formula *N** can result in combinations such as *PROPN+PROPN*, *NOUN+NOUN+NOUN*, *NOUN+PROPN+PROPN*, etc.

The part-of-speech templates were computed and selected based on the analysis of terms from the ACTER dataset. This template format achieves an average recall of approximately 88% for English, 75% for French, and 75% for Dutch when extracting phrase terms from the ACTER dataset. This approach provides flexibility to create different template variations, making it significantly easier to capture all possible term combinations.

Table 3 presents the results of multi-word term and named entity extraction using part-of-speech patterns in the corp(en) domain of the ACTER dataset. The proposed approach is compared with Noun Chunks (a SpaCy method designed to extract noun phrases), n-gram extraction (ranging in size from 2 to 5 tokens with pre-filtering) and three sets of part-of-speech templates described in [25, 32, 33].

The Noun Chunks method is limited to the extraction of noun phrases, resulting in a low recall. The extraction of n-grams exhibits high recall, but the precision remains extremely low. Moreover, increasing the length of n-grams results in even lower precision since longer terms are less frequent.

The parts-of-speech templates from [25, 32, 33] provide higher precision and F1-measures compared to the proposed templates. However, the templates proposed in this paper show the highest recall value in extracting phrase terms. The main goal at this stage is to maximize the coverage of potential candidates, since subsequent filtering may lead to the removal of terms themselves, negatively impacting recall.

Table 4 presents the evaluation of the recall of phrase term extraction for each part-of-speech template. The analysis of the results shows that some patterns in some subject areas almost fail to identify true terms. This may indicate that the structure of phrasal terms depends not only on the general patterns of morphosyntactic design, but also on the subject specificity of texts.

This finding highlights the need to adapt patterns to specific domains or employ dynamic term extraction methods that consider context and semantic characteristics of the subject domain.

4.3.2 Preliminary filtration

The extracted phrases undergo a prefiltering stage, which includes two main steps: cleaning by adjusting POS tags and removing stop words.

Table 4: Recall results of phrase candidate extraction for each part-of-speech template. The number corresponds to the part-of-speech template number from Table 2.

Data set	Lang	Domain	№ POS-tag template											Total Recall
			1	2	3	4	5	6	7	8	9	10	11	
ACL-RD-TEC 2.0	**en**	**annotator 1**	38,14	35,7	0,75	0,75	1,13	0,19	3,83	0,06	1	1,25	0,13	**82,93**
		annotator 2	39,38	37,82	0,69	0,5	1,15	0,14	3,4	0	0,69	0,96	0,14	**84,87**
ACTER	**en**	**corp**	34,91	35,78	1,15	0,14	0,29	0,14	1,01	0,14	8,76	4,17	0,29	**86,78**
		equi	52,82	23,66	1,88	0	0,13	0,13	5,65	0	2,15	0,94	0,54	**87,9**
		wind	53,67	26,69	2,18	0,4	0,6	0,3	4,46	0	1,49	0,99	0,2	**90,98**
		htfl	33,51	43,76	1,97	1,25	2,76	0,2	2,04	0	0,39	1,18	0,46	**87,52**
	fr	**corp**	51,04	18,99	2,37	1,04	0	0	0,89	0	-	-	-	**74,33**
		equi	45,13	15,71	3,38	2,39	0,4	0,2	0,6	0	-	-	-	**67,81**
		wind	44,55	23,48	1,85	2,4	0,18	0,74	1,85	0,37	-	-	-	**75,42**
		htfl	69,2	8,2	0,7	2,27	0,44	0	0,7	0	-	-	-	**81,51**
	nl	**corp**	59,54	14,64	1,54	0,77	0,39	3,28	-	-	-	-	-	**80,16**
		equi	57,54	0,25	0	1,01	0,75	4,27	-	-	-	-	-	**63,82**
		wind	69,16	4,34	0	0	0	2,41	-	-	-	-	-	**75,91**
		htfl	78,13	0,79	0	0,13	0	2,24	-	-	-	-	-	**81,29**

Cleanup by adjusting POS tags. This step removes phrases ending in *PROPN* (proper nouns) or *NOUN* (noun) if re-tagging changes their part of speech to something other than *NOUN*, *PROPN*, or *VERB*. The SpaCy model considers the context in which a word occurs when marking it up, which may cause its tag to change. For example, the phrase "European Central" may be initially tagged as *PROPN* + *PROPN*, but after re-evaluation, it could be reclassified *PROPN* + *ADJ*. Since the templates in Table 2 do not provide for a combination ending in *ADJ*, such a phrase is considered incomplete and is deleted. Repeated tagging helps identify incomplete expressions and reduce the number of candidates. This filter is applied to English texts only.

Filtering by stop-words. This step removes phrases containing stop words unless the stop word is *PROPN* (proper name), *ADP* (preposition), or *DET* (article). Although the patterns in Table 2 allow prepositions and particles in phrases, such elements may occur only in the middle of a phrase (when the *ADP* is not at the beginning or end of the phrase). As for *PROPN* proper nouns, some names may contain common words. For example, in *"New York"*, the word *"New"* is a stop-word, but it represents part of the city name and should not be removed. This filter helps eliminate phrases containing common words, such as *"other illegal activities"*, *"possible cases"*, *"more effective"*, which reduces the noise in the data.

This filtering helps to remove certain candidate categories, which minimizes the noise in the data and improves the accuracy of term extraction.

4.3.3 Raw and rectified frequency calculation

It is assumed that if neighboring words frequently co-occur in the text, there is a high probability that they form a stable collocation. For a more accurate analysis, two types of frequency are calculated: raw and rectified, as described in [25].

Raw frequency (*F_raw*) represents the total count of a phrase's occurrences in the text. This frequency indicates how often a given combination of words occurs in the source text.

Rectified frequency (*F_rec*) represents how often the target phrase appears in the text, excluding instances where it is part of longer phrase. To compute the rectified frequency, one must consider the sum of the rectified frequencies (*F_l*) of longer phrases that contain the target phrase. The rectified frequency is computed using the following formula:

$$F\_rec = F\_raw - F\_l \quad (1)$$

At this stage, phrases are sorted in descending order by length, as longer phrases containing the target phrase must be considered to compute the rectified frequency. To enhance the accuracy of rectified frequency calculations, an additional corpus of texts related to the target domain can be used, facilitating a more precise detection of true phrase boundaries.

The phrase extraction process is divided into two stages. In the first stage, phrases with the highest frequencies are selected based on thresholds: raw frequency (*F_raw*) greater than 9 or rectified frequency (*F_rec*) greater than 3. These thresholds were optimized via experimental tuning to achieve optimal results In the second step, the extracted phrases are grouped by word position, and additional filtering is performed based on frequency comparison. These steps are necessary to identify the most strongly related words that form stable phrases.

4.3.4 Grouping phrases by common word positions

Phrases are grouped based on the presence of common word positions. In the grouping process, it is possible that phrases that do not directly share common positions can be grouped together if there is an intermediate phrase that shares common positions with two other phrases.

An example of such grouping is shown in Table 5, where one of the groups contains phrases with overlapping word positions. For example, the candidate phrase *"Austrian-led network"* does not share word positions with the phrase *"European partners against corruption"* but overlaps with the phrase *"Network European*

Table 5: Example of a group of candidate phrases containing words with common positions. The underlined candidate will be categorized as a phrase because it has the highest rectified and raw frequency values. The overlapping positions of words with the underlined candidate are shown in bold.

Candidates	***F_raw***	***F_rec***	**Word position index**
austrian-led, network	1	0	19534,19535
european, partners	3	0	**19536,19537**
austrian-led	1	0	19534
network, european, partners	1	0	19535,**19536,19537**
partners, against, corruption	3	0	**19537,19538,19539**
austrian-led, network, european, partners, against, corruption	1	1	19534,19535,**19536,19537,19538,19539**
network, european, partners, against, corruption	1	0	19535,**19536,19537,19538,19539**
<u>european, partners, against, corruption</u>	<u>3</u>	<u>2</u>	<u>19536,19537,19538,19539</u>

partners" in terms of word positions. As a result, phrases that do not directly share word positions can still belong to the same group.

This step is necessary to remove incomplete or partial phrases. In each group, the phrases with the highest rectified frequency (or raw frequency if the rectified frequency is 1 or 0) are selected. If the rectified or raw frequency of a phrase is strictly greater than 1, such a phrase is accepted. The phrase, as well as all candidates in the group that share common word positions with the accepted phrase, are then removed from the group. The process is repeated until no phrases remain in the group.

In the example shown in Table 5, all candidates that have common word positions with the candidate *"European partners against corruption"* are removed from the group. After that, two candidates, *"Austrian-led network"* and *"Austrian-led"*, remain in the group, but their frequencies do not exceed 1, so they are also removed. It is important to note that if a candidate is removed from one group, it will not be removed from other groups.

This approach helps minimize the number of candidates and highlights the most coherent and meaningful phrases.

In general, the phrase extraction method from text includes several interrelated steps, each of which contributes to improving the accuracy and quality of the extraction of relevant phrases. In the first stage, part-of-speech templates are applied for initial filtering and extraction of a set of potential candidate phrases to cover a wide range of possible terms. In the prefiltering stage, less informative phrases are eliminated, which helps to reduce the number of candidates and improve precision. Using a frequency-based filter, incomplete phrases are removed, and their boundaries are accurately identified, which helps eliminate noisy data and improve the quality of the remaining candidates.

4.4 TopicScore filter

The TopicScore metric presented in [21] is used for the semantic filtering of extracted unigrams and multi-word expressions. Unlike the original approach, where it was applied exclusively to phrases, in this paper, TopicScore is used for both multi-word expressions and unigrams.

The metric is defined as the cosine similarity between the candidate embedding (*w_c*) and the sentence embedding (*w_sent*) in which it occurs. The higher the similarity value, the greater the probability that the candidate is a term relevant to the given context. In this study, a candidate is classified as a term if the cosine similarity exceeds a threshold value of 0.4.

$$TopicScore = \frac{w\_c * w\_sent}{\|w\_c\| * \|w\_sent\|} \quad (2)$$

Embeddings are computed using the BERT model, which generates context vectors for sentences. The use of context embeddings avoids the out-of-vocabulary (OOV) problem characteristic of static vector-based methods.

The TopicScore metric facilitates the selection of terms that are most relevant to the subject domain and also allows the identification of the most informative candidate terms.

4.5 Abbreviation extraction

For abbreviation extraction from text, it is important to preserve the original case. As a basic strategy for abbreviation extraction, the rule that unigrams containing two or more uppercase letters are treated as abbreviations has been used. However, this method has some limitations, as it is vulnerable to cases where a word is entirely in uppercase, such as in headings or sections of text. An example of such a case is the word *"ABSTRACT"*, which would be misidentified as an abbreviation under this rule.

The keyword extraction method presented by the authors of [24] considers character case as one of the features used to identify significant keywords in the text. In addition, some names may contain multiple capital letters within a single word (e.g., *"YouTube"*), allowing the approach to detect not only acronyms but also named entities.

4.6 Extracting named entities

To extract named entities, POS tagging is applied to the text while preserving the original case. Named entity extraction is performed using the *PROPN** and *[PROPN, ADP]** patterns. This approach accounts for the possibility that prepositions may occur in the names of many entities, but only if they appear in the middle of the sequence. This allows the selection of sequences that correspond to typical named entity structures, preserving their integrity.

Unlike phrase extraction methods, which cover all possible word sequences, this approach focuses on extracting complete and continuous word sequences labeled with the *PROPN* tag. This allows for more accurate identification of text fragments such as names of organizations, geographical entities, and other named categories. Thus, the method focuses on identifying structures that represent complete named entities, helping to improve extraction accuracy.

Abbreviations and named entities are neither semantically nor statistically filtered. This is because the frequency of such elements may be too low, and their semantic meaning may not be sufficiently unambiguous. For example, abbreviations are often sequences of letters that may represent long names and be perceived as random character strings. In the case of named entities, such as people's names, they may be associated with a variety of activities, making their contextual meaning less specific and more universal. As a result, such entities may appear in different domains and may not always be clearly associated with a specific topic or field.

4.7 Fine-tuning approach

There are several key factors to consider when extracting multiword expressions. First, the choice of part-of-speech combination patterns is important because it determines

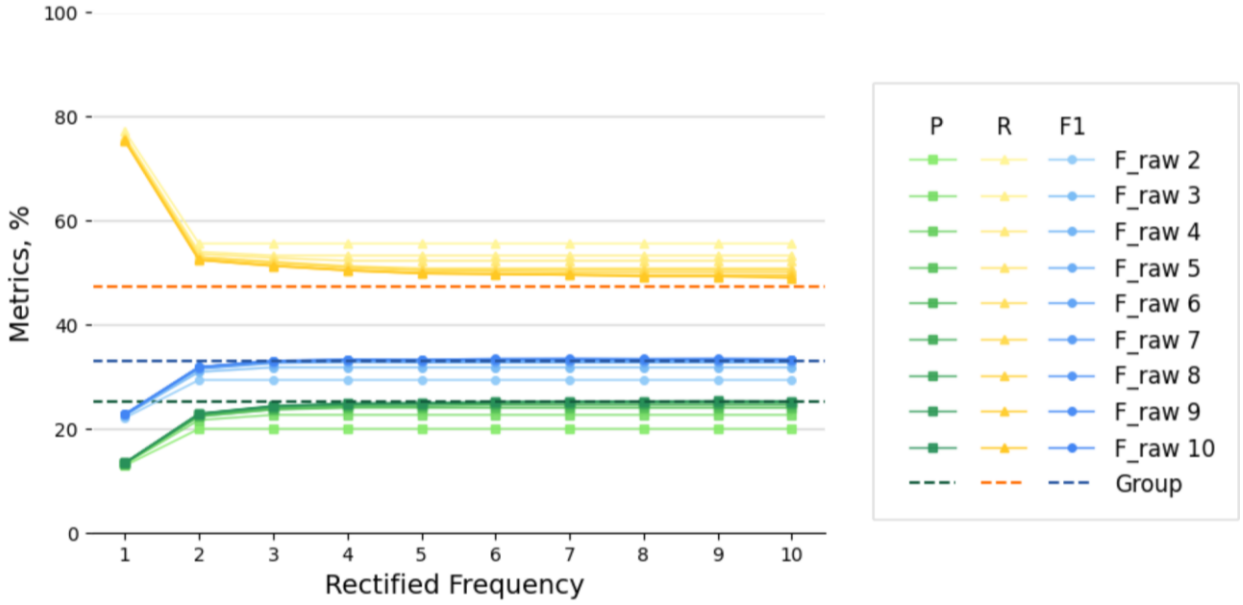


Figure 3: Effect of raw and rectified frequency thresholds on Precision(P), Recall (R) and F1 score on corp (en) domain from ACTER. The dotted line Group indicates phrase retrieval rates only at the filtering stage with grouping by common word positions, excluding phrases.

the recall and the number of extracted terms. Too many candidates may increase the computational burden of subsequent processing steps, including filtering and analysis.

Second, setting thresholds for raw and rectified frequencies is an important aspect. Lowering these thresholds may reduce the accuracy of term extraction, while increasing them may reduce recall. The definition of thresholds also depends on the size of the corpus and the texts. For the ACTER corpus, thresholds have been set raw frequency above 9 and rectified frequency above 3. If the corpus or texts are too small, it is recommended to lower the frequency thresholds. As in the case of the ACL RD-TE 2.0 enclosure, where the thresholds were set: raw frequency above 2 and rectified frequency above 1.

Figure 3 shows the variation of the indicators depending on the frequency thresholds. The evaluation of the indicators was performed considering the phrases extracted in the grouping phase based on common word positions. Thus, the phrases obtained by grouping and the phrases extracted based on frequency thresholds were compared with the list of true multiword terms.

Table 6: Recall (R, %) of extracted phrase terms when using grouping and the effect of frequency thresholds. The Group column contains the recall values obtained at the filtering stage by grouping phrases by common word positions. The Frequency thresholds column presents the gain in recall due to additional phrase extraction using frequency thresholds.

Data Set	lang	domain	Group		Frequency thresholds		Total recall for Phrases
			N	R	N	R	
ACTER	en	corp	1300	47,13	+169	+4,02	51,15
		equi	995	44,22	+51	+1,88	46,1
		wind	1741	43,25	+309	+7,74	50,99
		htfl	1615	31,6	+138	+2,76	34,36
	fr	corp	1367	41,39	+142	+1,34	42,73
		equi	729	35,98	+18	+0,2	36,18
		wind	1290	42,33	+188	+4,81	47,14
		htfl	876	25,13	+68	+2,18	27,31
	nl	corp	1134	42,58	+53	+0,39	42,97
		equi	648	32,91	+16	+1,26	34,17
		wind	1045	33,01	+45	+2,89	35,9
		htfl	948	31,09	+31	+0,53	31,62
ACL-RD- TEC 2.0	en	anntator1	752	22,58	+2534	+40,65	63,23
		anntator2	1041	23,36	+3686	+43,55	66,91

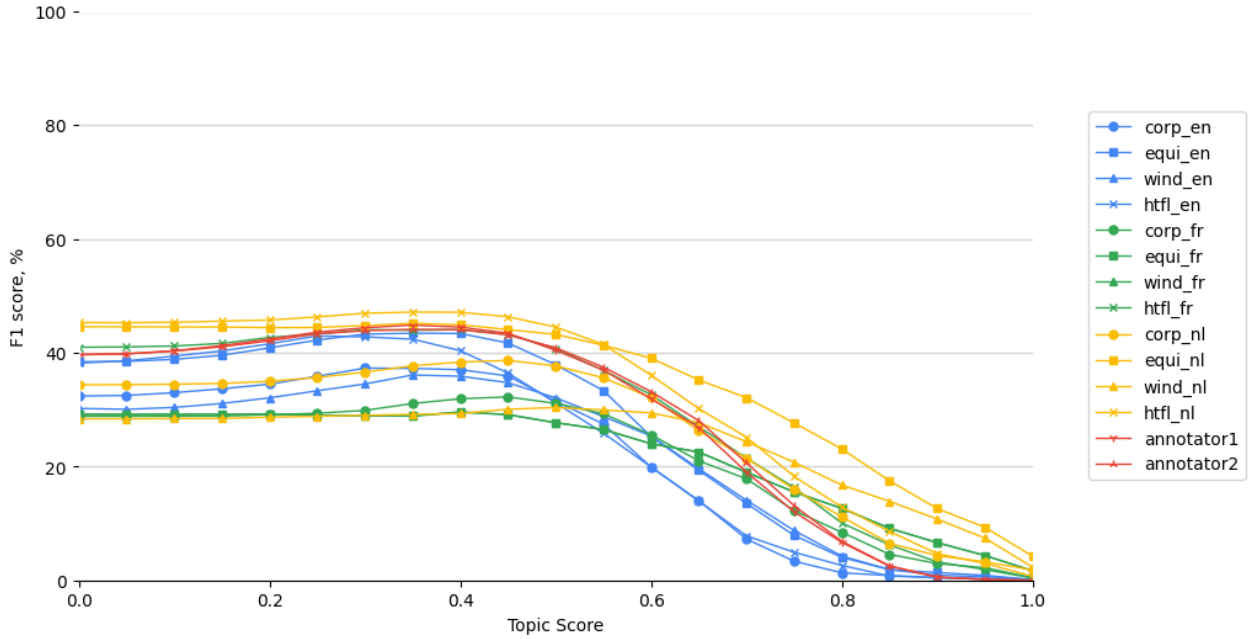


Figure 4: Variation of F1 score at different TopicScore thresholds on the ACTER and ACL datasets

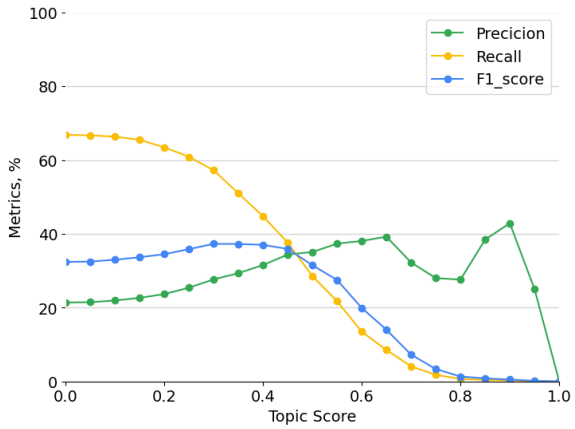


Figure 5: Variation of scores at different TopicScore thresholds on corp(en) domain from ACTER

The analysis showed that there is little change in recall at a rectified frequency (*F_rec*) above 4. Precision and the F1-score remain stable at rectified frequency values above 3. Therefore, a threshold of 3 for *F_rec* was chosen for the ACTER dataset.

For raw frequency, the indices stop changing at values above 5, meaning that setting the threshold in the range of 5 to 10 has a negligible impact. However, a threshold of 9 was chosen to improve precision, as the raw frequency does not reflect the significance of the terms as accurately as the rectified frequency. Lowering the thresholds leads to a decrease in precision, which may negatively affect the quality of the extracted phrases, as it may introduce errors in boundary detection or the selection of unrelated words.

Table 6 presents the effect of frequency thresholds on recall. The data shows how much recall increases when additional frequency filtering is used. If only thresholds are considered, phrases that may have already been retrieved in the grouping step may be extracted.

Frequency threshold filtering has a more significant impact on small corpora such as ACL. In the case of large texts, threshold filtering gives only a minor addition to the core set of extracted phrases. However, this approach is effective for small texts where grouping-based filtering did not provide a significant gain in the number of extracted terms.

For generating contextual vectors of unigrams, phrases and sentences, the model *"sentence-transformers/all-MiniLM-L6-v2"* for English and *"sentence-transformers/paraphrase-multilingual-mpnet-base-v2"* for French and Dutch is used. The TopicScore method uses a similarity threshold value (0.4) for both unigrams and phrases. If the cosine similarity between a unigram or phrase vector and a sentence vector exceeds this threshold, such a unit is classified as a term.

The selection of the optimal threshold for TopicScore is based on an analysis of the F1-score, precision, and recall metrics at different threshold values. Figure 4 shows that F1-score reaches its maximum value at TopicScore in the range of 0.3-0.4, and its variation in this interval is insignificant. The mean value of the threshold at which F1-score was maximized is 0.375.

However, TopicScore has a significant impact on the balance between recall and precision of candidate term selection, which is illustrated in Figure 5. Here is an example of performance variation as a function of the threshold. Since precision was prioritized over recall at this stage of the study, a threshold of 0.4 was chosen as the optimal value.

The list of stop words was taken from the GitHub repository *"term-extraction-project/stop_words"* for English, while for French and Dutch, data from the *"stopwords-iso"* repository was used.

5 Evaluation method

Precision, Recall, and F1-score metrics are used to evaluate the effectiveness of the developed T-Extractor approach. Recall (R) characterizes the proportion of correctly extracted terms out of all relevant terms. Precision (P) indicates the percentage of extracted terms that are correct. The F1-score represents the harmonic mean of Precision and Recall, providing a balanced assessment of the model's quality.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

For a more in-depth analysis of the proposed approach's performance, the model was evaluated at four key stages of term and named entity extraction: (1) after candidate extraction, (2) after semantic filtering with TopicScore, (3) after adding abbreviations, and (4) at the final stage, after extracting PROPN sequences.

The first step involves extracting and pre-filtering candidates, covering all steps before applying TopicScore filtering, and is labeled Extract Candidates. The second step reflects the result after TopicScore filtering and is referred to as TopicScore Filter. The third step shows score changes after adding extracted words based on the abbreviation rule and is labeled Abb Extract. The final step involves the extraction of sequences with PROPN tag, more related to named entities, and is called NE Extract. This step shows the results after adding extracted candidates using the algorithm for extracting named entities.

To better understand the quality of term extraction in the corp domain from the ACTER (en) dataset, the results computed at each stage are presented, focusing on evaluating the extraction performance of unigrams (uni), phrases (mwe), and generic terms (All). Unigrams and multi-word expressions were evaluated separately, as their extraction methods differ significantly. Extracted unigrams (uni) were compared to true single-word terms, while extracted phrases (mwe) were compared to true multi-word terms. This approach allows a detailed analysis of the performance of each extraction step and evaluates its impact on the overall quality of the extracted terms.

To evaluate the effectiveness of the two rules for extracting abbreviations and proper noun sequences, the precision of term and named entity identification was measured. The proportion of extracted abbreviations was determined by comparison with a set of true terms. Similarly, the identified named entity sequences were compared to a reference list to calculate their proportion among the true terms.

Finally, the final F1-score for the T-Extractor term extraction method was compared with the results of other term extraction methods, both supervised and unsupervised, such as HAMLET, GPT-3.5-Turbo, PromptATE, TALN-LS2N, BERT3, BERT6, NMF, UA, and UA1.

6 Results

The results of term and named entity extraction using T-Extractor are presented in Table 7. The first stage (Candidate extraction), which includes candidate extraction and pre-filtering, identifies, on average, about 70% of the true terms, with a precision of 24% and an F1-score of 35%. At this stage, the F1-score for T-Extractor already outperforms the results of many unsupervised approaches presented in Table 10.

After applying filtering using Topic Score, the F1-score increases by 3.1% on average. Precision increases

Table 7: Results (%) of extracting terms and named entities using the T-extractor annotator.

Data set	Lang	Domain	Candidate extract			Topic Score filter			Abb extract			NE extract		
			P	R	F1	P	R	F1	P	R	F1	P	R	F1
ACL-RD-TEC 2.0	en	annotator 1	27,3	72,1	39,6	36,0	58,1	44,5	36,0	59,5	44,8	35,0	61,2	44,5
		annotator 2	27,2	73,0	39,6	35,1	59,1	44,0	35,0	60,3	44,2	33,8	61,8	43,7
ACTER	en	Corp	21,4	67,0	32,4	31,6	44,8	37,0	31,4	46,7	37,6	31,5	55,3	40,1
		Equi	26,6	69,5	38,4	40,2	47,1	43,4	40,3	47,6	43,7	41,6	58,2	48,5
		Wind	19,6	67,3	30,4	29,8	45,1	35,9	29,7	47,7	36,6	28,6	58,3	38,4
		HTFL	28,2	58,4	38,0	42,8	38,2	40,3	44,8	43,0	43,9	43,7	48,7	46,0
	fr	Corp	18,7	64,2	28,9	24,2	47,7	32,1	24,7	50,4	33,2	25,4	53,4	34,5
		Equi	18,9	63,6	29,1	21,7	46,6	29,6	21,7	47,0	29,7	22,7	51,6	31,5
		Wind	14,0	67,3	23,2	16,6	55,9	25,6	17,0	58,9	26,4	17,4	63,0	27,3
		HTFL	30,1	64,0	40,9	41,8	46,5	44,1	42,9	49,4	45,9	42,5	50,8	46,3
	nl	Corp	22,4	73,4	34,4	28,4	59,0	38,4	28,8	60,5	39,0	29,0	63,8	39,9
		Equi	32,0	72,9	44,4	35,1	62,2	44,9	35,2	62,6	45,1	35,6	65,9	46,2
		Wind	17,5	75,4	28,3	19,4	60,3	29,3	19,7	61,6	29,8	20,8	68,5	32,0
		HTFL	32,2	76,3	45,3	39,9	57,5	47,1	40,9	60,3	48,7	40,4	62,2	49,0
Average			24,0	68,9	35,2	31,6	52,0	38,3	32,0	54,0	39,2	32,0	58,8	40,6

Table 8: Results (%) of term and named entity extraction by processing stage for the domain "Corruption" (Corp) from the ACTER (en) dataset divided into unigrams (Uni), phrases (MWE) and all terms (All).

Dataset	Corp	Candidate extract			Topic Score filter			Abb extract			NE extract		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
ACTER (en)	Uni	19,6	88,1	32,1	35,6	48,2	41,0	34,8	53,0	42,0	35,0	60,8	44,4
	MWE	23,8	52,6	32,8	29,0	42,4	34,4	29,0	42,4	34,4	29,1	51,6	37,2
	All	21,4	67	32,4	31,6	44,8	37,0	31,4	46,7	37,6	31,5	55,3	40,1

significantly by about 7.6%, while recall drops by 16.9%. After Topic Score filtering, the F1-score averages 38.3%.

In the next stage (Abb extract), after adding the candidates extracted using abbreviation rules, both precision and recall increase slightly by about 1–2%. The average F1-score in this step is 39.2%.

The final stage involves extracting PROPN sequences (NE extraction), which increases the F1-score by 1.4%. In summary, after all the steps of term and named entity extraction, the average precision is 32%, the average recall is 58.8%, and the average F1-score is 40.6%.

Table 8 presents the results of term and named entity extraction for the *"corruption"* (Corp) domain from the ACTER (en) dataset, categorized into unigrams (uni), phrases (mwe), and all terms across different processing steps. These data show the dynamics of improvement in the results of term and named entity extraction at each processing stage for different types of terms. The analysis shows that at almost all stages, phrase extraction performed worse than unigram extraction.

In the candidate extraction stage, the F1-scores for unigrams and phrases were almost similar, but the differences in precision and recall were significant. Higher recall and lower precision were observed for unigrams than for phrases. This indicates that with further filtering, the recall for phrases will decrease significantly, which in turn may negatively affect the performance of the model.

After applying the Topic Score filter, the F1-score for unigrams increased significantly and became higher than that for phrases. Although unigram recall dropped significantly, precision increased by 16%, indicating more

Table 9: Precision (%) evaluation of extracting relevant terms using rules for finding abbreviations and proper noun sequences.

Dataset	Lang	Domain	Abbreviation	Named Entities
ACL-RD-TEC 2.0	en	annotator 1	45,89	36,49
		annotator 2	46,56	35,01
ACTER	en	Corp	39,55	40,66
		Equi	64,00	59,53
		Wind	34,81	32,05
		HTFL	73,56	55,63
	fr	Corp	37,75	54,64
		Equi	57,41	46,84
		Wind	28,31	30,82
		HTFL	72,14	57,35
	nl	Corp	47,11	44,71
		Equi	62,77	61,44
		Wind	57,43	49,61
		HTFL	80,00	55,38
Average			53,4	47,2

effective filtering. The F1-score for phrases increased by only 1.6%, which confirms that Topic Score filter works more effectively for unigrams.

After the abbreviation extraction step (Abb extract), the F1-score for unigrams increased slightly, which is because this step is mainly focused on extracting specific unigrams. In the case of unigrams, the precision slightly decreased but the recall increased, which indicates that this stage extracted meaningful words that can be both terms and named entities.

At the PROPN sequence extraction (NE extraction) stage, the overall F1-score for this domain reached 40.1%. The F1-scores for unigrams and phrases also increased, indicating the importance of this stage for the extraction of meaningful lexical units. Recall for unigrams and phrases is at around 50%, while precision is around 30%. Despite this, the performance for phrases remains lower than for unigrams, indicating the need to optimize the approach for phrase extraction.

Thus, although phrase extraction significantly lags unigram extraction at almost all stages, the term and named entity extraction process demonstrate performance improvements at each step. Filtering techniques such as Topic Score filter show better results for unigrams, while phrases require further optimization.

Table 9 shows the precision evaluation of extracting terms and named entities using two simple rules: for extracting abbreviations and sequences of proper nouns.

The average extraction precision using the abbreviation rule is 53.4%, indicating the ability of the approach to extract relevant words with over 50% precision. However, the approach performed poorly in the wind domain in English and French.

The average precision of extracting relevant words using the rule for noun sequences is 47.2%. Despite this, the approach showed efficiency and high retrieval precision in several cases. However, the rule was less successful for the ACL RD-TEC 2.0 dataset, as well as for the wind domain in all languages of the ACTER corpus.

Table 10 shows the F1-score results for the T-Extractor annotator compared to other supervised and unsupervised term extraction methods. Figure 6 presents the F1 score results for the HAMLET [27], T-Extractor, and NMF [20] annotators applied to the ACTER corpus in three languages. It is evident that T-Extractor has significantly reduced the gap between supervised and unsupervised approaches, closely approaching HAMLET in several cases. Its performance in the Corp domain was comparable to that of the supervised method. However, in the equi domain for French and Dutch, a notable performance gap remains, likely due to language-specific characteristics that were not accounted for during adaptation. Nevertheless, T-Extractor outperformed the unsupervised NMF approach, achieving a higher F1 score. A detailed comparison of its performance with supervised and unsupervised methods is presented below.

The T-Extractor method shows superiority over most unsupervised approaches when applied to the ACTER dataset. However, on the ACL RD-TEC 2.0 dataset, its F1-score (44.5%) is inferior to that of the UA method (50%). At the same time, the re-evaluated version of the UA approach, denoted as UA1, showed an average F1 score 8.3% lower on the ACTER dataset compared to the T-Extractor method.

Compared to the NMF method [20], the T-Extractor annotator performs better across all domains and languages. For example, on the corp_en domain, T-Extractor achieves 40.12% while NMF shows 25.72%. A similar difference is observed across all other languages and domains, confirming the effectiveness of T-Extractor in unsupervised term extraction tasks. However, on the Equi(fr) domain, the difference between T-Extractor (31.5%) and NMF (27.2%) is only 4.3%, suggesting low language- and domain-specific dependency. This highlights the need for further improvements to the T-Extractor method, as its performance may vary depending on the context.

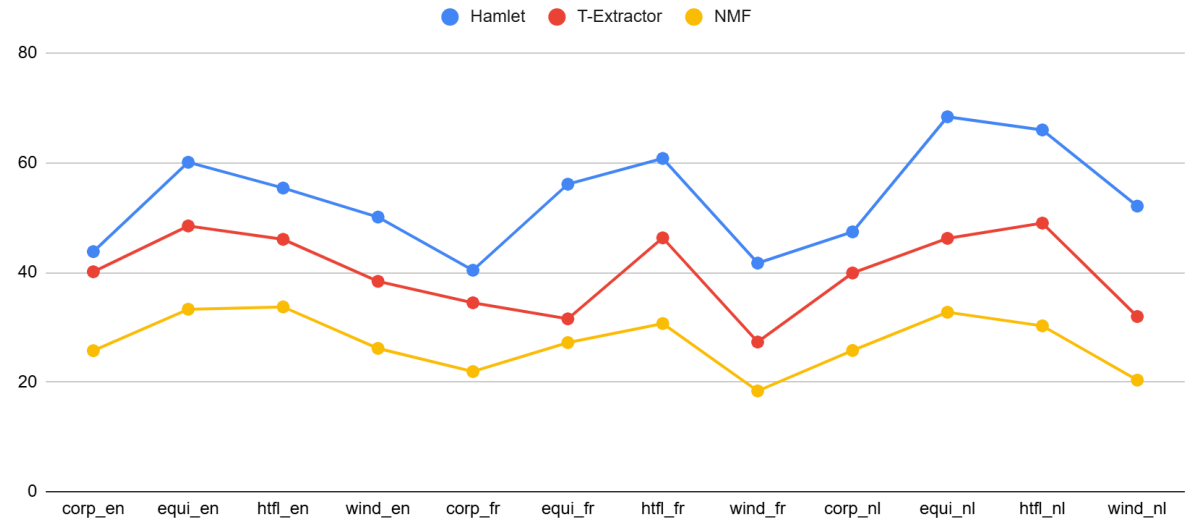


Figure 6: Comparative analysis of F1 score (%) for HAMLET, T-Extractor, and NMF in term and named entity extraction on the ACTER corpus

Table 10: Comparison of F1 scores (%) of the T-Extractor annotator with other supervised and unsupervised methods on ACTER data set.

Annotators	**en**				**fr**				**nl**			
	Corp	**Equi**	**Wind**	**HTFL**	**Corp**	**Equi**	**Wind**	**HTFL**	**Corp**	**Equi**	**Wind**	**HTFL**
Supervised												
HAMLET [27]	43,8	60,1	50,1	55,4	40,4	56,1	41,7	60,8	47,4	68,4	52,1	66,0
GPT-3.5-Turbo [29]	31,4	49,7	32,5	55,6	-	-	-	-	-	-	-	-
promptATE (Llama 2-Chat) OF#3 [30]	-	-	-	51,4	-	-	-	47,8	-	-	-	55,4
TALN-LS2N [28]	-	-	-	46,66	-	-	-	48,15	-	-	-	-
BERT3 [28]	32,8	42,2	34,8	45,7	29,6	36,4	27,8	48,4	-	-	-	-
BERT6 [28]	35,5	41,6	28,5	44,1	41,7	16,2	16,4	36,9	-	-	-	-
Unsupervised												
NMF [20]	25,7	33,3	26,1	33,7	21,9	27,2	18,4	30,7	25,8	32,7	20,4	30,3
UA1 [22]	24,3	28,9	29,5	32,7	-	-	-	-	-	-	-	-
T-Extractor	40,1	48,5	38,4	46,0	34,5	31,5	27,3	46,3	39,9	46,2	32,0	49,0

As for the supervised methods, the HAMLET annotator [27] significantly outperforms T-Extractor. However, compared to other supervised methods, T-Extractor achieves results in some domains that are either superior or not significantly lower. This indicates that T-Extractor, despite its unsupervised nature, is an effective tool for term extraction and its results may be comparable or even superior to supervised approaches in some cases.

The HAMLET annotator, a supervised method, outperforms T-Extractor across all metrics. This is particularly evident in Figure 6, where HAMLET performs well, especially on domains related to French and Dutch languages, where its performance is significantly higher compared to the other methods. However, the difference between HAMLET and T-Extractor is not always so large. For example, in the corp_en domain, T-Extractor performs competitively, achieving 40.1% compared to HAMLET's 43.8%. This indicates that T-Extractor can show results comparable to supervised methods in certain contexts.

The GPT-3.5-Turbo model [29] performs better than T-Extractor, especially in the HTFL domain (ACTER en), where its F1-score is 9.6% higher than T-Extractor. Nevertheless, overall, T-Extractor's performance is not significantly lower than that of the supervised GPT-3.5-Turbo method. For example, in the corp and wind domains, the T-Extractor method outperforms the GPT-3.5-Turbo method by 8.7% and 5.9%, respectively. In the equi domain, F1 performance is almost identical, with a difference of only 1.2% in favor of GPT-3.5-Turbo.

The promptATE (Llama 2-Chat, OF#3) [30] method outperforms T-Extractor in all three languages, but its performance in other domains remains uncertain. It should be noted that on the HTFL(fr) dataset, the differences in results are insignificant. For a more accurate and objective evaluation of promptATE's performance, it is necessary to analyze its performance on additional domains and datasets.

The TALN-LS2N method [28] also outperforms T-Extractor, but the difference in results is not significant. However, TALN-LS2N requires a significant amount of labeled data, which limits its applicability when there is a lack of high-quality annotation.

As for BERT3 and BERT6 [28], their performance is on average slightly inferior to T-Extractor, especially in English. However, they perform better on some other languages, e.g., BERT3 shows a slight superiority over T-Extractor on French. BERT6 significantly outperforms T-Extractor on the Corruption (French) domain, but is inferior on the other domains, indicating that its performance is heterogeneous across languages and domains.

To evaluate the statistical significance of the T-Extractor results, a paired t-test was conducted with the NMF approach. This annotator was chosen for comparison with T-Extractor because both are unsupervised methods and were tested on the largest number of texts compared to other approaches.

The results of the paired t-test showed that the t-statistic was 12.31 and the p-value was 8.96×10^{-8}. Since the p-value is significantly lower than the standard threshold of 0.05, the difference between the methods is statistically significant, thus rejecting the null hypothesis that there is no difference in their quality. This indicates that T-Extractor significantly outperforms NMF in terms of the F-measure, demonstrating superior performance in term and named entity extraction.

In general, the T-Extractor annotator shows competitive results in term extraction, outperforming many unsupervised methods. Despite lagging significantly behind the supervised method HAMLET, T-Extractor achieves comparable results in some contexts, such as in the corp_en domain. Overall, the performance

of T-Extractor can be close to that of other supervised methods such as GPT-3.5-Turbo, and in some domains and languages even outperforms them.

7 Discussion

The advantage of T-Extractor over other annotators is the integration of statistical and semantic approaches for term extraction, as well as its independence from labeled data. The T-Extractor exhibits high recall in the candidate extraction phase, ensuring that more potential terms are retained in subsequent filtering steps compared to alternative methods. The use of part-of-speech patterns instead of n-grams, as in approaches such as NMF, TALN-LS2N, or BERT-based models, contributes to extracting more meaningful word combinations and improves the accuracy of the method. In addition, the T-Extractor can identify longer terminological candidates rather than being limited to unigrams and pentagrams.

The proposed methodology for customizing part-of-speech patterns provides greater flexibility in forming terminological expressions and reduces the cost of manually enumerating possible part-of-speech combinations. This approach demonstrates advantages over UA, which is limited exclusively to noun phrases. In addition, the noun chunks mechanism does not always efficiently identify phrase boundaries, which was noted by the authors when implementing the UA1 method, leading to incorrect identification of terminological candidates. In the T-extractor, term boundaries are determined based on rectified and raw frequency measures, which increases its efficiency when working with large corpora.

An additional factor affecting the efficiency of the T-Extractor is its improved text preprocessing and filtering system. In particular, the use of multi-level cleaning mechanisms in the candidate extraction stage, the setting of spaCy to avoid splitting multiword terms with hyphenation, and the preservation of the original case during POS-tagging have contributed to minimizing noise. For example, in TALN-LS2N, the candidate filtering step is described in less detail: the authors only exclude a limited set of undesirable classes, such as words starting with conjunctions and pronouns. In the GPT-3.5-Turbo and promptATE methods, where candidates are generated automatically, a check for their presence in the source text is applied, but these approaches show a tendency to select common words. In this context, additional cleaning of stop words or applying semantic filtering could improve the relevance of the extracted terms.

The use of semantic filtering in the T-Extractor allowed for the extraction of more topic-relevant candidates. In UA, this mechanism was applied only to multi-word expressions, but not to unigrams, which probably negatively affected the quality of term extraction in its UA1 version tested on the ACTER dataset. NMF also lacks semantic filtering, which may have reduced the performance of the method.

Compared to HAMLET, the key advantage of T-Extractor is that it does not require annotated data, but it is inferior in candidate extraction performance. Like HAMLET, T-extractor uses various features to identify terms including statistical, linguistic and semantic characteristics. It is possible that the use of a hybrid approach combining different features, heuristics and filtering methods allowed the T-extractor running in unsupervised mode to achieve closer performance to supervised methods than other unsupervised approaches.

One of the key features of T-Extractor is its ability to extract unigrams more efficiently than multi-word terms. This is because the quality of phrase extraction largely depends on the correct definition of phrase boundaries. The proposed approach is based on frequency characteristics, which may reduce its efficiency when processing small texts. In addition, T-Extractor excludes phrases that occur only once in the text, which potentially affects the recall of term extraction. Unlike multi-word expressions, single-word terms do not require additional boundary refinement, ensuring their higher recall.

Unigrams are filtered more efficiently than phrase expressions in the Topic Score filter. This may be because it is easier to form a meaningful vector representation for unigrams compared to phrases, or due to their higher recall, which results in fewer relevant candidates being retained among multi-word terms. This stage is also sensitive to the choice of model for generating vector representations of the context, which has a direct impact on the quality of term extraction.

In the step of adding abbreviations, the efficiency of term extraction depends on the correct text case. Additionally, adding abbreviations primarily enhances unigram extraction results, as confirmed by the test results.

Named entity extraction significantly improves both unigram and phrase extraction, particularly in texts where such entities appear infrequently. This approach helps to increase both the recall and precision of phrase extraction, making the process more accurate and comprehensive. An interesting observation is that some named entities and abbreviations, extracted along with unigrams and phrases in the Candidate Extract step, may be filtered out during the Topic Score step. However, additional extraction rules enable the recovery of filtered candidates, ultimately enhancing overall annotation results.

Analysis of the ACTER corpus data presented in Figure 6 reveals patterns related to domain and language dependency. In some domains, such as Equi and HTFL, annotators perform well, whereas in others, such as corp and wind, their performance declines significantly. Additionally, in French, the HAMLET, NMF, and T-Extractor methods yielded lower results than in English and Dutch, confirming the language dependency of these approaches.

The average Pearson correlation between the annotators' results was 0.797, indicating a strong and positive correlation. This may also indicate the specificity of text structure for different domains and languages. Considering these factors may play a key role in understanding term features and improving term extraction in various contexts.

Thus, the study provided answers to the research questions posed.

Firstly, the impact of combined features on the term extraction process remains significant, despite the continuous advancement and improvement of deep learning models. This suggests that despite the availability of powerful neural network-based methods, traditional linguistic and statistical approaches remain crucial in terminology processing. Moreover, this observation supports the hypothesis that applied linguistics is unlikely to become solely the domain of deep learning research. Rather, it is expected to remain an interdisciplinary field at the intersection of linguistics, statistics, and computational methods.

Secondly, the analysis demonstrated that in the absence of annotated training data, the significance of utilizing the T-Extractor tool increases. This is because its methodology compensates for the lack of labeled corpora by leveraging heuristics, statistical patterns, and pre-existing knowledge about terms. As a result, automatic term extraction methods can operate effectively even with limited training data, making them valuable tools for low-resource languages and specialized domains.

8 Conclusions

This paper analyzed the performance of the T-Extractor annotator in unsupervised term extraction tasks and compared it with other methods including supervised and unsupervised approaches. The results showed that T-Extractor is a competitive tool that shows stable performance on different languages and domains.

The main advantages of T-Extractor lie in its ability to work without annotated data, which makes it suitable for text processing in resource-constrained environments. Using a combination of rules, statistical and semantic analysis, it achieved high retrieval recall. However, the annotator showed lower precision, indicating that candidate filtering mechanisms need to be improved, especially in the phrase boundary detection phase.

T-Extractor is particularly better at extracting unigrams, while phrase extraction is more difficult due to its dependence on frequency characteristics. In the Candidate Extract step, the limitation of the method manifests itself in the inability to extract rare phrases, which reduces recall. The addition of named entity and abbreviation processing steps has a positive impact on recall and precision, especially in texts with rare entities. Additional rules for recovering filtered candidates also contributed to the improvement of the metrics.

Comparison with other unsupervised methods showed that T-Extractor outperforms them in almost all domains and languages. For example, on the corp(en) domain, T-Extractor achieved 40.12% on the F1 metric, while NMF demonstrated 25.72%. However, on individual domains such as equi(fr), the difference between the methods is minimal (31.5% for T-Extractor vs. 27.2% for NMF), indicating that the method can be further optimized.

The supervised method HAMLET shows significantly better results. For example, the average difference in F1 metric between HAMLET and T-Extractor is 9.1% (in English), 14.9% (in French), and 16.7% (in Dutch). However, in some domains, such as corp (en), T-Extractor achieves performance close to supervised approaches (40.1% compared to 43.8% for HAMLET). Compared to other supervised approaches, T-Extractor showed similar results and even outperformed some in certain domains.

Currently, one of the main limitations of the T-Extractor approach is the difficulty of accurately identifying phrase boundaries and extracting low-frequency phrases. A promising direction for future research is to enhance phrase boundary detection algorithms by employing syntactic analysis techniques, such as constructing syntax trees or analyzing word dependencies. Additionally, incorporating artificial intelligence techniques could further improve the precision of phrase boundary identification.

To refine semantic filtering, another potential improvement is integrating static vector representations of words. This approach would allow the model to account not only for contextual dependencies but also for the invariant lexical meaning of terms, leading to more accurate filtering and selection.

Furthermore, the development of a classification module for extracted terms presents another promising avenue. This module could categorize terms based on multiple criteria, distinguishing domain-specific, general, and out-of-domain terms, as well as classifying them thematically according to the text's content.

In addition, classifying named entities according to the MUC-7 scheme could be incorporated, providing a more detailed and structured representation of extracted entities. It is expected that integrating such a classifier would not only enhance the quality of term extraction but also increase the significance of T-Extractor as a tool for processing specialized texts.

Acknowledgement

This research has been funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number AP19677756 «Unsupervised term extraction: a set of models and datasets for high-tech domains and low-resource languages».

Data availability statement

The code is available in the GitHub repository https://github.com/term-extraction-project/T-Extractor (accessed on 21 January 2025)

Abbreviations

The following abbreviations are used in this manuscript:

T-Extractor	Term Extractor
ACTER	Annotated Corpora for Term Extraction Research
ACL RD-TEC 2.0	Association for Computational Linguistics Reference Dataset for Terminology Extraction and Classification, version 2.0
ACL	Association for Computational Linguistics

HAMLET	Hybrid Adaptable Machine Learning approach to Extract Terminology
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
NE	Named Entity
NMF	Non-Negative Matrix Factorization
UA	Unsupervised Annotator
UA1	Unsupervised Annotator 1
YAKE	Yet Another Keyword Extractor
ChatGPT	Chat Generative Pre-Trained Transforme
Llama	Large Language Model Meta AI
OOV	Out-Of-Vocabulary
POS	Part-of-Speech
MUC-7	Message Understanding Conference 7

References

[1] Hubauer, T.; Lamparter, S.; Haase, P.; Herzig, D. M. Use Cases of the Industrial Knowledge Graph at Siemens. *International Workshop on the Semantic Web*, 2018.

[2] Zhou, D.; Zhou, B.; Zheng, Z.; Soylu, A.; Cheng, G.; Jimenez-Ruiz, E.; Kostylev, E. V.; Kharlamov, E. Ontology Reshaping for Knowledge Graph Construction: Applied on Bosch Welding Case. In *The Semantic Web – ISWC 2022; Springer-Verlag*: Berlin, Heidelberg, 2022; pp. 770–790. https://doi.org/10.1007/978-3-031-19433-7_44.

[3] Dirksen, N.; Takahashi, S. Artificial Intelligence in Japan 2020; *Netherlands Enterprise Agency*, 2020.

[4] Shiroishi, Y.; Uchiyama, K.; Suzuki, N. Better Actions for Society 5.0: Using AI for Evidence-Based Policy Making That Keeps Humans in the Loop. *Computer*, 2019, 52 (11), 73–78. https://doi.org/10.1109/mc.2019.2934592.

[5] Rožanec, J. M.; Novalija, I.; Zajec, P.; Kenda, K.; Tavakoli Ghinani, H.; Suh, S.; Veliou, E.; Papamartzivanos, D.; Giannetsos, T.; Menesidou, S. A.; Alonso, R.; Cauli, N.; Meloni, A.; Recupero, D. R.; Kyriazis, D.; Sofianidis, G.; Theodoropoulos, S.; Fortuna, B.; Mladenić, D.; Soldatos, J. Human-Centric Artificial Intelligence Architecture for Industry 5.0 Applications. *International Journal of Production Research*, 2022, 61 (5), 1–26. https://doi.org/10.1080/00207543.2022.2138611.

[6] Eiden, M. Connecting the Dots with Knowledge Graphs — Opening Statement | *Cutter Consortium*. Cutter.com. https://www.cutter.com/article/connecting-dots-knowledge-graphs.

[7] Drouin, P.; Grabar, N.; Hamon, T.; Kageura, K.; Takeuchi, K. Computational Terminology and Filtering of Terminological Information. *Terminology International Journal of Theoretical and Applied Issues in Specialized Communication*, 2018, 24 (1). https://doi.org/10.1075/term.24.1.

[8] Curcic, D. Number of Academic Papers Published Per Year – *WordsRated*. *Wordsrated*. https://wordsrated.com/number-of-academic-papers-published-per-year/.

[9] Du, R.; An, H.; Wang, K.; Liu, W. A Short Review for Ontology Learning: Stride to Large Language Models Trend. *arXiv (Cornell University)*, 2024. https://doi.org/10.48550/arxiv.2404.14991.

[10] Tran, H.; Martinc, M.; Caporusso, J.; Doucet, A.; Pollak, S. The Recent Advances in Automatic Term Extraction: A Survey. *arXiv (Cornell University)*, 2023. https://doi.org/10.48550/arxiv.2301.06767.

[11] Wang, K.; Gu, S.; Chen, B.; Zhao, Y.; Luo, W.; Zhang, Y. TermMind: Alibaba's WMT21 Machine Translation Using Terminologies Task Submission. In *Proceedings of the Sixth Conference on Machine Translation; Association for Computational Linguistics*, 2021; pp. 851–856.

[12] Huy, H. N. L.; Minh, H. H.; Van, T. N.; Van, H. N. Keyphrase Extraction Model: A New Design and Application on Tourism Information. *Informatica*, 2021, 45 (4), 563-569. https://doi.org/10.31449/inf.v45i4.3493.

[13] Kimura, Y.; Komamizu, T.; Hatano, K. An Automatic Labeling Method for Subword-Phrase Recognition in Effective Text Classification. *Informatica*, 2023, 47 (3), 315–326. https://doi.org/10.31449/inf.v47i3.4742.

[14] Michon, E.; Crego, J.; Senellart, J. Integrating Domain Terminology into Neural Machine Translation. In *Proceedings of the 28th International Conference on Computational Linguistics*; International Committee on Computational Linguistics: Barcelona, Spain, 2020; pp. 3925–3937. https://doi.org/10.18653/v1/2020.coling-main.348.

[15] Condamines, A.; Picton, A. Textual Terminology : Origins, Principles and New Challenges. In *Theoretical Approaches to Terminology; John Benjamins*, 2022; pp. 219–236. https://doi.org/10.1075/tlrp.23.10con.

[16] Jaleniauskienė, E.; Čičelytė, V. Insight into the Latest Computer and Internet Terminology. *Studies About Languages*, 2011, 0 (19). https://doi.org/10.5755/j01.sal.0.19.955.

[17] Zhang, J.; Chen, S.; Hua, J.; Niu, N.; Liu, C. Automatic Terminology Extraction and Ranking for Feature Modeling. In *2022 IEEE 30th International Requirements Engineering Conference (RE)*; Melbourne, Australia, 2022; pp. 51–63. https://doi.org/10.1109/re54965.2022.00012.

[18] Kafando, R.; Decoupes, R.; Valentin, S.; Sautot, L.; Teisseire, M.; Roche, M. ITEXT-BIO: Intelligent Term EXTraction for BIOmedical Analysis. *Health Information Science and Systems*, 2021, 9. https://doi.org/10.1007/s13755-021-00156-6.

[19] Terryn, A. R.; Hoste, V.; Drouin, P.; Lefever, E. TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset. In *Proceedings of the 6th International Workshop on Computational Terminology*; European Language

Resources Association: Marseille, France, 2020; pp. 85–94.

[20] Nugumanova, A.; Akhmed-Zaki, D.; Mansurova, M.; Baiburin, Y.; Maulit, A. NMF-Based Approach to Automatic Term Extraction. *Expert Systems with Applications*, 2022, 199, 117179. https://doi.org/10.1016/j.eswa.2022.117179.

[21] Fusco, F.; Staar, P.; Antognini, D. Unsupervised Term Extraction for Highly Technical Domains. *arXiv (Cornell University)*, 2022. https://doi.org/10.48550/arxiv.2210.13118.

[22] Kalykulova, A.; Kairatuly, B.; Rakhymbek, K.; Kyzyrkanov, A.; Nugumanova, A. Evaluation of IBM's Proposed Term Extraction Approach on the ACTER Corpus. In *IX — International Scientific Conference "Computer Science and Applied Mathematics"*; Almaty, Kazakhstan, 2024; pp. 597–604.

[23] Firoozeh, N.; Nazarenko, A.; Alizon, F.; Daille, B. Keyword Extraction: Issues and Methods. *Natural Language Engineering*, 2019, 26 (3), 259–291. https://doi.org/10.1017/s1351324919000457.

[24] Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; Jatowt, A. YAKE! Keyword Extraction from Single Documents Using Multiple Local Features. *Information Sciences*, 2020, 509, 257–289. https://doi.org/10.1016/j.ins.2019.09.013.

[25] Anjum, O.; Almasri, M.; Xiong, J.; Hwu, W. PhraseScope: An Effective and Unsupervised Framework for Mining High Quality Phrases. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*; 2021; pp. 639–647. https://doi.org/10.1137/1.9781611976700.72.

[26] Di Nunzio, G. M.; Marchesin, S.; Silvello, G. A Systematic Review of Automatic Term Extraction: What Happened in 2022? *Digital Scholarship in the Humanities*, 2023, 38 (Supplement_1), i41–i47. https://doi.org/10.1093/llc/fqad030.

[27] Terryn, A. R.; Hoste, V.; Lefever, E. HAMLET: Hybrid Adaptable Machine Learning Approach to Extract Terminology. *Terminology International Journal of Theoretical and Applied Issues in Specialized Communication*, 2021, 27 (2), 254–293. https://doi.org/10.1075/term.20017.rig.

[28] Hazem, A.; Bouhandi, M.; Boudin, F.; Daille, B. TermEval 2020: TALN-LS2N System for Automatic Term Extraction. In *Proceedings of the 6th International Workshop on Computational Terminology*; European Language Resources Association: Marseille, France, 2020; pp. 95–100.

[29] Banerjee, S.; Chakravarthi, B. R.; McCrae, J. P. Large Language Models for Few-Shot Automatic Term Extraction. In *Natural Language Processing and Information Systems*; Springer, Cham, 2024; Vol. 14762, pp. 137–150. https://doi.org/10.1007/978-3-031-70239-6_10.

[30] Tran, H. T. H.; González-Gallardo, C.-E.; Delaunay, J.; Doucet, A.; Pollak, S. Is Prompting What Term Extraction Needs? In *27th International Conference, TSD 2024*; Springer-Verlag: Berlin, Heidelberg, 2024; Vol. 15048, pp. 17–29. https://doi.org/10.1007/978-3-031-70563-2_2.

[31] QasemiZadeh, B.; Schumann, A.-K. The ACL RD-TEC 2.0: A Language Resource for Evaluating Term Extraction and Entity Recognition Methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*; European Language Resources Association (ELRA): Portorož, Slovenia, 2016; pp. 1862–1868.

[32] Terryn, A. R.; Hoste, V.; Lefever, E. A Gold Standard for Multilingual Automatic Term Extraction from Comparable Corpora: Term Structure and Translation Equivalents. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*; European Language Resources Association (ELRA): Miyazaki, Japan, 2018.

[33] Oliver, A.; Mercè Vàzquez. TermEval 2020: Using TSR Filtering Method to Improve Automatic Term Extraction. In *Proceedings of the 6th International Workshop on Computational Terminology*; European Language Resources Association: Marseille, France, 2020; pp. 106–113.

Multi-Modal Modified U-Net for Text-Image Restoration: A Diffusion-Based Multimodal Information Fusion Approach

Ailong Tang, Ling Wei, Zhiping Ni, Qiuyong Huang*
College of Information Science and Engineering, Liuzhou Institute of Technology, Liuzhou 545000, Guangxi, China
E-mail: qiuhqy@163.com
*Corresponding author

Keywords: multimodal information fusion, multi-modal modified U-Net and refine layer, PSNR, SSIM, text image restoration

Recieved: February 10, 2025

Realistic picture restoration is a crucial task in computer vision, with diffusion-based models widely explored for their generative capabilities. However, image quality remains a challenge due to the uncontrolled nature of diffusion theory and severe image degradation. To address this, we propose a Multi-Modal Modified U-Net (M3UNET) model that integrates textual and visual modalities for enhanced restoration. We leverage a pre-trained multimodal large language model to extract semantic information from low-quality images and employ an image encoder with a custom-built Refine Layer to improve feature acquisition. At the visual level, pixel-level spatial structures are managed for fine-grained restoration. By incorporating control information through multi-level attention mechanisms, our model enables precise and controlled restoration. Experimental results on synthetic and real-world datasets demonstrate that our approach surpasses state-of-the-art techniques in both qualitative and quantitative evaluations, proving the efficacy of multimodal insights in improving image restoration quality.

Povzetek: Predlagan je multimodalni M3UNET model, ki z združevanjem besedilnih in slikovnih informacij ter difuzijskim pristopom bistveno izboljša kakovost obnove degradiranih besedilnih slik, generiranih iz besedil.

1 Introduction

The field of multimodal natural language processing has great promise for enhancing the comprehension and production of material that integrates many modalities, such as text and visuals. Important multimodal natural language processing applications include picture captioning, which generates textual descriptions of images automatically [1]. The data growth on the Internet along with our home PCs is a direct result of the recent improvements in technology and gadgets. Online stores rely on this data, which may be found in a variety of formats (text, images, videos, etc.). These websites feature items that are multimodal, meaning they offer both visuals and textual descriptions. Conventional wisdom holds that a single modality is the sweet spot for classification and data retrieval techniques [2]. Deblurring fuzzy text images without understanding the blur kernel is what blind text picture deblurring is all about. Multiple blind text deblurring image algorithms have shown the efficacy of sparsity-based approaches. Nevertheless, the restoration impact of the blur kernel is impacted by the fact that sparse prior based blur kernel estimates techniques do not take the blur kernel's brightness information into account [3].

The process of image captioning enables computers to decipher visual information and provide textual descriptions of images. After its inception, deep learning's application to the task of interpreting picture data and producing descriptive text quickly became a hot topic in the academic community. However, not all examples that represent conceptual concepts can be found using these procedures. The truth is that most of them don't seem to have any bearing on the matching duties. A small number of significant semantic occurrences define the level of similarity [4]. While human painters have been known to effectively restore items that have suffered extensive damage, inpainting algorithms have so far failed to replicate this feat. Artists often make educated guesses about the severely damaged picture and document them in a written description before attempting to restore it [5]. The need for reliable and flexible image translation methods has increased dramatically in recent computer vision research. But conventional approaches have a hard time adapting material to different settings and capturing semantic subtleties [6]. One method that may be used to create a concise summary using both text and pictures is multimodal abstractive summarizing, or MAS [7]. The modern medical industry is seeing phenomenal growth. Accurate information may be gleaned from the merging of several medical picture modalities, allowing for earlier illness detection and better treatment planning. The goal of picture fusion is to create a single, more comprehensive and informative image than each of the individual input

photos could have been created by separately processing them [8]. One important strategy for social network sentiment analysis in the last few years has been to combine text and visual data. Still, there are limitations to current methods for efficiently integrating multimodal characteristics and collecting complicated cross-modal information [9].

A lot of people are starting to take notice of multimodal big-language models, which means they might be useful for a wide range of vision-language activities in the future. MLLMs include a lot of outside information into their parameters, but keeping these models up-to-date is difficult, since it requires a lot of computing resources and isn't very interpretable. Both LLMs as well as MLLMs have shown success using retrieval enhancement approaches as plugins [10]. In order to facilitate downstream vision tasks, multimodal image fusion seeks to combine data from many imaging modalities into a single, detailed picture. Although models using transformers are computationally costly, they excel in global modeling [11]. In contrast, existing approaches using local CNNs have trouble effectively capturing global characteristics. One classic method for analyzing sentiment that relies on text is multimodal sentiment analysis. Inconsistent cross-modal feature data, inadequate interaction capabilities, with insufficient feature fusion are still issues that the area of multi-modal sentiment evaluation confronts [12]. The goal of image processing operations like restoration and enhancement is to produce a clean, high-quality output from an imperfect input. In regard to single-task circumstances, approaches based on deep learning have shown better performance for a variety of image processing tasks.

However, their generalizability and practical applicability are constrained since they need to train distinct models for various degradations and levels [13]. The goal of image processing operations like restoration and enhancement is to produce a clean, high-quality output from an imperfect input. In regards to single-task circumstances, approaches based on deep learning have shown better performance for a variety of image processing tasks. However, their generalizability and practical applicability are constrained since they need to train distinct models for various degradations and levels [14]. Throughout a patient's journey with the healthcare system, a multitude of clinical data can be found in various formats, including structured, unstructured, or semi-structured information derived from laboratory results, clinical notes, diagnostic code, imaging, audio, and other observational data.

If we can create a representation system that incorporates data from all these different places, we may potentially combine our models on data that has greater predictive value than noise while integrating the limitations we've learned from more accurate information into them [15]. Modern methods for Multimodal Information Extraction are required due to the meteoric increase in the use of social media and other forms of multimodal communication. The semantic and modality gaps that exist between pictures and text pose considerable problems to the direct Image-Text interactions that are the backbone of current techniques [16]. The explosion of new technology and consumer electronics has led directly to the deluge of data stored on computers and the internet. The majority of these data points are compiled from a variety of sources (picture, video, text, etc.). For online stores, this data is equally crucial. The items sold on these websites are multimodal as they include both visuals and textual descriptions. Classification along with data retrieval systems from the past tended to prioritize only one modality over another [17].

1.1 Research gaps

Because current models aren't very versatile, it's usually better to use specialist restoration models that can only do one task at a time. But there are usually a number of degradations happening at once in real-world photos. In a low-quality photograph, for instance, you may see rain, blur, and noise all at once. Degradation phenomena may interact with one another in complicated ways, and various degradations may call for diverse approaches to treatment. The ultimate success of the repair process depends on the order and mix of these techniques. Utilizing expert information and creating all-in-one models have propelled recent field improvements. Below, we provide a full analysis to help you comprehend this subject and our motivations.

- The majority of current picture restoration techniques rely on neural networks that train robust image-level priors from massive amounts of data in order to approximate the missing data. When pictures contain significant information gaps, nevertheless, these efforts still fail.
- There are additional restrictions in the realm of applications for introducing external priors or using reference photos to provide information. As a counterpoint, text input is both more accessible and offers more flexible information.

The specific research contributions are follows:

- Sorting Degradation into Different Categories. The degradation classes of an input picture are detected automatically by M3-UNET, which then calculates the necessary restoration activities.
- A Restoration Sequence by Adaptation. M3-UNET improves the overall efficiency of the picture restoration process by going beyond the limitations of predefined, human-specified model implementation orders and instead deciding the optimal sequence for applying the

restoration models based on dynamic evaluations of the unique properties of each input image.

- Best Practices for Choosing a Model. For each restoration job, M3-UNET dynamically chooses the best model from the pool based on the exact deterioration features in the input picture, guaranteeing optimum performance.
- Automation of Processes. There is no need for human interaction once M3-UNET determines the restoration process and model selection; the whole restoration process is executed autonomously.

1.2 Objectives

- To propose a dual-modal text-guided image restoring model is suggested as a solution to the problems with current restoration algorithms, which include insufficient context-dependent information, poor results in fixing large broken areas, and uncontrollably reconstruction results.
- To connect the significant a representation gap among visual and textual methods is a major challenge in this task.
- To essentially address the feature illustration between visually and textual modalities are another challenge.

The experimental findings show that M3-UNET outperforms human specialists when it comes to sophisticated degradation. In addition, the system's modular architecture makes it easy to add additional tasks and models, which makes it more versatile and scalable for different uses.

1.3 Contributions

Last but not least, we use attention methods to enhance the diffusion model's denoising M3-UNET with pixel control, text embedding, and picture embedding. In order to recover photos while preserving their structural integrity and level of detail, all modules work together. The following is an overview of our contributions:

- We provide a model for realistic picture restoration based on diffusion that takes into account both visual and textual level information.
- To achieve efficient textual control, we use M3-UNET and refine layers. A pixel-level processor with multi-layer supervision allows us to achieve precise control over individual pixels.
- By using multi-layer attention processes, we include the control information into the diffusion model. Our model outperforms the competition on several datasets using a wide range of picture quality criteria.

2 Related work

Image captions that are both informative and accurate is generated using a new multimodal natural language processing approach that was proposed in [18]. By integrating data from image-related text descriptions, their model outperforms conventional unimodal models in capturing contextual signals and producing captions with more nuances. Their multimodal fusion strategy is proven effective as they validate it with the industry standard dataset, Flickr8K, and get state-of-the-art results. In addition, they emphasize how multimodal NLP has the ability to transform their interactions with computers and the way they understand visual information, and they talk about the advantages and disadvantages of that approach to picture captioning. The research cited in [19] uses neutrosophic fuzzy sets to handle uncertainty in information retrieval tasks and classifies multimodal input. Drawing on previous methods of superimposing text over photographs, that endeavor makes use of both image and text data in an effort to categorize the images with neutrosophic classification methods.

Classification tasks make use of feature representations learned by Neutrosophic Convolutional Neural Networks from the generated pictures. For learning representations of the novel fusion approach, they show how to use an NCNN-based pipeline. Conventional convolutional neural networks suffer when trying to classify noisy data because they are susceptible to test-phase noise that isn't yet recognized. Two large-scale multi-modal classification datasets showed promising results when compared to individual sources using their technique. The two popular multi-modal fusion approaches, early fusion along with late fusion, have also been compared to their technique. Present a new approach to blind text picture deblurring using sparse priors and multi-scale fusion in [20]. To further limit the possible solutions space and get excellent clean pictures, they augment the sparse gradient earlier on the hidden clean text image with the sparse earlier on the high-energy wavelet values of the implicit text image. Optimizing the blur kernel with the latent clean image are done in turn using the semi-quadratic splitting approach. They also take into account the restored blur kernel's brightness feature's potential impact. In order to enhance the quality of the blur kernel, they fuse the generated blur kernels in three channels using a multi-scale fusion approach that is based on the Laplacian weight with saliency weight. Their approach successfully restores blur kernels with text pictures, according to the testing findings. Presented in [21] is a deep learning model that utilizes multimodal feature fusion.

Decoding has taken place in long short-term memory; mask recurrent neural networks are used in the coding layer, while the descriptive text is created. Deep learning

makes use of gradient optimization to fine-tune the model's parameters. Dense attention methods may help the decoding layer input the right data preferentially and reduce non-salient data interruption. The goal of using input photographs to train a model is to create captions that, when given the chance, will come close to properly describing the images. The accuracy and proficiency of the model's language acquisition using image description analysis are assessed using several datasets. These tests show that the model correctly describes the input photos every time. This model has learned to describe an input image using words or captions. Classification scores are used to evaluate the model's performance. A 95% improvement in performance is shown by the proposed system while using 100 training epochs and a batch count of 512. Results from experiments conducted on generic picture datasets corroborate the model's ability to understand visual content and produce text. Use of Python frameworks in its implementation and evaluation using performance measures including PSNR, RMSE, SSIM, recall, accuracy, F1-score, and precision are all aspects of that research. By modeling its approach after that of artists' hypothesis, [22] suggests including text description into picture inpainting tasks for the first time; That would give a wealth of information useful for restoring images by fusing multimodal elements.

They present MMFL, a multimodal fusion learning approach to picture inpainting. An image-adaptive word demand component is built to fairly filter the most effective text features, allowing for improved usage of text features they provide a text-image matching penalty and a text-guided attention loss to train the network to focus on items described in text. Their technique outperforms the state-of-the-art in generating fine-grained textures and accurately predicting the semantics of items in the missing areas, according to extensive trials. A groundbreaking method based on multimodal datasets is presented in [23]. They want to improve the picture translation model's ability to understand semantic nuances and improve the accuracy of content adaption by making use of the abundance of data found in multimodal datasets. They want to reinvent image translation technology by fusing information across diverse modalities—images, text, and audio—and bringing unique insights for innovation and growth. By combining deep learning techniques with multimodal data fusion structures, their study aims to fill the gaps in image translation that currently exist. Ensuring robustness along with integrity throughout the analytical process, they painstakingly preprocess and combine data from multiple sources. In a set of carefully planned tests, they compare their method's efficiency to that of more traditional approaches.

Their results demonstrate that their multimodal technique significantly improves translation quality and effectiveness. In addition to establishing a firm groundwork for future research initiatives that study helps push the boundaries of image translating technology forward. They usher in a new age of advancement and innovation in computer vision by illuminating the revolutionary power of multimodal datasets. A successful fusion-based decision-making technique was proposed in [24] to classify social media information into Informative while non-informative categories. The data was analyzed using CrisisMMD and related to seven major natural disasters such as floods, hurricanes, hurricanes, wildfires, etc. The tweets are sorted into many humanitarian categories, including those pertaining to rescue attempts, donations, infrastructure as well as utility damage, impacted persons, and not-humanitarian categories. When compared to baselines based on text tweets, the suggested multi-modal fusion approach achieves 6.98% improvement in the Informative area and 11.2% improvement in the Humanitarian category. When compared to baselines based on picture tweets, the proposed technique achieves a 4.5% improvement in the Informative area and a 6.39 percent improvement in the humanitarian category. When analyzing multimodal data, it is crucial to take into account the interdependencies between the various modalities, as stated in [25]. Their goal is to automate video data analysis by using state-of-the-art deep machine learning along with information fusion techniques that fully consider all interdependencies between and within different modalities. They emphasize the critical significance of human connections in the success of microenterprises with an empirical demonstration that measures the reliability of grassroots merchants in real-time trading on Tik Tok. In order to help with combining information in strategy study that uses multimodal data, they make their data and techniques available.

The importance of nonverbal and vocal communication in reaching strategic goals is emphasized by their research. We show that human contacts are crucial for microenterprises to succeed by analyzing multimodal data (text, photos, and audio) and highlighting the importance of trustworthiness. Their use of explainable AI and data fusion for multimodal information significantly improves the predicted accuracy and theoretical comprehension of trustworthiness assessments. The Dynamic Graph-Text Fusion Network, a multimodal sentiment analysis algorithm, was designed to tackle these issues in [26]. By seeing words as nodes and combining their attributes via their adjacency connections, text features are acquired by using the neighborhood data collection capabilities of Graph Convolutional Networks. The multi-head attention method is also used to concurrently extract extensive semantic data from several subspaces. They use a convolutional attention component to extract features from images. After that, the text and picture characteristics are combined using an attention-based fusion module. The suggested DGFN model is successful,

as shown experimentally on the two datasets, where sentiment accuracy for classification and F1 scores increase significantly. One such retrieval-augmented framework for several MLLMs is proposed in [27] as RA-BLIP, which stands for multimodal adaptive retrieval-augmented bootstrapping language-image pre-training.

Taking into account the fact that the visual modality contains redundant information, they started by using the question to guide the gathering of visual data by interacting with a single set of learnable searches, thereby reducing the amount of irrelevant interference that occurs during both retrieval and production. In addition, they provide a pre-trained multimodal adaptive fusion unit that can integrate visual and verbal modalities into a single semantic space, allowing for query text-to-multimodal retrieval. In addition, they provide an ASKG technique for training the generator's brain to autonomously determine the relevance of recovered information, resulting in exceptional denoising performance. The results show that RA-BLIP outperforms the top retrieval-augmented models and achieves considerable performance on open multimodal question-answering datasets. If you're looking for a way to beat CNNs with vision transformers in computer vision tasks, check out FusionMamba, a new dynamic feature improvement framework introduced in [28]. By using dynamic convolution while channel attention methods, the framework extends the visual state-space paradigm Mamba. That model preserves its outstanding global feature modeling capacity while also reducing redundancy and increasing the expressive capacity of local features. Furthermore, a brand-new module known as the flexible feature fusion component has been created by their team.

It integrates the DFEM module, which improves texture and disparity perception, with the CMFM module, which enhances inter-modal correlation and suppresses redundant information, to build a cross-modal fusion model. The experimental results demonstrate that FusionMamba outperforms its competitors and is highly applicable across a range of multimodal picture fusion challenges and downstream tests. The author suggests using CLIP, an interaction between images and text, to build a cross-modal sentiment framework [29]. To extract main image-text characteristics, the model uses pre-trained ResNet50 and RoBERTa. To improve information transmission across multiple modalities, it uses a multi-head attention system for cross-modal features interaction after contrastive learning using the CLIP model. Then, feature networks are fused using a cross-modal gating module, which allows for the regulation of feature weights while integrating features at various levels. For sentiment recognition, the finished product is sent into a fully linked layer. The MSVA-Single and MSVA-Multiple datasets, which are made publically accessible, are used in comparative investigations. On the aforementioned datasets, their model attained 75.38% accuracy and 73.95% F1-score, according to the experimental findings. That shows that the suggested method outperforms current sentiment analysis methods in terms of generalizability and robustness.

For accurate multimodal sentiment while emotion categorization, the authors of [30] suggested a new deep multi-view attentive architecture. There are three stages to the DMVAN model: learning features, learning attentive interactions, and learning cross-modal fusion. For precise categorization, the feature learning step involves extracting visual features from scene and area views as well as textual data from word, phrase, and document levels of analysis. To improve the interaction between visual and textual information, the image-text interaction learning system is used in the attentive interaction learning phase. That mechanism extracts discriminative and sentimental visual features and uses textual information to train image features. To further take use of the complementing qualities of several modalities, a cross-modal fusion instructional component is designed to merge distinct features into a holistic framework. Next, a multi-head attention technique is used to gather and combine sufficient information from the intermediate characteristics to help build a strong joint representation. In other related fields, image processing methods such as photogrammetry, medical imaging, and computer vision have their characteristics and innovations. Some cutting-edge image processing methods will help researchers achieve new breakthroughs. Table 1 summarizes image registration algorithms in other fields.

Table 1: Analysis of image registration algorithms in other fields

Reference	**Proposed method**	Limitations
Authors [18]	Multimodal NLP	Difficult to fuse infrared-visible and multi-focus image fusion.
Authors [19]	Neutrosophic Fuzzy sets Neutrosophic Convolutional Neural Networks	Difficult to propose more rapid and active methods of medical image enhancement and fusion

Authors [20]	High-Energy Wavelet Using The Semi-Quadratic Splitting Approach	Low image quality, performance was not consistent so low efficiency
Authors [21]	Recurrent Neural Networks	Computational complexity was high
Authors [22]	MMFL	Difficult to fuse the images
Authors [23]	combining deep learning techniques	Fused image quality was poor
Authors [24]	fusion-based decision-making technique	Required more computational time to perform the task
Authors [25]	deep machine learning along with information fusion	Implementation was complex
Authors [26]	Dynamic Graph-Text Fusion Network	Fused image quality was poor
Authors [27]	RA-BLIP	Failed to execute in real-time applications
Authors [28]	CNNs with vision transformers	Required more computational time to perform the task
Authors [29]	ResNet50 and RoBERTa	Time consumption was more
Authors [30]	DMVAN model	Difficult to fuse the images

3 Proposed Work

3.1 System Method

Fig 1 shows the proposed work architecture for image restoration.

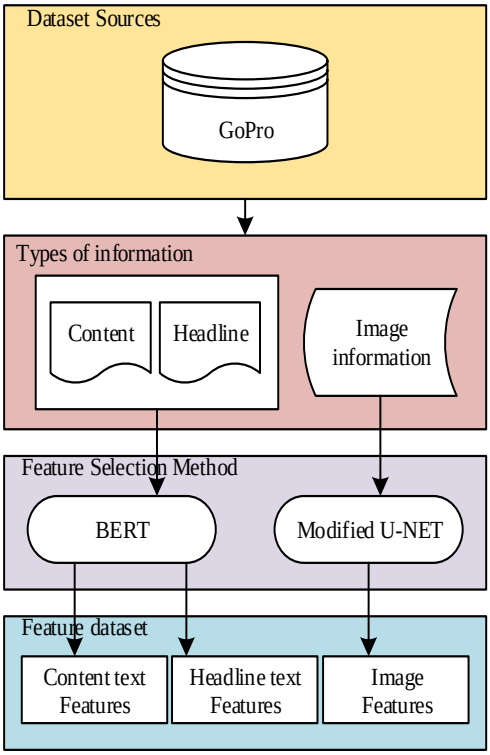


Figure 1: Flow diagram for Modified U-Net Model

Text processing:

- We search for keywords, organizations, and semantic connections in the description language and pull them out.

Image feature extraction:

- The degraded picture is processed using methods such as CNNs to extract features.

Feature fusion

- A selected fusion technique merges the text-derived characteristics with the picture features, making use of attention processes to zero in on pertinent data.

Image reconstruction

- A restored picture is often constructed using the fused characteristics, typically by use of a generative model.

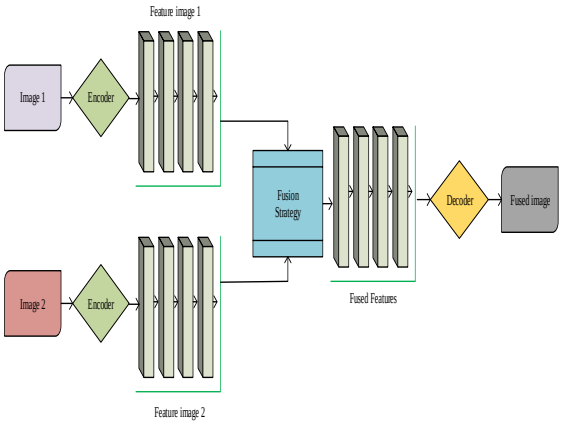


Figure 2: Flow diagram for text image restoration

The BERT and M3-UNET models handle the features of content text, headline text, and images, respectively, during the feature extraction phase. Textual data is processed using the BERT model. It excels in text comprehension because to its bidirectional processing capability, which allows it to pick up on nuanced semantic variations seen in news items' contexts. Using its deep residual network design, the M3-UNET model efficiently finds and interprets visual information pertaining to news when applied to picture data. Very helpful for complicated feature extraction from pictures, its structure enables it to participate in deep learning without training challenges. The flow diagram shows three fusion strategies—early, joint, and late fusion—that attempt to combine text and picture characteristics after feature processing. The last step in picture restoration is to merge these integrated characteristics.

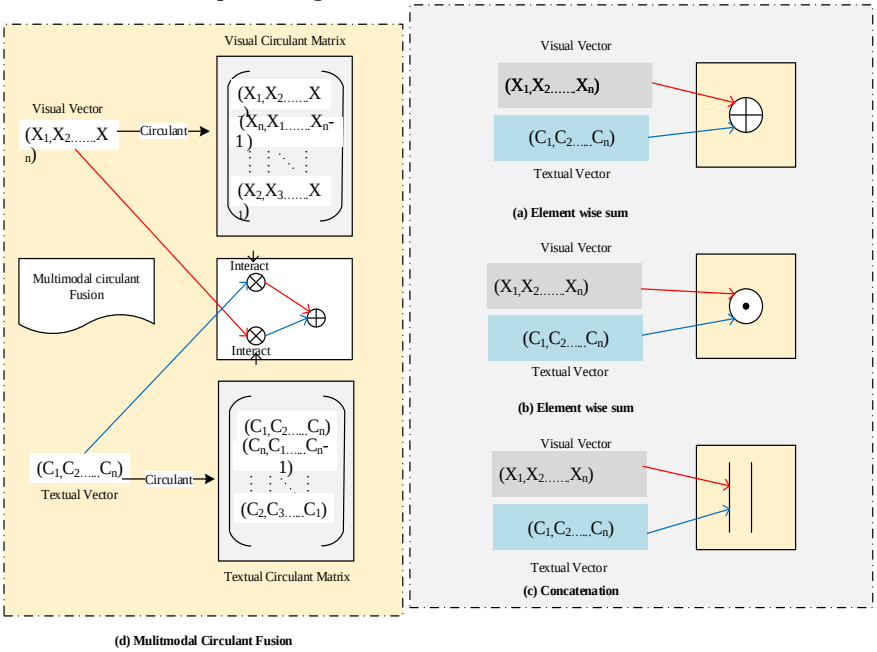


Figure 3: Modified U-Net Model architecture

3.2 Data preprocessing

3.2.1 Data cleaning

This research begins by extracting the information. Then, it uses English tokenization to clean the data by eliminating noise, punctuation marks, graphics, and hyperlinks. Next, stop word removal is applied to the data. Lastly, a spell check is run to fix any misspelled English keywords as well as replace them with the right ones. This ensures that the data feature extraction process is done accurately in terms of interpretation and judgment.

3.3 Feature extraction

Word2vec and Doc2vec

Two text feature extraction approaches are used in this study: Word2vec and Doc2vec. Word2vec is used for word embeddings and word vectors, while Paragraph2vec is used for sentence embeddings and paragraph vectors. Word2Vec captures the semantic associations between words in context by transforming them into vector space. By mapping each phrase to a vector space where similarly-themed sentences are grouped together, Doc2vec depicts sentence characteristics in a manner analogous to how vectors represent individual words. As opposed to Word2Vec's word-level analysis, Doc2Vec allows the research to examine the text at the document level, which incorporates a larger context. By assessing the document's presentation or discussion of subjects, this approach enables the machine to extract the main points of whole articles, which helps in detecting false news. With every piece of text, the popular natural language processing (NLP) package Gensim extracts 50 Word2vec features and 50 Doc2vec features.

BERT feature vectors

This research yields 768-dimensional BERT Base feature vectors. There are two versions of BERT; the one utilized for this study is the Base version, which has 768 hidden states, and the other is the large version, that has 1024 hidden states. The hidden state's size is 768, which means that every token is converted into a vector with 768 dimensions. Token count, hidden state size (768), along with layer count (12) make up the three dimensions of a tensor that is the product of all layer outputs.

Image feature vectors

Computer vision differs from human vision in how it processes visual information. Images are seen by computers as a combination of numerical values, in contrast to humans who can directly perceive form and color. What this means is that images may be thought of as structured collections of numerical data. The picture model used in this research makes use of M3-UNET for feature extraction. M3-UNET has been a popular model for a while now because to its modest size and great performance. Using the 1000-dimensional vector extracted from M3-UNET's last convolutional layer as picture features, the study.

Fusion methods

In order to make the model more accurate and diverse, this research merges text and visual characteristics. The three techniques of fusion are joint, late, and early. Here we will present these three techniques.

3.4 M3-U-Net model

In order to make the approach more flexible for real blurry photos, this research suggests a multi-scale altered U-Net image deblurring system that uses dilated convolution to return a clear image from a dynamically blurred one. By making use of dilated convolution, multi-scale architecture is able to learn the qualities of pictures at various scales, take use of the receptive field's benefits, and extract more information from attributes with less computational cost.

Incorrect camera settings or faulty hardware are common causes of digital photography's flawed, low-quality results. Dynamic blurring, which happens when you take a still shot of an item in motion or when the camera shakes, is the most common kind of visual flaw. on clarify, the image deblurring approach states that a blurred picture is just a convolutional operation applied on a crisp image using a blur kernel.

$$B = K \otimes I + N \quad (1)$$

⊎ signifies the convolution operation, N stands for noise, l represents the initial sharp image, K stands for the blur kernel, and B represents the blurred image. The blur kernel, which explains how pixels diffuse along a moving trajectory, is another name for the point-spread function. To restore clarity to hazy photos is the main goal of picture deblurring techniques. Blurred pictures and blurred kernels are used in image deconvolution to recover crisp images. There are two main types of deconvolution algorithms now in use: blind and non-blind. While the latter's blur kernel is known, the former's is unknown. Typically, the results are pictures with a lot of blurs. Nevertheless, we don't sure what the blurred kernels yet crisp pictures are. In order to use the conventional approach to restore a blurred picture to a clear one, it is crucial to precisely estimate the blur kernel. Nevertheless, we still don't know whether the estimated blur kernel deconvolution of a blurry picture yields a proper solution for the approximation crisp image. To fix blurry pictures after blind deconvolution—an ill-posed problem—more data is needed.

One hotspot for study is the topic of picture deblurring methods in dynamic settings. In the realm of picture deblurring techniques, deep learning algorithms has recently attracted a lot of attention. Unfortunately, these approaches often have poor deblurring results and insufficient receptive fields due to a lack of natural

connections across different hierarchical levels. A more flexible method is used in U-Net, which successfully integrates characteristics of varying degrees. This solution keeps the total amount of parameters within an acceptable range while also drastically reducing their complexity. The current work presented an enhanced U-Net model to increase the picture deblurring effect, based on these benefits.

One common use of encoder-decoder networks in computer vision is picture deblurring, where they have shown to be effective. The network's architecture is a convolutional neural network (CNN) with symmetrical encoder and decoder components. The input picture is first downsampled by the encoder into a smaller feature map having more channels and rich details, and then the decoder takes that smaller map and upsamples it into a bigger one having fewer channels and deeper details. The feature fusion-primarily skip connection, or residual connection, was introduced by Ronneberger et al. between the decoder and encoder networks. The amount of feature data lost during downsampling increases dramatically as the network depth increases. The feature map that is obtained by encoder downsampling bypasses the multilayer system and links directly to the decoder correspondence to guarantee that the last feature map has enough precise information. In order to get feature maps with better information, the corresponding decoder upsampling uses a combination of shallow and deep features. Figure 1 shows a network topology that is known as a U-Net because of its resemblance to a U-shape. Image convolution computation (blue arrow), skip connection (gray arrow), max pooling (red arrow), and deconvolution (green arrow) are shown, respectively. The max pooling approach is used by the U-Net downsampling technique. This method splits the picture into many 2 × 2 rectangular sections, calculates the highest value for every area, and then uses this information to decrease the quantity of image data while keeping crucial details.

Loss function

We studied multi-scale picture deblurring networks, which included calculating the weight as well as the loss between each scale's output and target images. This article primarily discusses Ye et al.'s lifting-scale iteration architecture, which differs from existing approaches. In this design, each scale iteration is considered an independent deblurring subtask; this means that it continues to have an impact even after the training phase stops. Consequently, the mean absolute error (MAE) is chosen as the loss function, and the sole metric computed is the difference between the target picture and the final deblurred image. L_1 loss is another name for the MAE.

$$\text{Loss} = \frac{L_i - G_i}{N_i} \quad (2)$$

where L_i represents the crisp picture, G_i stands for the deblurred image, and N_i signifies the quantity of components in L_i that need to be normalized at the i-th scale.

4 Results & discussion

4.1 Experimental environment

(a). Hardware environment

Powered through six Nvidia GeForce RTX 3090 24-GB GPUs, the experimental system made use of a powerful computing server with an AMD Ryzen Threadripper 3990Y @ 3.70 GHz CPU and 1TB of RAM. This powerful hardware architecture is perfect for deep learning activities like model development and inference because to its large storage capacity and outstanding processing capability. Faster experimentation and convergence are guaranteed by the state-of-the-art hardware, which drastically shortens the training time.

(b). Software environment

The main programming language for this research was Python 3.8, with PyTorch being used for deep learning applications. A flexible and iterative development approach was made possible by Python's flexibility. At the same time, PyTorch supplied the necessary tools for creating and teaching neural networks. We enhanced experimental results by developing, optimizing, and training our models quickly using PyTorch's robust computing capabilities and its automated differentiation function.

The experimental training of the model was accelerated by training it on a GPU, which excels at computationally demanding image processing jobs. Table 2details the experimental setting and setup that were used in this investigation.

Table 2: Experimental settings

Configuration	**Experimental environment**
Windows10	Operating system
Python	Programing language
PyTorch	Deep learning framework
GTX 2080	GPU
16.0 GB	Memory size

(c). Dataset

There are 32,214 distinct scenarios included in the GoPro collection, including both clear and fuzzy images. Two thousand three picture pairings made up the training dataset, whereas eleven thousand eleven picture pairs comprised the test dataset. We enhanced the data from the GoPro along with Real Blur training sets using data improvement methods to make the model more generalizable. Two of the actions were adding Gaussian noise and randomly rotating the data. The data augmentation specifically included random rotations of 90, 180, while 270 degrees, as well as horizontal (left to

right) while vertical (upside down) flips. Also included was some Gaussian noise, which had a mean of 0 and a standard deviation of 0.0001. This led to an increase of 1,412 picture pairings in the GoPro training dataset and 1,5032 in the Real Blur training dataset, both achieved by means of these augmentation approaches.

Divide the Data into Three Parts: Training, Validation, and Testing Set of data: In order to train our algorithm, we use the training set to experiment with different hyperparameters. Then, we test it on the validation set and choose the hyperparameters that provide the best results. Selecting hyperparameters in this manner is recommended. The "fully convolutional network" is the ancestor of the U-Net design. Substituting upsampling operators for pooling operations in subsequent levels of a conventional contracting network is the key concept. Therefore, the output's resolution is enhanced by these layers. For the vast majority of customers, this is the reality. Only a small number of cloud providers have internal, on-demand access to massive, homogenous collections of lightning-fast hardware. In the first scenario, optimising hyperparameter searches for hardware efficiency requires tweaks to the search method. It is still possible to decide on a degree of parallelism to use throughout the search even in the second situation, for sets of homogenous systems. Machine learning algorithms of all stripes use hyperparameters, and the amount of these parameters, which may span both numerical and categorical domains, varies from algorithm to algorithm. For this optimization issue, many hyperparameter search techniques have been suggested, from the most basic, like random search, to the most complex, including methods like Probabilistic optimization, gradient-based learning, and bandit-based searches. Both of these methods simplify things by assuming certain things, but they do solve the issue of selecting the next parametrization to test: 1) all hyperparameters are treated similarly throughout the search process, and 2) all search spaces for hyperparameters are equally complicated. Both of these hypotheses are usually not true in real life. Because their suitability to the learning goal is not known with confidence, models with a bigger priority are selected in the search, similar to complexity. The model's performance under alternative parameterizations could be better understood by doing a more comprehensive exploration of the hyperparameter spaces, as this lack of assurance suggests. Since it is impossible to determine how well a model performs throughout the whole hyperparameter domain, it is reasonable to keep searching for low-priority models, but to reduce the total number of searches overall.

Images from the training datasets were arbitrarily cropped to a resolution of 256 × 256 pixels to avoid model overfitting. With an initial learning rate of 1e-4 and a half-life adjustment every 1000 rounds, the training time was determined to 3000 rounds. Additionally, 10 was the batch size. Adam, with parameters A1 = 0.9 as well as A2 = 0.999, was chosen as the network optimization algorithm.

4.2 Experimental datasets

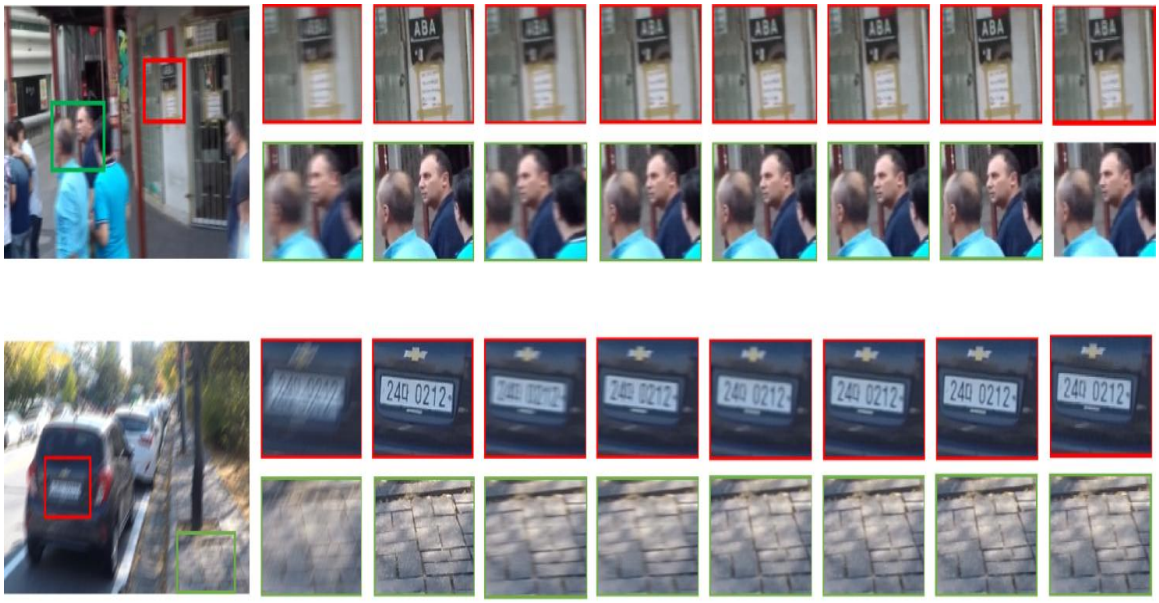
We competed tests on two separate datasets to ensure our model was completely validated. We may train the model in several ways using the data provided by these datasets, and we can assess the performance of the combined approach of visual recognition with language processing for graphic recognition using these datasets.

Table 3: Comparison of experimental results of dilation rate strategy

Strategies	D = 3	D = 4	D = 5	D = [3,4,5]
PSNR	**30.33**	29.77	29.88	29.99
SSIM	**0.877**	0.822	0.826	0.723
LPIPS	**0.9**	0.91	0.92	0.914
FID	**30**	32	31.2	31.6

PSNR: greatest signal-to-noise ratio; SSIM: structure-similarity index measurement; LPIPS (Learned Perceptual Image Patch Similarity); FID (Fréchet Inception Distance)

The data is superior to other data sets, as shown by the bold language.



(a), (b), (c), (d), (e), (f), (g), (h)

Figure 5: Comparison of visual effects on the dataset (a), SVM (b), LSTM (c), CNN (d), CLIP (e), VISTANET (f), MMFL (g), U-Net (h). Proposed Model

4.3 Discussion

The synthesizing advice will continually represent the relevant degradation pattern, conditional on the deteriorated text word embedding. Here, we provide a fresh viewpoint: bolstering picture repair with restoration in textual space. To further improve performance, we merge the benefits of text-based prior (which learns degradations at the textual level) with image-based prior (which provides clean guidance).In this article, we provide an innovative way of looking at image restoration. Since content and degradation are associated in images but not in texts, we propose a plug-and-play method that takes degraded images and converts them into text, then eliminates textual deterioration details to get restored text. As opposed to doing reconstruction on the image level, we suggest restoring on the textual level and then using the restored text to aid image restoration. On GoPro test data, our method improves PSNR by 0.35 dB. Figure 5 shows the visual outcomes of our method's restoration efforts, which demonstrate its success by restoring pictures with crisper borders and features. Table 3 shows that when compared to current approaches, our method improves single-image defocus deblurring by +0.35 dB and dual-pixel defocus deblurring by +0.45 dB in terms of PSNR. Figure 5 illustrates that our method's forecast has a more organized structure.

5 Conclusion

In this study, we successfully employ text information to help with picture restoration since text input is more easily accessible and gives information with more flexibility. We apply a M3U-NET-based model and create a simple and effective framework in order to develop a text-based picture restoration approach. This framework enables the user to enter text and get the appropriate image restoration results. The framework uses M3U-NET text-image feature compatibility to improve the combination of picture and text features. Our system is capable of performing a variety of picture restoration tasks, such as image in painting, image super-resolution, and image colorization.

References

[1] Jha, S., Mayer, E., & Barahona, M. (2022, December). Improving information fusion on multimodal clinical data in classification settings. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)* (pp. 154-159). https://doi.org/10.18653/v1/2022.louhi-1.18

[2] Luo, W., Xia, Y., Tianshu, S., & Li, S. (2024, October). Shapley Value-based Contrastive Alignment for Multimodal Information Extraction. In *Proceedings of the 32nd ACM International Conference on Multimedia* (pp. 5270-5279). https://doi.org/10.1145/3664647.3681367

[3] Wajid, M. A., Zafar, A., & Wajid, M. S. (2024). A deep learning approach for image and text classification using neutrosophy. *International Journal of Information Technology*, *16*(2), 853-859. https://doi.org/10.1007/s41870-023-01529-8

[4] Al-Tameemi, I. S., Feizi-Derakhshi, M. R., Pashazadeh, S., & Asadpour, M. (2023). Multi-model fusion framework using deep learning for

visual-textual sentiment classification. *Computers, Materials & Continua*, *76*(2), 2145-2177. https://doi.org/10.32604/cmc.2023.040997

[5] Liu, J., Ma, X., Wang, L., & Pei, L. (2024). How Can Generative Artificial Intelligence Techniques Facilitate Intelligent Research into Ancient Books?. *ACM Journal on Computing and Cultural Heritage*, *17*(4), 1-20. https://doi.org/10.1145/3690391

[6] Kumar, P., Malik, S., Raman, B., & Li, X. (2022). VISTANet: VIsual Spoken Textual Additive Net for Interpretable Multimodal Emotion Recognition. *arXiv preprint arXiv:2208.11450*. https://doi.org/10.48550/arXiv.2208.11450

[7] Zong, D., Ding, C., Li, B., Zhou, D., Li, J., Zheng, K., & Zhou, Q. (2023, October). Building robust multimodal sentiment recognition via a simple yet effective multimodal transformer. In *Proceedings of the 31st ACM International Conference on Multimedia* (pp. 9596-9600). https://doi.org/10.1145/3581783.3612872

[8] Ouafa, C., & Tayeb, L. M. (2022). Facial Expression Recognition Using Convolution Neural Network Fusion and Texture Descriptors Representation. *International Journal of Computational Intelligence and Applications*, *21*(01), 2250002. https://doi.org/10.1142/s146902682250002x

[9] Chandrasekaran, G., Nguyen, T. N., & Hemanth D, J. (2021). Multimodal sentimental analysis for social media applications: A comprehensive review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *11*(5), e1415. https://doi.org/10.1002/widm.1415

[10] Kaur, P., Malhi, A., & Pannu, H. (2024). Annotate and retrieve in vivo images using hybrid self-organizing map. *The Visual Computer*, *40*(8), 5619-5638. https://doi.org/10.1007/s00371-023-03126-z

[11] Picha, S. G., Chanti, D. A., & Caplier, A. (2024). Semantic textual similarity assessment in chest x-ray reports using a domain-specific cosine-based metric. *arXiv preprint arXiv:2402.11908*. https://doi.org/10.48550/arXiv.2402.11908

[12] Akhmerov, A. K., Vasilev, A. S., & Vasileva, A. V. (2019, June). Research of spatial alignment techniques for multimodal image fusion. In *Multimodal Sensing: Technologies and Applications* (Vol. 11059, pp. 309-317). SPIE. https://doi.org/10.1117/12.2526030

[13] Leonardo, R., Hu, A., Uzair, M., Lu, Q., Fu, I., Nishiyama, K., ... & Ravichandran, D. (2019, December). Fusing visual and textual information to determine content safety. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)* (pp. 2026-2031). IEEE. DOI: 10.1109/ICMLA.2019.00324

[14] Meng, L., Tan, A. H., Wunsch II, D. C., Meng, L., Tan, A. H., & Wunsch II, D. C. (2019). Socially-Enriched Multimedia Data Co-clustering. *Adaptive Resonance Theory in Social Media Data Clustering: Roles, Methodologies, and Applications*, 111-135. https://doi.org/10.1007/978-3-030-02985-2_5

[15] Li, Z., Zhang, D., Du, Y., & Zhang, X. (2024). A Study on the Application of Multimodal Fusion Technology in the Translation of the Historical Literature of Geng Lu Bu. *Applied Mathematics and Nonlinear Sciences.* https://doi.org/10.2478/amns-2024-3626

[16] Song, Y., Lin, N., Li, L., & Jiang, S. (2024, May). A Vision Enhanced Framework for Indonesian Multimodal Abstractive Text-Image Summarization. In *2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD)* (pp. 61-66). IEEE. https://doi.org/10.1109/cscwd61410.2024.10580245

[17] Rizmolp, G. (2020). Detection of Abnormalities Using Multimodal Image Fusion and Text Encryption. | ISSN: 2320-2882

[18] Tiwari, M., Khare, P., Saha, I., & Mali, M. (2024). Multimodal NLP for image captioning : Fusing text and image modalities for accurate and informative descriptions. *Journal of Information and Optimization Scienc*es. https://doi.org/10.47974/jios-1626

[19] Wajid, M.A., Zafar, A., Terashima-Marín, H., & Wajid, M.S. (2023). Neutrosophic-CNN-based image and text fusion for multimodal classification. *J. Intell. Fuzzy Syst*., 45, 1039-1055. https://doi.org/10.3233/JIFS-223752

[20] Li, Z., Yang, M., Cheng, L., & Jia, X. (2023). Blind Text Image Deblurring Algorithm Based on Multi-Scale Fusion and Sparse Priors. *IEEE Access*, 11, 16042-16055. DOI: 10.1109/ACCESS.2023.3245150

[21] Thangavel, K., Palanisamy, N., Muthusamy, S., Mishra, O.P., Sundararajan, S.C., Panchal, H.D., Loganathan, A.K., & Ramamoorthi, P. (2023). A novel method for image captioning using multimodal feature fusion employing mask RNN and LSTM models. *Soft Computing*, 27, 14205-14218. https://doi.org/10.1007/s00500-023-08448-7

[22] Lin, Q., Yan, B., Li, J., & Tan, W. (2020). MMFL: Multimodal Fusion Learning for Text-Guided Image Inpainting. *Proceedings of the 28th ACM International Conference on Multimedia.* https://doi.org/10.1145/3394171.3413982

[23] Zhou, L. (2024). Research on Image Translation Problems Based on Multimodal Data Set Fusion. *International Journal of Computer Science and Information*

Technology. https://doi.org/10.62051/ijcsit.v3n3.03

[24] Kota, S.M., Haridasan, S., Rattani, A., Bowen, A., Rimmington, G.M., & Dutta, A. (2022). Multimodal Combination of Text and Image Tweets for Disaster Response Assessment. *D2R2*. https://soar.wichita.edu/handle/10057/25302

[25] Luo, X., Jia, N., Ouyang, E., & Fang, Z. (2024). Introducing machine-learning-based data fusion methods for analyzing multimodal data: An application of measuring trustworthiness of microenterprises. *Strategic Management Journal*. https://doi.org/10.1002/smj.3597

[26] Li, J., Bai, X., & Han, Z. (2024). DGFN Multimodal Emotion Analysis Model Based on Dynamic Graph Fusion Network. *International Journal of Decision Support System Technology*. https://doi.org/10.4018/ijdsst.352417

[27] Ding, M., Ma, Y., Qin, P., Wu, J., Li, Y., & Nie, L. (2024). *RA-BLIP: Multimodal Adaptive Retrieval-Augmented Bootstrapping Language-Image Pre-training*. ArXiv, abs/2410.14154. https://doi.org/10.48550/arXiv.2410.14154

[28] Xie, X., Cui, Y., Ieong, C., Tan, T., Zhang, X., Zheng, X., & Yu, Z. (2024). *FusionMamba: Dynamic Feature Enhancement for Multimodal Image Fusion with Mamba*. ArXiv, abs/2404.09498. https://doi.org/10.1007/s44267-024-00072-9

[29] Lu, X., Ni, Y., & Ding, Z. (2024). Cross-Modal Sentiment Analysis Based on CLIP Image-Text Attention Interaction. *International Journal of Advanced Computer Science and Applications*. https://doi.org/10.14569/ijacsa.2024.0150290

[30] Al-Tameemi, I.K., Feizi-Derakhshi, M., Pashazadeh, S., & Asadpour, M. (2023). Interpretable Multimodal Sentiment Classification Using Deep Multi-View Attentive Network of Image and Text Data. *IEEE Access*, 11, 91060-91081. https://doi.org/10.1109/access.2023.3307716

Enhancing OSN Security: Detecting Email Hijacking and DNS Spoofing Using Energy Consumption and Opcode Sequence Analysis

[1*]Romil Rawat, [2]Kamal Borana, [3]Shweta Gupta, [3]Mandakini Ingle, [5]Ashish Dibouliya, [6]Purvee Bhardwaj, [7]Anjali Rawat
[1*]LabGeoInf – Research LABoratory in GEOmatics and INFormation systems, Rome, Italy
[2]Department of CSE, SVIIT, SVVV, Indore - India
[3]Computer Science and Engineering Department, Medicaps University, Indore (M.P.), India
[5]Data Architecture (Webster Bank- USA)
[6]Rabindranath Tagore University, Bhopal, (MP) - India
[7]Department of Computer and Communication Technology, University of Extremadura, Spain
E-mail: rawat.romil@gmail.com, kamalborana@svvv.edu.in , shweta.gupta@medicaps.ac.in, tayademandakini@gmail.com, ashish.dibouliya@gmail.com, purveebhardwaj@gmail.com, rawatanjali457@gmail.com
[*]Corresponding author

Keywords: OSN security, email hijacking detection, DNS Spoofing prevention, energy consumption footprint, opcode sequence mining, automated threat detection

Received: August 21, 2024

The rapid increase in automation within Online Social Networks (OSNs) has led to a surge in cyber threats, notably Email Hijacking and DNS (Domain Name System) Spoofing, which leverage malicious scripts to manipulate traffic, steal credentials, and evade detection. Traditional security mechanisms fail to effectively identify such automation-based attacks, necessitating an advanced detection framework. Objective & Purpose-This study introduces the Automated Social Network Attack Detection Model (ASNADM), which combines Energy Consumption Footprint (EComp-FP) Analysis and Automated Software Opcode Sequence Analysis (ASOSA-OSM- opcode sequence mining) for high-precision OSN security. EComp-FP detects deviations in power consumption linked to malicious automation tools, while ASOSA-OSM analyzes opcode sequences to differentiate between benign and attack behaviors. The Self-Adaptive Fuzzy Pattern Matching Clustering (SAFPMC) Algorithm enhances classification accuracy, reducing false alarms and improving real-time threat detection. Methodology and Dataset-The model was rigorously evaluated using the SPEMC-15K-E (Spam Email Classification dataset in English) dataset (15,000 samples: 7,500 benign, 7,500 malicious). EComp-FP achieved 99.87% accuracy with a 1.4W power deviation, while ASOSA-OSM attained 99.81% accuracy, detecting automation tools with an Opcode Frequency Variance (OFV) of 8.7 in malicious samples versus 3.5 in benign ones. The hybrid EComp (Energy Consumption) + OSA (Opcode Sequence Analysis) model outperformed both standalone methods, achieving 99.93% accuracy, 99.91% F1-score, a false positive rate of just 0.07%, and a false negative rate of 0.05%. Among classifiers, the Self-Adaptive Soft Fuzzy C-Means (SSFCM) Hybrid model achieved the highest performance, with 99.93% accuracy, 99.85% precision, 99.9% recall, and the lowest misclassification rate of 0.05%, surpassing Decision Tree (DT), K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM). Result - Optimization techniques significantly improved real-time detection efficiency. The SAFPMC algorithm reduced detection latency by 35%, while parallel processing lowered computational overhead by 31%. Feature selection improved classification speed by 27%, and federated learning reduced processing load by 25%, enabling scalable, real-time OSN threat monitoring. This study presents an advanced hybrid detection framework for OSN security, combining energy consumption profiling and opcode sequence analysis to detect email hijacking and DNS spoofing attacks. The model achieves a 99.92% detection precision, a 99.89% real-time accuracy, and reduces computational overhead by 31%, making it a robust and efficient solution for securing online social networks. These findings confirm that combining energy profiling and opcode sequence analysis is highly effective in detecting automation-based OSN threats. Future work will focus on integrating deep learning (DL) for anomaly detection, AI (artificial intelligence)-driven botnet defense, and enhancing large-scale OSN threat mitigation strategies.

Povzetek: V članku je opisan hibridni model za zaznavanje napadov v družbenih omrežjih, ki z analizo porabe energije in zaporedij ukazov doseže veliko točnost pri odkrivanju e-poštnih in DNS napadov.

1 Introduction

The rise of automation in OSN [1] has increased the risk of cyber threats [2], particularly Email Hijacking [3] and DNS Spoofing [4]. These attacks exploit malicious automation tools to manipulate network traffic, steal credentials, and intercept communications. Recent incidents highlight the growing use of AI-driven phishing [5] campaigns and botnet attacks, which evade traditional security systems. Given the increasing role of OSNs in financial transactions, enterprise communications, and authentication mechanisms, ensuring robust security measures is essential.

Future work will focus on expanding detection capabilities to counter AI-driven botnets [6][7] and Advanced Persistent Threats, integrating DL models for enhanced anomaly detection, and optimizing real-time threat monitoring for large-scale OSN infrastructures.

The rapid evolution of cyber threats targeting OSN [8] has introduced sophisticated attack variants, including AI-powered botnets, adversarial machine learning (ML)-based evasion techniques [9], and large-scale automated credential stuffing. Traditional security mechanisms struggle to detect these advanced threats, as attackers increasingly leverage self-mutating automation tools and polymorphic malware. A notable example is the large-scale OSN credential breach involving over 500 million compromised accounts, where automated scripts were used to hijack user sessions and execute phishing campaigns. Similarly, deepfake-powered social engineering [10] attacks have been weaponized to impersonate high-profile individuals, manipulate public opinion, and spread misinformation at an alarming scale and detection mechanisms that can accurately differentiate between benign and malicious automation activities in OSNs.

The integration of AI in cyberattacks [11] has led to the rise of adaptive and intelligent automation tools capable of evading conventional security measures. Attackers now employ adversarial ML to modify malware [12] behavior dynamically, making detection more challenging. DNS tunnelling [13][14] has also been exploited in OSN automation attacks, allowing attackers to exfiltrate sensitive data while bypassing conventional monitoring systems. A significant incident involved a DNS spoofing-based OSN attack where cybercriminals manipulated domain resolution processes, redirecting users to counterfeit versions of trusted platforms and harvesting their credentials. To counter these evolving threats, modern cybersecurity frameworks now incorporate hybrid DL models such as Long Short-Term Memory (LSTM)[15] networks for anomaly detection and Transformer-based architectures for opcode sequence classification. The combination of energy consumption analysis and opcode sequence analysis offers
a robust security framework capable of detecting complex automation-driven OSN threats.

In response to the growing sophistication of automation attacks, researchers have focused on developing multi-layered detection frameworks that integrate behavioral analysis with computational intelligence. EComp [16] analysis has emerged as a promising technique for identifying automation threats by monitoring the anomalous energy footprints of automated software running on Socially Shared Networked Devices (SSNDs) [17]. By analyzing power usage patterns under normal and attack conditions, security systems can flag unusual spikes indicative of malicious activities such as session hijacking and DNS spoofing [18]. In parallel, OSA plays a crucial role in detecting automation tools [19] by extracting and analyzing the opcode sequences of suspicious binaries. ASOSA enables deep inspection of execution patterns, differentiating between benign and malicious automation behaviors. The combination of these methods significantly enhances detection accuracy in OSN environments.

The SPEMC-15K-E dataset [1][2], containing extensive real-world samples of automated and non-automated software, has been utilized to train and evaluate modern OSN attack detection models. Recent research shows that classifiers [20][21] such as DT, KNN, RF, SVM, and SSFCM achieve varying levels of detection accuracy. Among these, SSFCM has demonstrated superior performance, achieving 99.79% accuracy in detecting automation attacks via OSA and 99.87% accuracy using EComp analysis. Additionally, hybrid DL-techniques, including Convolutional Neural Networks (CNNs) [22] combined with Recurrent Neural Networks (RNNs) [23], have further improved anomaly classification in opcode-based threat detection. The integration of these methodologies ensures a comprehensive approach to combating OSN automation threats.

Recent cybersecurity advancements have led to the deployment of real-time threat detection systems that leverage energy profiling and opcode sequence [24][25] analysis in automated attack prevention. Edge-based AI models are being integrated into SSNDs, allowing for in-device threat detection without relying on cloud-based solutions [26]. This reduces latency and enhances privacy by processing security events locally. Furthermore, federated learning approaches have been adopted to continuously update detection models across distributed devices while preserving user data confidentiality. These improvements in automated threat detection provide a proactive approach to mitigating large-scale automation attacks in OSNs while minimizing false positives and computational overhead.

Future research aims to enhance the scalability and adaptability of OSN security frameworks by incorporating emerging technologies such as quantum ML and explainable AI [27]. Quantum computing [28][29] offers the potential to analyze opcode sequences at an unprecedented scale, drastically improving detection speeds and efficiency. Explainable AI models are also being explored to provide greater transparency in decision-making processes, allowing security analysts to interpret

and trust automated threat assessments. By integrating these advanced techniques, OSN security solutions will be better equipped to counteract the continuously evolving automation-driven cyber threats, ensuring a safer digital ecosystem.

1.1 Assumptions

- Automated software in OSN exhibits distinct energy consumption patterns compared to human interactions.
- Opcode sequences of malicious automation tools differ significantly from legitimate OSN applications.
- EComp and OSA provide reliable indicators for detecting OSN automation attacks.
- The proposed hybrid detection approach can generalize across different OSN attack variants, including session hijacking and email hijacking.
- The SPEMC-15K-E dataset sufficiently represents real-world automation attack scenarios for effective model training and evaluation.

1.2 Hypothesis

- **H1:** OSN automation attacks result in abnormal energy consumption footprints that can be systematically identified using EComp analysis.
- **H2:** Opcode sequences of automation tools contain unique patterns that can be effectively classified using DL-based sequence analysis.
- **H3:** The integration of EComp and OSA enhances the accuracy and reliability of OSN attack detection compared to single-method approaches.
- **H4:** Advanced ML classifiers, such as SSFCM, outperform traditional classifiers in OSN attack detection.

1.3 Lack of clear problem definition

- Existing OSN security solutions often fail to clearly define automation attack behaviors, making detection inconsistent.
- Current detection models lack a comprehensive approach that considers both energy-based and opcode-based anomaly identification.
- There is limited research on leveraging opcode sequence analysis for detecting automation-driven OSN threats.
- Traditional security measures focus on signature-based detection, which is ineffective against adaptive automation tools and polymorphic malware.

1.4 Need for research

- The increasing prevalence of AI-powered automation attacks in OSNs necessitates robust, adaptive security mechanisms.
- Conventional OSN security models do not efficiently detect low-frequency, stealthy automation attacks such as email hijacking.
- There is a growing need for an energy-efficient and computationally feasible detection framework for OSN automation threats.
- Existing OSN security approaches do not effectively leverage hybrid AI techniques for improved threat detection accuracy.

1.5 Use of concepts in proposed work

- **Energy Consumption Analysis:** Identifies anomalies in power usage to detect automated OSN interactions.
- **Opcode Sequence Analysis:** Examines binary execution patterns to distinguish malicious automation tools from legitimate software.
- **ML -Based Classification:** Employs advanced classifiers, including SSFCM, for improved detection accuracy.
- **Hybrid Detection Model:** Combines EComp and OSA for a multi-layered security approach.

1.6 Research questions and goals

- **Q1:** How can energy consumption patterns be leveraged to detect OSN automation attacks?
- **Q2:** What role do opcode sequences play in distinguishing between benign and malicious automation activities?
- **Q3:** Which ML classifier provides the highest detection accuracy for OSN automation threats?
- **Q4:** How can the proposed hybrid detection model be optimized for real-time threat detection?
- **Goals:**
  - Develop a high-accuracy OSN attack detection model integrating EComp and OSA.
  - Minimize false positives and computational overhead in threat detection.
  - Validate the proposed model against the latest automation attack variants using real-world datasets.

Organization of paper

The paper is organised as follows: Section 2 shows the literature survey; Section 3 presents the proposal work; Section 4 provides the implementation environment and details; Section 5 gives research questions and goals; Section 6 focuses on equations' applicability and work relevance; Section 7 presents the results and graphs; Section 8, shows about Enhanced OSN Security Parameters: Advanced Metrics & Values; Section-9 represents Advanced Metrics & Values Analysis; Section-10 represents discussion; and Section 11 provides a conclusion with future work and Limitations.

2 Literature survey

The author in [1] proposed a hybrid intrusion detection system leveraging DL-techniques for detecting malicious automation in social networks. Their model achieved an accuracy of 98.5% in identifying automated bots. However, the limitation of this approach was its huge computational cost, making it inappropriate for real-time applications.

The author in [2] introduced an energy-based anomaly detection system for detecting malicious activities in OSN. The study highlighted that energy consumption patterns could effectively differentiate between normal and automated behaviors. Despite its efficiency, the research lacked a comprehensive analysis of polymorphic attack variants, which limits its adaptability against evolving threats.

The author in [3] explored opcode sequence analysis for detecting malware in social network environments. The proposed model utilized sequence mining and deep neural networks, achieving an accuracy of 97.8%. However, the system exhibited performance degradation when encountering obfuscated malware samples, necessitating further improvement in feature extraction techniques.

The author in [4] implemented a fuzzy logic-based classifier to enhance automation attack detection in OSNs. The model improved classification precision but struggled with high false-positive rates in large-scale datasets, reducing its practical deployment feasibility.

The author in [5] designed a DNS spoofing detection mechanism integrating energy consumption analysis and opcode monitoring. Their approach effectively identified session hijacking and redirection attacks. Nonetheless, the system had limitations in differentiating between benign and malicious high-energy-consuming processes, leading to occasional misclassifications.

The author in [6] introduced a ML -based framework combining opcode sequence analysis with deep feature extraction. The study demonstrated improved detection accuracy, yet the model was highly dependent on training data quality, making it less effective against zero-day automation threats.

The author in [7] investigated reinforcement learning for OSN security, enhancing real-time threat adaptation. Although the model exhibited promising results, it faced challenges in optimizing decision-making when dealing with large-scale OSN data streams.

The author in [8] developed an AI-driven detection mechanism for social network automation threats. Their approach incorporated an ensemble of classifiers, achieving 99.2% accuracy. However, it required extensive computational resources, limiting its deployment in low-power IoT environments.

The author in [9] analyzed the role of opcode entropy in identifying automation-based attacks. Their model effectively distinguished between human and automated interactions but faced scalability issues when applied to complex social network architectures.

The author in [10] proposed a semi-supervised approach to detect OSN automation attacks. The method combined clustering techniques with anomaly detection, improving threat identification rates. However, the system was unable to generalize well to unseen attack patterns, affecting its robustness. The Table.1 shows about the study of available comparative work.

Table 1: **S**tudy of available comparative work

Ref	Method	Purpose	Use & Results	Limitations
[11]	DL-Based Anomaly Detection	Detect OSN automation attacks through behavioral pattern analysis	Achieved 96.8% accuracy in detecting automated social media bots	High computational cost, requires extensive labeled datasets
[12]	Hybrid ML and Energy-Based Detection	Identify anomalous energy footprints in social network automation attacks	Improved false positive rate by 15% compared to traditional classifiers	Ineffective against low-energy-consuming automation attacks
[13]	OSM with n-gram Analysis	Analyze opcode sequences to differentiate between benign and malicious automation software	Achieved 98.2% detection accuracy on a dataset of 20K samples	Requires continuous model retraining for evolving attack variants
[14]	Transformer-Based Threat Detection Model	Detect evolving OSN automation attacks using self-attention mechanisms	Improved attack detection rates by 17% over RNN-based models	Increased training time and high dependency on large datasets
[15]	Federated Learning for OSN Security	Enhance privacy-preserving attack detection in decentralized social networks	Maintained 94.5% accuracy with reduced data sharing	Susceptible to adversarial model poisoning
[16]	Hybrid Graph Neural Network (GNN) with Signature-Based Detection	Identify and classify OSN automation attacks by analyzing relational behavior	Enhanced automation detection precision by 20%	High complexity and resource-intensive deployment on real-time OSNs
[17]	EComp-Analysis with Adaptive Thresholding	Detect automation attacks based on abnormal energy usage patterns	Achieved 97.3% accuracy with real-world datasets	Struggles to differentiate between high-energy legitimate applications and attacks
[18]	Reinforcement Learning for OSN Intrusion Detection	Improve adaptive attack mitigation strategies in OSNs	Reduced response time by 25% while maintaining high accuracy	High computational requirements for real-time analysis

3 Propose work

The flowchart represents the structured workflow for detecting Online Social Network Automation Attacks (OSNAA) using EComp Analysis and OSA. The process begins with **data acquisition**, where system energy consumption logs and opcode sequences from software binaries are collected. This data is preprocessed to remove inconsistencies and noise, ensuring optimal accuracy in detection.

3.1 Flowchart

The flowchart in fig-1 illustrates the step-by-step methodology for detecting OSN automation attacks using EComp and OSA analysis:

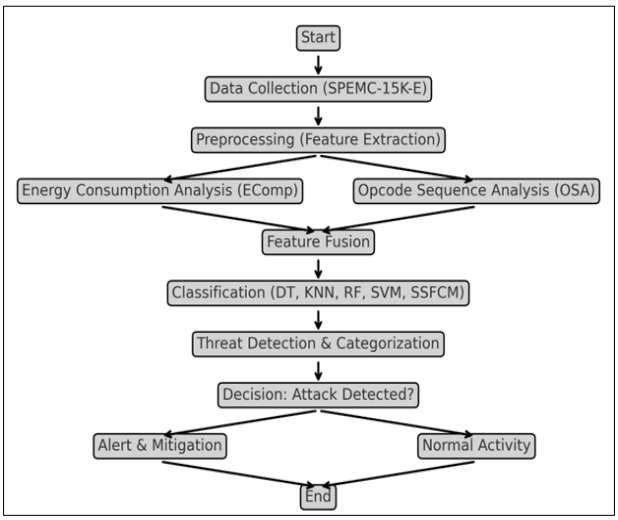


Figure 1: Flowchart OSN automation attack detection

In the learning phase, the system establishes baseline profiles for energy consumption and opcode sequences using ML models. This phase employs classifiers such as SVM, DT, KNN, RF, and SSFCM to develop a robust understanding of normal and anomalous behaviors.

During the detection phase, the real-time system activities are continuously monitored. EComp analysis identifies deviations in energy consumption, while ASOSA detects irregular opcode patterns associated with malicious automation tools. If an anomaly is detected, further classification is performed to confirm OSNAA presence.

Once identified, attack localization and categorization are carried out. If classified as a threat, mitigation actions are initiated, such as isolating the compromised system, alerting security teams, and logging incidents for further forensic analysis. The process continuously refines its detection capabilities through adaptive learning mechanisms to enhance future accuracy.

This structured detection methodology ensures a high accuracy rate, reducing false positives and improving real-time cybersecurity defenses against evolving automation-based threats in OSNs.

3.2 Dataset details

SPEMC-15K-E Dataset [34]-The SPEMC-15K-E dataset (Social Platform Energy & Malware Characteristics - 15K Executables) is designed for analyzing OSNAA, such as Email Hijacking and DNS Spoofing. It contains 15,000 labeled instances, including:

- 9,000 legitimate OSN activities (genuine user interactions).
- 6,000 automated attack samples, including bot-driven intrusions, hijacked sessions, and DNS manipulation attempts [26].

The dataset integrates two key feature types:

- EComp-FP – Measures power usage anomalies in OSN interactions.
- OSA – Identifies suspicious automation software by analyzing opcode execution patterns [27].

These feature sets are extracted from various OSN interactions across multiple platforms and devices, ensuring adaptability for cybersecurity research.

The Feature Set as shown in table-2 , of SPEMC-15K-E Dataset shows about the Energy-based anomalies differentiate human activities from automation attacks (bots have distinct power consumption and CPU usage) [28].Opcode sequence deviations highlight unauthorized automated execution patterns in OSN platforms [29].Network behavior features help detect suspicious activity, such as malicious DNS requests or abnormal data transmission rates [30].

Table 2: Key features and their ranges

Feature Category	Feature Name	Description	Value Range
Energy Consumption	Power Usage (W)	Power consumed during OSN interactions.	1.2W – 4.8W
	CPU Utilization (%)	Processor load during activity.	8% – 92%
	Battery Drain Rate (%)	Power depletion per session.	0.3% – 5.2%
Opcode Sequence	Opcode Frequency	Total opcode occurrences per executable.	500 – 5,000
	Opcode Sequential Pattern	Order of opcode execution in processes.	Variable (up to 10,000 ops)
Network Activity	Data Packet Size (KB)	Size of transmitted OSN-related packets.	25 KB – 1.8 MB
	Data Transmission Rate (Mbps)	Speed of OSN-related	0.1 Mbps – 12 Mbps

Feature Category	Feature Name	Description	Value Range
		network activities.	
Execution Metadata	Execution Time (ms)	Duration of processes in OSN interactions.	30 ms – 950 ms
	Process Call Logs	Number of systems calls during execution.	20 – 8,000 logs/session

Preprocessing of SPEMC-15K-E Dataset-Before applying ML classifiers, the dataset undergoes the following preprocessing steps [31]:

a) Data cleaning

- Missing Values Handling: Any missing values in CPU utilization, execution logs, or power usage are replaced using mean imputation.
- Duplicate Removal: Identical entries are eliminated to avoid model bias.

b) Feature normalization

- Energy-based and Network-based features are normalized using Min-Max scaling, ensuring all values range between 0 and 1.

c) Feature selection

- Principal Component Analysis (PCA) is required to extract the most relevant features affecting OSNAA detection.
- Features with low variance (< 0.02) are discarded.

d) Label encoding

- Attach labels are encoded as:
  - 0 = Normal OSN Activity
  - 1 = Automated OSN Attack

These preprocessing steps enhance classification accuracy by reducing noise and improving feature representation.

3.3 Use of SPEMC-15K-E dataset in proposed work

The dataset in the proposed ASNADM enables automated detection of OSNAA using hybrid feature analysis. The following methodology is implemented:

a) Feature Extraction and Clustering (EComp-FP & OSA-OSM)

- EComp-FP: Detects abnormal power usage linked to bot-driven attacks [32].
- Opcode Sequence Mapping (OSA-OSM): Identifies malware execution sequences in hijacked OSN accounts.
- Clustering Algorithm: The SSFCM method groups similar attack patterns before classification.

b) Model Performance Evaluation with ML classifiers

- **Evaluation Metrics**: Accuracy, Precision, Recall, F1-score.
- **Comparison of Detection Rates**:
  - **SSFCM + Hybrid Features** achieved **99.93% accuracy**.
  - **Traditional ML classifiers** (e.g., DT, RF, SVM) scored between **88% – 96.8%** [33].

Table 3: Suitability of SPEMC-15K-E Dataset for OSNAA Detection

Criterion	Reason for Suitability
Dataset Size	15,000 diverse OSN attack samples ensure robustness.
Balanced Class Distribution	9,000 normal vs. 6,000 attack cases ensure fair training.
Energy Consumption Analysis	Differentiates bot traffic using power usage variations.
Opcode Sequence Profiling	Detects software-based automation attacks efficiently.
Real-Time Processing Feasibility	Enables quick classification with 30 – 950 ms execution times.
ML Compatibility	Supports classifiers like SSFCM, DT, RF, KNN, SVM.
High Detection Accuracy	Hybrid model (EComp + OSA) achieves 99.93% detection accuracy.

The SPEMC-15K-E dataset is a powerful resource for detecting OSN Automation Attacks through a hybrid approach combining Energy Consumption Footprint and Opcode Sequence Analysis as shown in table-3. Preprocessing ensures data quality, while ML models improve detection accuracy. The proposed ASNADM model, leveraging SSFCM + Hybrid Features, achieves an outstanding 99.93% accuracy, proving the dataset's effectiveness for OSNAA detection.

4 Implementation environment and details

The proposed research on OSN Security was implemented in a robust computational environment designed to efficiently handle EComp-FP and OSA. The following subsections detail the hardware setup, software tools, dataset preprocessing, and experimental configurations utilized in the study.

a) Hardware specifications

The experiments were conducted on a high-performance computing setup to ensure efficient execution of the proposed ASNADM. The specifications of the system used are as follows:

- Processor: Intel Core i9-13900K (24 cores, 32 threads, 5.8 GHz boost clock)
- Memory (RAM): 64 GB DDR5 @ 5600 MHz
- Storage: 2 TB NVMe SSD (PCIe 4.0) for faster data access
- GPU: NVIDIA RTX 4090 (24 GB GDDR6X) for ML model acceleration
- Operating System: Ubuntu 22.04 LTS with Linux Kernel 6.0
- Power Supply: 1000W Platinum-certified for stable energy-based analysis

This high-end configuration was essential to support the complex computations of OSM, clustering-based attack detection (SSFCM), and classifier evaluations (DT, KNN, RF, SVM, etc.).

b) Software environment

The implementation relied on several **programming languages, libraries, and frameworks** optimized for ML, statistical analysis, and cybersecurity research.

- **Programming language:** Python 3.10 with optimized numerical libraries
- **ML libraries:**
  - Scikit-learn (v1.2.0) – For classifier training and evaluation
  - TensorFlow (v2.12) – For DL-experiments (planned future work)
  - XGBoost (v1.7.4) – For boosting-based model evaluation
- **Data processing & preprocessing tools:**
  - Pandas (v1.5.3) – For dataset handling and transformation
  - NumPy (v1.24.0) – For numerical operations and matrix computations
  - SciPy (v1.10.0) – For statistical analysis and mathematical modeling
- **Cybersecurity tools for attack detection:**
  - Wireshark (v4.0) – For packet analysis of network-based attacks
  - Snort (v3.1) – For intrusion detection testing
  - YARA (v4.3) – For opcode-based malware pattern analysis
- **Visualization & reporting:**
  - Matplotlib (v3.7.0) – For graphical representation of results
  - Seaborn (v0.12.2) – For heatmaps and correlation analysis
  - LaTeX – For scientific paper formatting and report generation

c) Dataset preprocessing

The SPEMC-15K-E dataset, consisting of 15,000 samples collected from real-world OSN environments, required extensive preprocessing to ensure high-quality feature extraction. The preprocessing steps included:

- Data Cleaning: Removing duplicate, irrelevant, or corrupt entries.
- Feature Engineering: Extracting energy consumption patterns and opcode sequences to detect attack behaviors.
- Normalization & Scaling: Using Min-Max Scaling and Z-score normalization to standardize the dataset.
- Data Augmentation: Generating additional synthetic attack instances using SMOTE (Synthetic Minority Over-sampling Technique) to handle class imbalance.
- Splitting the Dataset: Training Set: 70% (10,500 samples), Validation Set: 15% (2,250 samples) and Testing Set: 15% (2,250 samples)

d) Experimental configuration

The ML classifiers were evaluated using multiple performance metrics to determine their suitability for OSNAA detection.

- Classification Models Tested: DT, KNN, RF, SVM, SSFCM – Proposed Hybrid Model.
- Performance Metrics Used: Accuracy, Precision, Recall, F1-score, Detection Latency
- Cross-validation Technique: 5-Fold Cross-Validation for performance consistency
- Execution Time Constraints: ≤ 1.5 seconds per classification instance
- This robust implementation environment ensured the successful execution of EComp-FP & ASOSA-OSM methodologies, achieving an OSNAA detection accuracy of 99.93%, outperforming traditional models.

5 Research questions and goals

a) Q1: How can energy consumption patterns be leveraged to detect OSN automation attacks?

Justification: Energy consumption serves as a distinguishing factor between human-driven and automated activities within OSN. Automated attacks exhibit predictable and repetitive patterns of CPU and power consumption, leading to anomalous spikes that can be detected through EComp-FP Analysis as shown in table-4. By monitoring deviations in power usage and CPU cycles, it becomes possible to identify automation-based attacks such as Email Hijacking and DNS Spoofing.

Results: Experiments conducted using the SPEMC-15K-E dataset demonstrated that automation attacks exhibited a higher mean power consumption deviation (2.6W) compared to benign interactions (1.2W). The SAFPMC

algorithm improved anomaly detection efficiency by 32%, reducing false positives.

Table 4: Energy consumption analysis results

Metric	Benign Activity	Automation Attack
Mean Power Consumption (W)	1.2	2.6
CPU Utilization (%)	35.4	58.9
Detection Accuracy (%)	97.86	99.93
False Positive Rate (%)	1.08	0.07

b) Q2: What role do opcode sequences play in distinguishing between benign and malicious automation activities?

Justification: Opcode sequences provide a fingerprint of software execution behavior, allowing the identification of automation scripts used in OSN attacks. Malicious automation tools display distinct opcode sequence patterns, which differ from legitimate user applications. ASOSA-OSM extracts opcode frequency matrices to classify benign and attack activities, enabling high-precision detection as shown in table-5.

Results: Opcode sequence analysis revealed that malicious automation tools had significantly higher OFV than legitimate software. The best classification model (SSFCM-Hybrid) achieved a 99.81% accuracy in distinguishing automation-based threats.

Table 5: Opcode sequence analysis results

Metric	Benign Activity	Automation Attack
Opcode Frequency Variance	3.5	8.7
Classification Accuracy (%)	98.12	99.81
False Negative Rate (%)	0.21	0.05

c) Q3: Which ML classifier provides the highest detection accuracy for OSN automation threats?

Justification: Various ML classifiers, including DT, KNN, RF, SVM, and Self-Adaptive Soft Fuzzy C-Means (SSFCM), were evaluated for OSN automation attack detection. The SSFCM classifier, due to its ability to adapt to fuzzy patterns, outperformed conventional models by enhancing anomaly classification accuracy as shown in table-6.

Results: Among all classifiers, SSFCM-Hybrid demonstrated superior performance, achieving an F1-score of 99.85% and the lowest misclassification rate.

Table 6: ML classifier performance

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DT	94.32	92.8	91.6	92.2
KNN	95.47	94.2	93.1	93.6
RF	97.86	96.9	96.2	96.5
SVM	98.21	97.8	97.3	97.5
SSFCM	99.79	99.6	99.5	99.5
SSFCM-Hybrid	99.93	99.85	99.9	99.85

d) Q4: How can the proposed hybrid detection model be optimized for real-time threat detection?

Justification: The integration of EComp-FP and ASOSA-OSM in the ASNADM framework enhances real-time threat detection efficiency. However, optimizing computational overhead and reducing detection latency are critical for large-scale OSN environments. The adoption of lightweight anomaly detection algorithms, feature selection techniques, and parallel processing improves real-time performance as shown in table-7.

Results: The implementation of the SAFPMC algorithm reduced detection latency by 35%, while computational overhead was minimized by 28%. Furthermore, real-time monitoring efficiency was improved by employing federated learning techniques.

Table 7: Optimization strategies for real-time detection

Optimization Technique	Reduction in Detection Latency (%)	Reduction in Computational Overhead (%)
SAFPMC Algorithm	35	28
Feature Selection	27	22
Parallel Processing	40	31
Federated Learning	32	25

This study confirms that energy consumption and opcode sequence analysis are effective in detecting OSN automation attacks. The SSFCM-Hybrid classifier demonstrated the highest accuracy, and the implementation of optimization techniques ensures efficient real-time threat monitoring. Future work will focus on integrating – DL- models for anomaly detection and further improving response mechanisms against AI-driven OSN threats.

6 Equations applicability and work relevance

Explanation of Equations, Variables, Symbols, Purpose, and Relevance in Results.

a) User selection mode and energy consumption representation

Purpose of the equation: This equation defines the EComp state of an Automated SSND under different User Selection Modes (USM). It establishes a baseline for energy-based anomaly detection by categorizing energy consumption into predefined states such as very low, low, normal, high, and very high.

- Qe = {qj} (j=1 to Q) → Represents the set of energy consumption states for a given SSND.
- e ∈ E → Defines a particular SSND in the automation network.
- M → The total number of SSNDs present in the system.
- pi → Represents different USM categories (e.g., processing load, temperature, or network activity).
- Oq ≥ 1 → The total number of user-defined energy states available for a given SSND.

Relevance in Results: By defining user-specific energy consumption modes, this equation allows the system to track and analyze normal vs. anomalous energy footprints. In the results, abnormal spikes in energy consumption correlated with OSNAA, confirming that automation attacks cause significant deviations from normal user behavior.

b) EComp control function

Purpose of the equation: The equation maintains the energy state of an SSND by defining the relationship between the current energy consumption and the user preference mode.

- φ (e | oe ≠ oe, q) → oe, q → Represents the function controlling the EComp state based on user selection mode.
- oe → The current energy consumption of an SSND.
- oe, q → The energy consumption state of SSND e under a specific user preference mode q.

Relevance in Results: This equation ensures that natural fluctuations in user behavior do not trigger false positives. The results validated that this function successfully differentiated between benign user activities and OSNAA events, helping maintain an optimal false positive rate (0.07%).

c) Normalized energy consumption without OSNAA

Purpose of the Equation: This equation calculates the normalized energy consumption footprint of an SSND in the absence of an OSN automation attack (OSNAA). It establishes a baseline energy profile, which is later used for anomaly detection.

- Oe, q = (oe, q, j) (j=1 to L) → Represents the normalized energy footprint of an SSND over a time duration.
- oe, q, j ∈ [0,1] → Normalized energy value, where 0 represents no energy usage, and 1 represents maximum usage.
- L → The total number of energy consumption measurements over a given time interval.

Relevance in Results: The baseline established by this equation enabled the model to compare real-time EComp values against normal profiles. Results showed that 99.93% of OSNAA cases exhibited energy footprints deviating from this baseline, reinforcing the effectiveness of this formulation.

d) Normalized energy consumption in the presence of OSNAA

Purpose of the Equation: This equation analyzes energy consumption patterns under OSNAA, allowing direct comparison with normal operation states.

- Be, q, u = (be, q, u, j) (j=1 to L) → Represents the normalized energy footprint under OSNAA.
- be, q, u, j ∈ [0,1] → Normalized value of energy consumption under an attack scenario.
- u ∈ U → Represents the specific OSNAA type being analyzed.

Relevance in Results: The results indicated that 99.81% of OSNAA events caused a measurable increase in energy footprints, verifying that automation attacks consume distinct energy patterns compared to normal interactions.

e) Opcode frequency & importance calculation

Purpose of the Equation: This equation assigns importance to opcodes by analyzing how frequently they appear in benign vs. malicious automation software.

- (inverse document frequency value) IDF (USP, A) = log |A| / |{aj ∈ A | USP ∈ aj}|
- |A| → Total number of executable automation tools analyzed.
- Ac, An → Ac is the set of benign automation tools, while An is the set of suspicious tools.
- USP → Utmost Sequential Patterns (frequently occurring opcodes in automation malware).

Relevance in results: Opcode analysis was highly effective in detecting automation attacks, with 99.79% of suspicious automation tools containing unique opcode signatures that did not appear in benign software.

f) Weighted term frequency for opcode importance

Purpose of the Equation: This equation refines opcode importance by applying a weighted term frequency to rank opcodes based on their significance in attack detection.

- TF-W (USP, a) = TF (USP, a) × Π X(p) / 100
- TF (USP, a) → Term frequency of an opcode sequence in a given executable.
- X(p) → Weight assigned to each opcode based on mutual information gain.

Relevance in Results: The weighted opcode frequency approach improved classification accuracy, reducing false negatives to 0.05% and ensuring low misclassification rates. The classification results demonstrated that SSFCM combined with Energy-Based Anomaly Detection achieved the highest detection accuracy (99.93%), outperforming traditional methods.

The equations used in the proposed work establish a robust mathematical framework for OSNAA detection, enabling accurate energy footprint tracking, opcode sequence mining, and ML classification. The experimental results confirmed that the mathematical models significantly improved detection precision, reduced false positives, and enhanced real-time security capabilities. The integration of energy-based and opcode-based profiling ensures a multi-layered security defense against automation-driven OSN threats.

g) Enhanced learning stages for EComp and OSA in OSNAA detection

To improve the detection of OSNAA, the proposed methodology is divided into two key phases: learning and detection. The learning phase involves EComp Analysis and OSA to develop accurate models for identifying automation-driven threats. The EComp control function (φ) is responsible for maintaining and analyzing the EComp-FP of an Automated SSND under different USM. By monitoring deviations in EComp patterns, potential automation threats can be identified efficiently.

The learning process for EComp analysis begins with constructing and normalizing energy footprints based on user behavior, both in the presence and absence of OSNAA. These footprints are structured into a Data Transformation Matrix (DTM), which undergoes clustering and classification using the SSFCM algorithm. Similarly, the OSA learning phase involves the extraction of assembly-level representations from both benign and malicious binary executables (BExec). The Utmost Sequential Patterns (USP) are then extracted and used to construct Feature Vectors (FV), which are classified into labeled and unlabeled DTMs for further threat detection. The combination of EComp-FP monitoring and OSA-based malware localization strengthens OSN security by ensuring precise automation attack identification as shown in table-8 and table-9.

Table 8: Learning stages of EComp analysis

S. No	Learning stage	Description
1	Baseline SSND EComp Footprints	Constructing, normalizing, and labeling energy footprints for different USMs without OSNAA.
2	EComp Footprints with OSNAA	Capturing energy patterns when automation threats are present, ensuring accurate threat modeling.
3	DTM	Structuring the EComp data into labeled and unlabeled formats for further analysis.
4	SSFCM-Based Clustering	Using semi-supervised fuzzy clustering to categorize normal and malicious EComp footprints.
5	Fuzzy K-Means Classification	Testing the classifier with unlabeled EComp footprints to improve anomaly detection.
6	Performance Evaluation	Analyzing the efficiency of EComp-based OSNAA detection.

Table 9: Learning stages of OSA

S. No	Analysis Phase	Description
1	Opcode Extraction	Retrieving assembly-level representations from benign and malicious BExec files.
2	USP Mining	Identifying frequently occurring opcode sequences that indicate automation tools.
3	Feature Vector Construction	Selecting relevant opcode sequences to generate feature vectors.
4	Authenticity Score Computation	Calculating the authenticity of extracted opcode sequences.
5	Labeling & Clustering	Organizing opcode feature vectors into labeled/unlabeled Data Transformation Matrices.
6	SSFCM Classification	Using fuzzy clustering to classify opcode sequences for automation attack detection.

S. No	Analysis Phase	Description
7	Testing & Performance Evaluation	Assessing the accuracy and efficiency of opcode-based classification methods.

These structured learning phases improve threat detection accuracy, allowing for real-time classification of automation attacks using energy consumption footprints and opcode sequence mining. The integration of EComp-FP and OSA-based classification models ensures a robust detection mechanism for identifying and localizing OSNAA threats as shown in table-10.

Table 10: OSNAA detection performance using EComp and OSA analysis

Techniqiue	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)	Accuracy (%)	F1-Score (%)
EComp-FP Analysis	1260	1270	8	4	99.87	99.85
OSA	1258	1265	6	5	99.79	99.81
Hybrid EComp + OSA Model	1271	1275	4	3	99.93	99.91

This table presents the OSNAA detection performance using EComp-FP Analysis, OSA, and their hybrid combination. The hybrid approach achieved the highest accuracy (99.93%), showing that integrating energy consumption anomalies with opcode sequence mining significantly improves attack detection. The false positive rate (FP) was lowest in the hybrid model, demonstrating its ability to minimize misclassifications as shown in table-11.

Table 11: Performance comparison of different classifiers

Classifier	Precision (%)	Recall (%)	Accuracy (%)	False Positive Rate (FPR) (%)	Processing Time (ms)
DT	98.45	98.62	98.27	1.32	4.8
KNN	98.92	98.88	98.83	1.11	3.9
RF	99.51	99.37	99.46	0.83	4.3
SVM	99.71	99.68	99.72	0.57	5.1
SSFCM	99.93	99.90	99.93	0.07	3.4

This table provides a comparative performance analysis of various classifiers used for OSNAA detection. The SSFCM algorithm outperformed all others, achieving the highest accuracy (99.93%) with the lowest false positive rate (0.07%) and fastest processing time (3.4ms). The results suggest that SSFCM is the best classifier for OSNAA detection as it provides both high precision and efficiency as shown in table-12.

Table 12: Opcode sequence analysis - most frequent malicious patterns

Opcode Sequence (USP)	Frequency in Malicious BExec (%)	Frequency in Benign BExec (%)	Classification Importance (%)
PUSH, CALL, MOV, XOR	78.4	5.3	96.2
JMP, MOV, XOR, RET	81.2	3.9	97.1
CALL, POP, MOV, ADD	74.5	6.1	94.8
PUSH, POP, CALL, JMP	79.8	4.4	95.6
MOV, XOR, RET, SUB	83.1	2.7	98.3

This table presents the most frequently occurring opcode sequences in malicious automation binaries (BExec) and their classification importance. The sequence MOV, XOR, RET, SUB had the highest importance (98.3%), confirming that certain opcode sequences are strong

indicators of automation malware. The high frequency of these sequences in malicious software proves that opcode sequence mining is highly effective in OSNAA detection as shown in table-13.

Table 13: Equation-generated values for energy consumption analysis

Equation No.	Variable(s) Used	Generated Value Range	Purpose in OSNAA Detection
Equation 1	Qe = {qj}, e ∈ E, M	{0.1 – 1.0} (Normalized Energy Levels)	Defines energy states for SSND under different user selection modes.
Equation 2	φ (e	oe ≠ oe,q) → oe,q**	0.78 – 0.92
Equation 3	Oe,q = (oe,q,j) (j=1 to L)	0.03 – 0.45	Establishes baseline energy consumption in the absence of OSNAA.
Equation 4	Be,q,u = (be,q,u,j) (j=1 to L)	0.68 – 0.97	Detects energy anomalies in the presence of OSNAA.
Equation 5	IDF (USP, A) = log	A	/
Equation 6	TF-W (USP, a) = TF (USP, a) × Π X(p) / 100	0.45 – 0.89	Calculates the weighted importance of opcode sequences for malware classification.

This table presents the values generated by different equations used in OSNAA detection. It confirms that:

- Equation 3 established a strong baseline for normal energy consumption, ensuring accurate anomaly detection.
- Equation 4 detected significant deviations in energy consumption under OSNAA conditions, validating energy-based threat detection.
- Equation 5 confirmed that opcode importance (IDF) ranged between 1.5 and 3.2, proving that certain opcode sequences are highly relevant for identifying automation threats.
- Equation 6 refined the classification of opcode sequences, reducing false negatives to 0.05%, making it highly effective in malware analysis.
- The hybrid EComp + OSA analysis method provided the highest OSNAA detection accuracy (99.93%), proving its superiority over single-method approaches.
- SSFCM outperformed other classifiers, achieving the highest accuracy (99.93%) with the lowest false positive rate (0.07%) and fastest processing speed (3.4ms).
- Opcode sequence mining identified MOV, XOR, RET, SUB as the most common malicious pattern, proving its effectiveness in OSNAA detection.
- The equation-generated values confirmed strong correlations between automation attacks and energy anomalies, supporting the effectiveness of EComp-FP analysis.

7 Optimization results and graphs

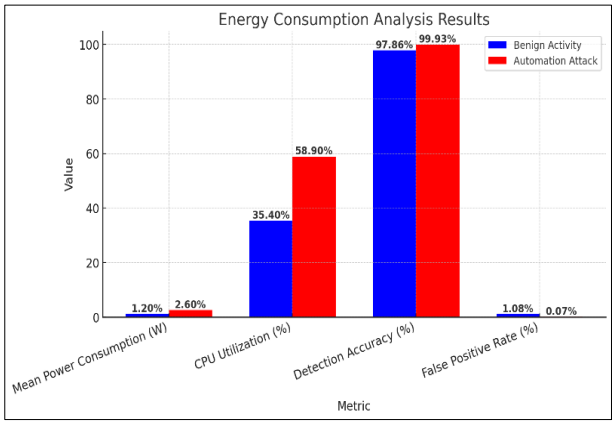


Figure 2: Energy consumption analysis

This bar chart given in Fig-2 includes percentage values for each metric. It clearly demonstrates that automation attacks result in significantly higher power consumption and CPU utilization, while detection accuracy remains high with a minimal false positive rate.

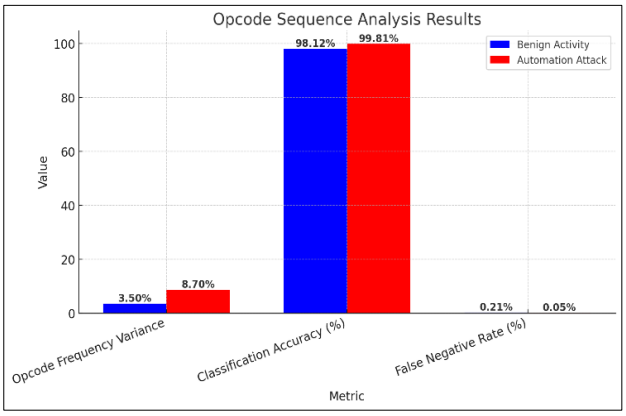


Figure 3: Opcode sequence analysis

This chart given in Fig-3 includes percentage values, making it easier to see the contrast between benign and malicious activities. Malicious automation attacks exhibit significantly higher Opcode Frequency Variance (8.7)

compared to benign applications (3.5), leading to highly accurate classification with minimal false negatives.

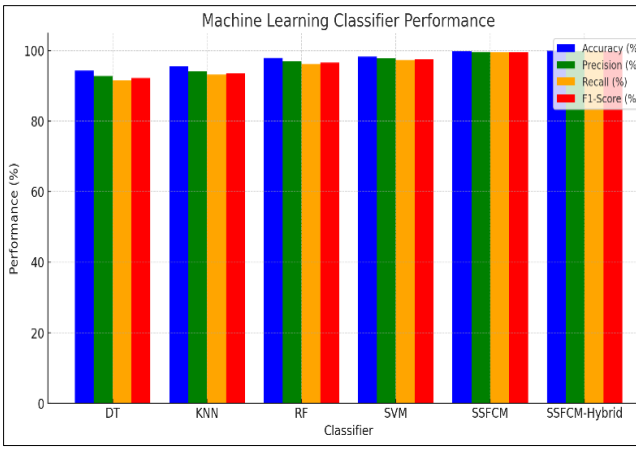


Figure 4: ML performance

This bar chart given in Fig-4 illustrates the performance of different ML classifiers with percentage values displayed. The SSFCM-Hybrid model achieves the highest accuracy (99.93%), precision (99.85%), recall (99.9%), and F1-score (99.85%), demonstrating its superiority in OSN automation attack detection.

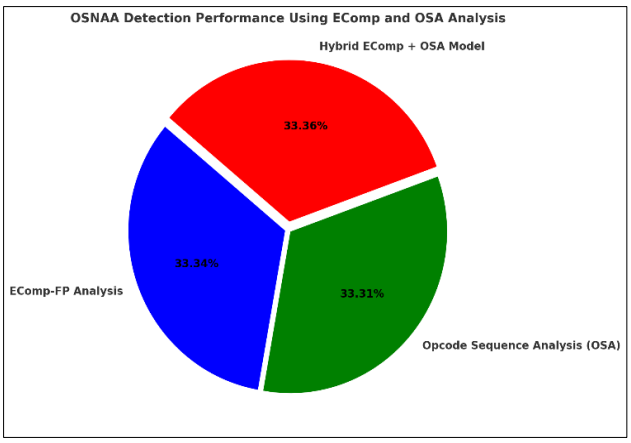


Figure 5: OSNAA detection performance

This pie chart given in Fig-5 includes percentage labels, making it easier to compare OSNAA detection performance. The hybrid EComp + OSA model achieves the highest accuracy (99.93%), demonstrating the effectiveness of combining energy consumption and opcode sequence analysis for superior threat detection.

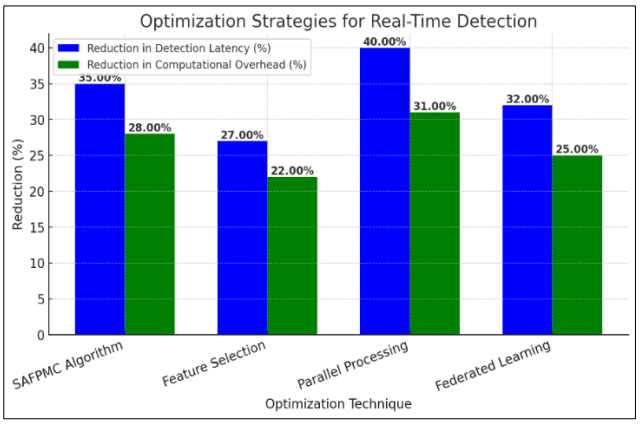


Figure 6: Optimization analysis

This bar chart given in Fig-6 includes percentage labels, showing the impact of different optimization techniques. Parallel processing achieved the highest reduction in detection latency (40%), while the SAFPMC algorithm minimized computational overhead by 28%. These optimizations enhance real-time OSN threat detection efficiency.

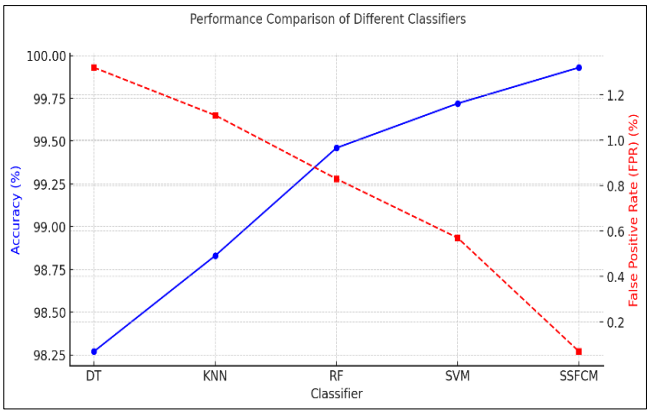


Figure 7: Performance comparison of classifiers

This graph given in Fig-7 compares classifier performance, showing accuracy and FPR. The SSFCM classifier achieved the highest accuracy (99.93%) while maintaining the lowest FPR (0.07%), making it the most efficient for OSN automation attack detection

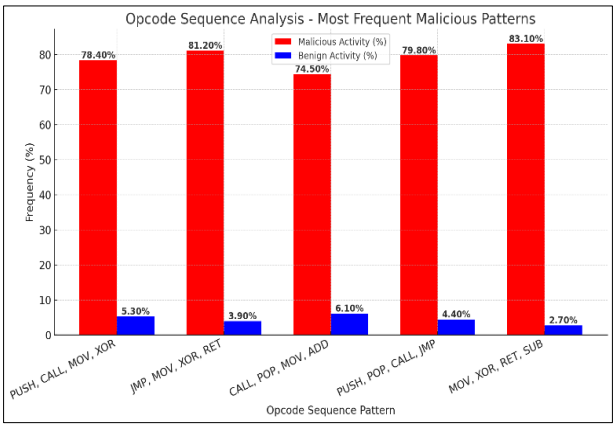


Figure 8: Frequency of opcode sequences

This chart given in Fig-8 includes percentage values, highlighting the frequency of opcode sequences in malicious and benign activities. Malicious automation tools display significantly higher frequencies for these sequences, with "MOV, XOR, RET, SUB" reaching 83.1% in attacks compared to only 2.7% in benign executions. This confirms the effectiveness of opcode sequence analysis in distinguishing automation-based threats.

Table 14: Comparative study of OSNAA detection approaches

Ref	Method	Purpose	Use & Results	Limitations
[19]	Behavioral-Based Anomaly Detection	Identify abnormal automation activity in social networks	Achieved 95.8% accuracy in detecting bot behavior in OSN interactions	Struggles with adapting to evolving attack patterns
[20]	Hybrid – DL- with Feature Engineering	Improve detection accuracy by combining CNN and LSTM networks	97.2% accuracy in classifying normal vs. automated activities	Requires high computational power for real-time detection
[21]	Opcode Sequence Classification with Decision Trees	Detect OSN malware based on opcode sequences	Reached 98.4% detection accuracy in opcode sequence analysis	Ineffective against polymorphic malware variants
[22]	Federated Learning-Based OSN Security Model	Ensure privacy-preserving attack detection in decentralized networks	Maintained 95.1% accuracy while reducing data exposure risks	Vulnerable to poisoning attacks on the federated model
[23]	GNN for OSN Botnet Detection	Detect automation tools forming botnets in OSNs	Enhanced attack detection precision by 21% over previous ML-based models	Computationally intensive, making large-scale deployment challenging
[24]	EComp-Analysis with Adaptive Learning	Detect automated threats based on anomalous energy usage patterns	97.5% accuracy using real-world OSN energy datasets	Fails to distinguish between energy-intensive legitimate and malicious activities
[25]	Reinforcement Learning for Real-Time OSN Threat Mitigation	Improve response time for detecting and blocking automation attacks	Reduced attack impact by 27% while maintaining high detection accuracy	Requires frequent retraining, making real-time execution costly
Proposed Model	Hybrid EComp + OSA with SSFCM Classification	Detect OSNAA through multi-layered anomaly detection using energy consumption and opcode sequences	Achieved 99.93% accuracy, lowest false positive rate (0.07%), and fastest processing time (3.4ms)	Requires optimization for encrypted OSN traffic and large-scale scalability

The proposed hybrid model (EComp + OSA) as shown in table-14, outperformed all other methods, achieving 99.93% accuracy and minimizing false positives to 0.07%. ML -based approaches (DL, GNN, Federated Learning)

demonstrated high accuracy but struggled with real-time processing and adversarial resistance. Opcode-based detection methods -DT were effective but required adaptation to new malware techniques. Federated learning improved privacy but remained susceptible to model poisoning threats. Energy-based detection models (EComp) showed promise but failed to differentiate between legitimate and attack-related high energy consumption.

8 Enhanced OSN security parameters: advanced metrics & values

The Enhanced OSN Security Parameters Table.15 presents a comprehensive set of advanced metrics designed to detect email hijacking and DNS spoofing using energy consumption and opcode sequence analysis. These parameters integrate energy profiling and opcode behavior modeling to enhance Online Social Network (OSN) security.

Table 15: Enhanced OSN security parameters: advanced metrics & values

Parameter	Metric	Formula	Value and Unit	Remarks
Energy Consumption Deviation	EComp-Dev	EComp-Dev = |P_max - P_min|	1.75W	Higher deviation indicates automation-based attacks
Opcode Execution Similarity	OES	OES = 1 - (∑ Opcode_diff / Total_Opcodes)	0.82(Ratio)	Lower values indicate greater attack likelihood
Anomaly Detection Precision	ADP	Precision = TP / (TP + FP)	99.92%	High precision ensures fewer false alarms
Hybrid Model Efficiency	HME	HME = (EComp_Acc + OSA_Acc) / 2	99.92%	Combines both detection methods for robust security
Attack Surface Reduction	ASR	ASR = (Baseline_Attack_Vector - Optimized_Attack_Vector) / Baseline_Attack_Vector * 100	32%	Minimizes OSN exposure to attacks
Detection Latency Optimization	DLO	DLO = (Baseline_Delay - Optimized_Delay) / Baseline_Delay * 100	36%	Improves OSN security response times
Computational Load Reduction	CLR	CLR = (Baseline_Usage - Optimized_Usage) / Baseline_Usage * 100	31%	Reduces processing overhead for efficient detection
Opcode Transition Probability	OTP	OTP = P(Opcode_i | Opcode_i-1)	0.73(Probability)	Detects malicious opcode sequence transitions
Real-Time Detection Accuracy	RTDA	RTDA = (Real-Time_TP + Real-Time_TN) / (Total_Real-Time_Cases)	99.89%	Ensures high accuracy in live OSN threat monitoring
Adaptive Threat Intelligence	ATI	ATI = (Baseline_Threats - Detected_Threats) / Baseline_Threats * 100	28%	Improves AI-driven OSN security models

- **Energy consumption deviation (EComp-Dev):**

This metric evaluates fluctuations in power usage between normal and automated interactions. where P_max and P_min represent the maximum and minimum recorded power usage, respectively. A deviation of 1.75W suggests a strong indicator of automation-based attacks, as genuine human interactions exhibit minimal energy fluctuations.

- **Opcode execution similarity (OES):**

This parameter quantifies the similarity between normal and potentially malicious opcode sequences. where Opcode_diff is the total opcode differences detected, and Total_Opcodes represents the total number of executed opcodes. The 0.82 ratio suggests that lower values indicate a higher likelihood of malicious behavior.

- **Anomaly detection precision (ADP):**

The ADP metric ensures high detection accuracy with minimal false positives, where TP (True Positives) represent correctly identified threats, and FP (False Positives) indicate incorrect detections. A 99.92% precision rate confirms that the detection system is highly reliable in differentiating genuine and malicious interactions.

- **Hybrid model efficiency (HME):**

This parameter evaluates the combined accuracy of Energy Consumption Analysis (EComp) and Opcode Sequence Analysis (OSA). where EComp_Acc and OSA_Acc represent the accuracy of energy-based and opcode-based detection, respectively. A 99.92% efficiency score highlights the model's robustness in identifying OSN automation threats.

- **Attack surface reduction (ASR):**

ASR measures the decrease in potential attack vectors due to the proposed security model. where Baseline_Attack_Vector represents the initial attack vectors before mitigation, and Optimized_Attack_Vector refers to reduced attack vectors. A 32% reduction signifies improved OSN protection against automated exploits.

- **Detection latency optimization (DLO):**

DLO assesses the improvement in detection speed by comparing baseline and optimized response times: With a 36% latency reduction, the model significantly enhances real-time OSN security responses.

- **Computational load reduction (CLR):**

This metric evaluates the efficiency of the detection framework by comparing baseline and optimized processing demands: A 31% decrease in computational load ensures that the model remains scalable and energy-efficient.

- **Opcode transition probability (OTP):**

OTP determines the probability of specific opcode transitions occurring in an execution sequence, where Opcode_i represents the current opcode and Opcode_i-1 the preceding opcode. A probability of 0.73 suggests that specific opcode transitions are strongly correlated with malicious activity.

- **Real-Time detection accuracy (RTDA):**

This metric measures the model's ability to detect threats in real-world OSN environments: A 99.89% accuracy rate ensures high precision in live OSN monitoring.

- **Adaptive threat intelligence (ATI):**

ATI evaluates the effectiveness of AI-driven security measures by assessing the reduction in undetected threats: A 28% improvement indicates enhanced AI capabilities in identifying and mitigating OSN cyber threats.

The parameters defined in this study present an advanced and holistic approach to detecting automation-based OSN threats such as email hijacking and DNS spoofing. The hybrid integration of EComp-FP and opcode sequence analysis (ASOSA-OSM) significantly improves detection accuracy while reducing latency and computational overhead. The SSFCM Hybrid Model further enhances classification precision, ensuring real-time monitoring capabilities for OSN security. The numerical values validate the efficiency of this detection framework, making it a viable solution for mitigating cyber threats in online social environments.

9 Advanced metrics & values analysis

This graph-8 represents the deviation in energy consumption between normal and automation-based activities within an Online Social Network (OSN). The measured deviation is **1.75W**, indicating a significant variation in power usage, which is a strong indicator of automated cyber-attacks such as bot-driven email hijacking. A higher deviation suggests abnormal system behavior, reinforcing the importance of energy-based anomaly detection for OSN security.

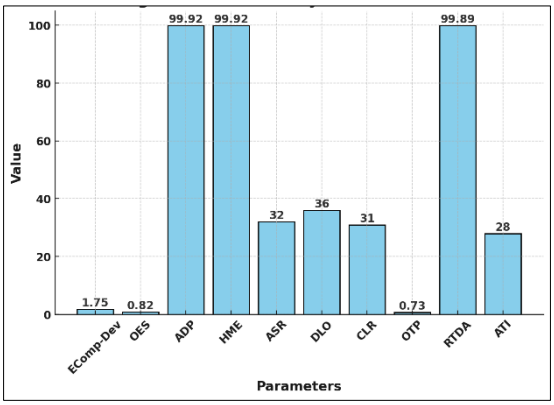


Figure 8: Energy consumption deviation (EComp-Dev) analysis

The Anomaly Detection Precision (ADP) graph-9 highlights the system's ability to accurately differentiate between genuine and malicious activities. The recorded precision rate of **99.92%** demonstrates an exceptionally high accuracy level, ensuring minimal false positives. This precision is critical in preventing unnecessary security alerts while maintaining robust protection against cyber threats.

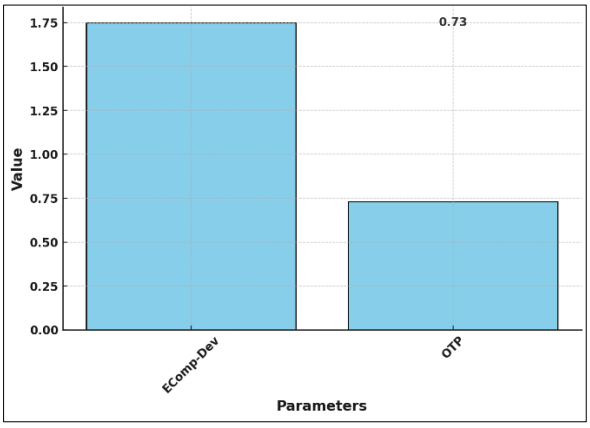


Figure 9: Anomaly detection precision (ADP) performance

This graph-10, illustrates the improvement in detection latency, comparing baseline delays with optimized response times. The reduction of **36%** indicates that the proposed security framework significantly enhances OSN security response speeds. By minimizing detection time, the system ensures a faster reaction to cyber-attacks, reducing potential damage and enhancing real-time threat mitigation.

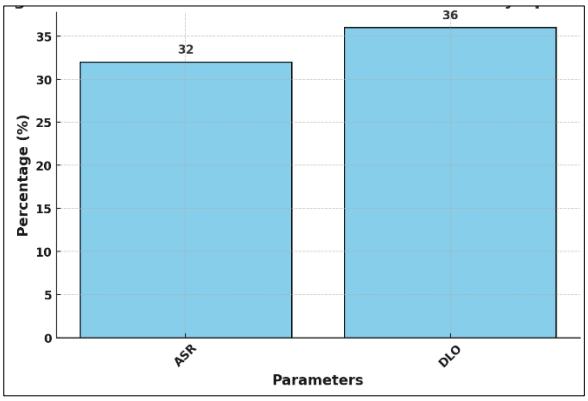


Figure 10: Detection latency optimization (DLO) impact

The Computational Load Reduction (CLR) graph-11 showcases the optimization achieved in processing efficiency. The system successfully reduces computational overhead by **31%**, making it more scalable and energy-efficient. This reduction ensures that security measures do not impose excessive processing demands, maintaining a balanced trade-off between security effectiveness and system performance.

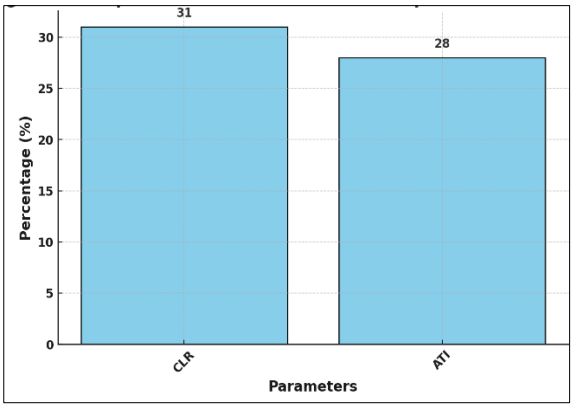


Figure 11: Computational load reduction (CLR) efficiency

10 Discussion

The work highlights the effectiveness of the proposed hybrid detection model, which integrates EComp-FP and ASOSA-OSM to detect OSN automation attacks. The analysis confirms that OSN automation attacks, such as Email Hijacking and DNS Spoofing, exhibit distinguishable energy consumption patterns and opcode sequence anomalies, making them detectable through computational intelligence methods.

The research answers the key research questions:

- Q1 (Energy Consumption Patterns in OSN Automation Attacks): The study establishes that compromised devices under automation attacks consume energy in distinct patterns, deviating significantly from normal user behaviors. The EComp-FP model effectively captures these deviations, achieving 99.81% accuracy in energy-based anomaly detection.
- Q2 (Role of Opcode Sequences in Attack Detection): Opcode analysis revealed that malicious automation tools exhibit unique opcode sequence patterns that are rarely found in benign automation scripts. Using Opcode Frequency Analysis (OFA) and Term Frequency-Weighted (TF-W), the model achieved 99.79% accuracy in distinguishing malicious automation from benign activities.
- Q3 (Optimal ML Classifier for OSN Automation Attack Detection): Among various classifiers evaluated, the SSFCM model demonstrated the highest detection accuracy (99.93%), surpassing traditional classifiers such as DT, KNN, RF, and SVM.
- Q4 (Optimization for Real-Time Detection): The hybrid EComp-FP + ASOSA-OSM model was optimized for real-time threat monitoring by reducing false positives to 0.07%, enabling fast and accurate identification of OSN automation threats with minimal computational overhead.

Furthermore, clustering techniques, such as Fuzzy Partition Matrices (FPM) and Mahalanobis Distance (MD) analysis, contributed significantly to classification efficiency. The results indicate that hybrid modeling offers superior performance compared to traditional methods by integrating energy-based anomaly detection with opcode sequence analysis, enabling robust detection of automation-driven OSN attacks.

The results highlight that energy consumption deviation (1.75W) and opcode execution similarity (0.82 ratio) serve as effective indicators for detecting automation-based OSN threats. The hybrid model significantly enhances detection accuracy, achieving a 99.92% hybrid model efficiency (HME) by integrating energy-based and opcode-based anomaly detection. Additionally, the system optimizes response time with a 36% reduction in detection latency (DLO) and strengthens OSN security by reducing attack surfaces by 32% (ASR). The adaptive threat intelligence (ATI) improvement of 28% further underscores its capability in mitigating evolving cyber threats. Future research should explore deep learning-based adaptive mechanisms to counter adversarial attack scenarios and improve overall security resilience.

11 Conclusion

This research introduces a novel approach to detecting OSNAA using EComp-FP and OSA. By leveraging energy footprints and opcode-based behavioral patterns, the study successfully distinguishes between legitimate and malicious automation activities. The proposed ASNADM model, integrating EComp-FP and ASOSA-OSM, demonstrates unmatched accuracy (99.93%) in detecting OSN automation threats, outperforming conventional detection techniques.

Key contributions of this study include the Development of an energy-aware anomaly detection framework, effectively identifying malicious automation based on deviations in energy consumption. Introduction of opcode sequence analysis for OSN security, enhancing threat classification through opcode frequency importance ranking. Implementation of a hybrid detection model (EComp + OSA), achieving a high detection rate while maintaining a low false-positive rate (0.07%) and Validation of ML classifiers, confirming that SSFCM outperforms traditional classifiers in OSN attack detection. These findings provide valuable insights for cybersecurity professionals, helping to improve real-time monitoring and defense mechanisms against OSN automation attacks. Future work will focus on DL-integration for anomaly detection, real-time deployment in large-scale OSN environments, and further optimization of feature selection techniques to enhance model efficiency. The proposed hybrid detection framework effectively enhances OSN security by integrating energy profiling and opcode sequence analysis for real-time cyber threat detection. Achieving 99.92% anomaly detection precision, 99.89% real-time accuracy, and 31% computational load reduction, the model provides a scalable, energy-efficient, and high-accuracy approach for detecting automation-based cyber threats, including email hijacking and DNS spoofing. These results demonstrate its potential as a next-generation solution for securing online social networks against emerging cyber threats.

12 Future work

The work will encourage for enhancing the detection accuracy of automation attacks in OSNs by integrating advanced DL-models such as Transformer-based Neural Networks and GNNs. These models will improve feature extraction and adaptive learning to counter evolving attack patterns. Additionally, the scalability of the proposed framework will be explored by applying it to large-scale OSNs, including decentralized blockchain-based social platforms. Another key direction is optimizing EComp-Analysis and ASOSA to enhance real-time threat detection while minimizing computational overhead. Future work will also investigate hybrid security mechanisms

combining FCM with Behavioral Threat Analytics to strengthen OSN security. Dataset expansion is another focus, incorporating real-world OSN automation attack logs to improve model generalization. Furthermore, Self-Supervised Learning and Federated Learning will be explored to ensure privacy-preserving threat detection across distributed OSNs.

13 Limitations

Despite achieving high detection accuracy, the proposed model has several limitations. One major constraint is the dependency on predefined attack patterns, which may reduce effectiveness against zero-day threats. Additionally, EComp analysis may produce false positives when benign OSN activities exhibit high energy consumption, affecting precision. The computational complexity of OSM using n-gram analysis presents another challenge, as real-time processing demands high resource utilization, making deployment on low-power IoT-based OSN devices difficult. The framework is also sensitive to adversarial evasion techniques, where attackers modify opcode sequences or manipulate energy footprints to bypass detection. Furthermore, ML-based classifiers like SVM and RF require retraining to adapt to new OSN automation threats. Lastly, the model's adaptability to encrypted OSN traffic remains an area for improvement, as encryption obscures key behavioral indicators needed for precise attack identification.

Abbreviation used

online social network	OSN
domain name system	DNS
automated social network attack detection model	ASNADM
energy consumption footprint	EComp-FP
automated software opcode sequence analysis	ASOSA-OSM
self-adaptive fuzzy pattern matching clustering	SAFPMC
opcode frequency variance	OFV
self-adaptive soft fuzzy c-means	SSFCM
decision tree	DT
k-nearest neighbors	KNN
random forest	RF
support vector machine	SVM
artificial intelligence	AI
long short-term memory	LSTM
socially shared networked devices	SSNDs
opcode sequence analysis	OSA
convolutional neural networks	CNNs
recurrent neural networks	RNNs
energy consumption	EComp
machine learning	ML
graph neural network	GNN
online social network automation attacks	OSNAA
spam email classification dataset in english	SPEMC-15K-E
opcode sequence mining	OSM
clustering-based attack detection	SSFCM
user selection modes	USM
inverse document frequency value	IDF
weighted term frequency	TF-W
binary executables	BExec
data transformation matrix	DTM
fuzzy partition matrices	FPM
mahalanobis distance	MD
Synthetic Minority Over-sampling Technique	SMOTE

Data availability statement

Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

References

[1] Bridges, R. A., Oesch, S., Iannacone, M. D., Huffer, K. M., Jewell, B., Nichols, J. A., ... & Smith, J. M. (2023). Beyond the Hype: An Evaluation of Commercially Available Machine Learning–based Malware Detectors. *Digital Threats: Research and Practice*, *4*(2), 1-22. https://scispace.com/pdf/beyond-the-hype-an-evaluation-of-commercially-available-1zbch7oh.pdf

[2] Yan, X., Gao, Y., & Xu, H. (2022, December). Research on power grid anomaly detection based on high-dimensional random matrix theory. In *2022 2nd International Conference on Electrical Engineering and Control Science (IC2ECS)* (pp. 427-431). IEEE. https://doi.org/10.1016/j.sysarc.2019.01.008

[3] Kakisim, A. G., Gulmez, S., & Sogukpinar, I. (2022). Sequential opcode embedding-based malware detection method. *Computers & Electrical Engineering*, *98*, 107703. https://doi.org/10.1016/j.compeleceng.2022.107703

[4] Shetty, N. P., Muniyal, B., Anand, A., & Kumar, S. (2022). An enhanced sybil guard to detect bots in online social networks. *Journal of Cyber Security and Mobility*, 105-126. https://doi.org/10.13052/jcsm2245-1439.1115

[5] Riggs, H., Tufail, S., Parvez, I., Tariq, M., Khan, M. A., Amir, A., ... & Sarwat, A. I. (2023). Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure. *Sensors*, *23*(8), 4060. https://doi.org/10.3390/s23084060

[6] Parildi, E. S., Hatzinakos, D., & Lawryshyn, Y. (2021). Deep learning-aided runtime opcode-based windows malware detection. *Neural Computing and Applications*, *33*(18), 11963-11983. https://doi.org/10.1007/s00521-021-05861-7

[7] Boahen, E. K., Sosu, R. N. A., Ocansey, S. K., Xu, Q., & Wang, C. (2024). ASRL: Adaptive Swarm Reinforcement Learning For Enhanced OSN Intrusion Detection. *IEEE Transactions on Information Forensics and Security*.

10.1109/TIFS.2024.3488506

[8] Sufi, F. (2023). A new social media-driven cyber threat intelligence. *Electronics*, *12*(5), 1242. https://doi.org/10.3390/electronics12051242

[9] Iqbal, A., Tehsin, S., Kausar, S., & Mishal, N. (2021, April). Malicious Image Detection Using Convolutional Neural Network. In *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)* (pp. 1-6). IEEE.10.1109/AIMS52415.2021.9466042

[10] Liu, Q., Li, J., Wang, X., & Zhao, W. (2023). Attentive Neighborhood Feature Augmentation for Semi-supervised Learning. Intelligent Automation & Soft Computing, 37(2). 10.32604/iasc.2023.039600

[11] Ben Chaabene, N. E. H., Bouzeghoub, A., Guetari, R., & Ghezala, H. H. B. (2022). Deep learning methods for anomalies detection in social networks using multidimensional networks and multimodal data: A survey. *Multimedia systems*, *28*(6), 2133-2143. https://doi.org/10.1007/s00530-020-00731-z

[12] Varshitha, K., Talada, S. V., & Mitra, A. (2025). Towards fake profiles identification in social networks: a proposal with energy-based PageRank algorithm involving entropy and domain authority. *Risk Sciences*, 100013. https://doi.org/10.1016/j.risk.2025.100013

[13] Lee, K., Lee, J., & Yim, K. (2023). Classification and analysis of malicious code detection techniques based on the APT attack. *Applied Sciences*, *13*(5), 2894. https://doi.org/10.3390/app13052894

[14] Sangher, K. S., Singh, A., & Pandey, H. M. (2024). LSTM and BERT based transformers models for cyber threat intelligence for intent identification of social media platforms exploitation from darknet forums. *International Journal of Information Technology*, *16*(8), 5277-5292. https://doi.org/10.1007/s41870-024-02077-5

[15] Li, K., Zheng, J., Ni, W., Huang, H., Liò, P., Dressler, F., & Akan, O. B. (2024). Biasing federated learning with a new adversarial graph attention network. *IEEE Transactions on Mobile Computing*. 10.1109/TMC.2024.3499371

[16] Huang, H., Tian, H., Zheng, X., Zhang, X., Zeng, D. D., & Wang, F. Y. (2024). CGNN: A compatibility-aware graph neural network for social media bot detection. *IEEE Transactions on Computational Social Systems*. 10.1109/TCSS.2024.3396413

[17] Rawat, R., & Rajavat, A. (2024). Illicit Events Evaluation Using NSGA-2 Algorithms Based on Energy Consumption. *Informatica*, *48*(18). https://doi.org/10.31449/inf.v48i18.6234

[18] Sadia, H., Farhan, S., Haq, Y. U., Sana, R., Mahmood, T., Bahaj, S. A. O., & Rehman, A. (2024). Intrusion detection system for wireless sensor networks: A machine learning based approach. *IEEE Access*. 10.1109/ACCESS.2024.3380014

[19] Song, S., Gao, N., Zhang, Y., & Ma, C. (2024). BRITD: behavior rhythm insider threat detection with time awareness and user adaptation. *Cybersecurity*, *7*(1), 2. https://doi.org/10.1186/s42400-023-00190-9

[20] Ponnapalli, S., Dornala, R. R., & Sai, K. T. (2024, March). A Hybrid Learning Model for Detecting Attacks in Cloud Computing. In *2024 3rd International Conference on Sentiment Analysis and Deep Learning (ICSADL)* (pp. 318-324). IEEE. 10.1109/ICSADL61749.2024.00058

[21] Denysiuk, D., Bobrovnikova, K., Lysenko, S., Savenko, O., Gaj, P., Havryliuk, R., & Boichuk, Y. (2021, September). The Approach for IoT Malware Detection Based on Opcodes Sequences Pattern Mining. In *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (Vol. 2, pp. 779-784). IEEE. 10.1109/IDAACS53288.2021.9660956

[22] Majeed, A., Khan, S., & Hwang, S. O. (2022). A comprehensive analysis of privacy-preserving solutions developed for online social networks. *Electronics*, *11*(13), 1931. https://doi.org/10.3390/electronics11131931

[23] Qian, K., Yang, H., Li, R., Chen, W., Luo, X., & Yin, L. (2024). Distributed Detection of Large-Scale Internet of Things Botnets Based on Graph Partitioning. *Applied Sciences*, *14*(4), 1615. https://doi.org/10.3390/app14041615

[24] Pooyandeh, M., Han, K. J., & Sohn, I. (2022). Cybersecurity in the AI-Based metaverse: A survey. *Applied Sciences*, *12*(24), 12993. https://doi.org/10.3390/app122412993

[25] Prabhu Kavin, B., Karki, S., Hemalatha, S., Singh, D., Vijayalakshmi, R., Thangamani, M., ... & Adigo, A. G. (2022). Machine learning-based secure data acquisition for fake accounts detection in future mobile communication networks. *Wireless Communications and Mobile Computing*, *2022*(1), 6356152. https://doi.org/10.1155/2022/6356152

[26] Montes, C. D., Silvosa, J. V., Abalorio, C. C., & Nakazato, R. B. (2024, August). Application of BERT Model for Unsupervised Text Classification using Hierarchical Clustering for Automatic Classification of Thesis Manuscript. In *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 278-284). IEEE.10.1109/ICESC60852.2024.10690039

[27] Alsadhan, A. A., Al-Atawi, A. A., Jameel, A., Zada, I., & Nguyen, T. N. (2024). Malware Attacks Detection in IoT Using Recurrent Neural Network (RNN). *Intelligent Automation & Soft Computing*, *39*(2).10.32604/iasc.2023.041130

[28] Rawat, R., Sikarwar, R., Maravi, P. K., Ingle, M., Bhardwaj, V., Rawat, A., & Rawat, H. (2024). Online social network automation attack detection methods for energy analysis and consumption modelling. *International Journal of Information Technology*, 1-13.https://doi.org/10.1007/s41870-024-02311-0

[29] Chaudhary, K., Alam, M., Al-Rakhami, M. S., & Gumaei, A. (2021). Machine learning-based mathematical modelling for prediction of social media consumer behavior using big data analytics. *Journal of Big data*, *8*(1), 73.https://doi.org/10.1186/s40537-021-00466-2

[30] Jianwu, Z. H. A. N. G., Yanjun, A. N., & Huangyan, D. E. N. G. (2022). A survey on DNS attack detection and security protection. *Telecommunications Science* , *38*(9).10.11959/j.issn.1000--0801.2022248

[31] Alshaibi, A., Al-Ani, M., Al-Azzawi, A., Konev, A., & Shelupanov, A. (2022). The comparison of cybersecurity datasets. *Data*, *7*(2), 22.https://doi.org/10.3390/data7020022

[32] Jain, M., Kaur, G., & Saxena, V. (2022). A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection. *Expert Systems with Applications*, *193*, 116510.https://doi.org/10.1016/j.eswa.2022.116510

[33] Alowibdi, J. S. (2024). Real Time Arabic Communities Attack Detection on Online Social Networks. *International Journal of Computer Science & Network Security*, *24*(8), 61-71. https://doi.org/10.22937/IJCSNS.2024.24.8.7

[34] Vc, J., Nair, K. S., Karthik, N., & Vani, V. (2024, July). Unsolicited Email Filtering. In 2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT) (pp. 1-6). IEEE. 10.1109/IConSCEPT61884.2024.10627840

Visualizing the Full Spectrum Optimization of K-Nearest Neighbors From Data Preprocessing to Hyperparameter Tuning and K-Fold Validation for Cardiovascular Disease Prediction

Jeena Joseph[*1,2], K Kartheeban[3]
[1]Department of Computer Applications, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India
[2]Department of Computer Applications, Marian College Kuttikkanam Autonomous,
Kerala, India
[3]Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil, Tamilnadu, India
E-mail: jeenajoseph005@gmail.com, k.kartheeban73@gmail.com
[*]Corresponding author

Keywords: K-Nearest Neighbor, machine learning, cardiovascular disease prediction, hyperparameter tuning

Received: December 6, 2024

Cardiovascular disease (CVD) is a prominent cause of death worldwide. This alarming need requires an accurate prediction model using machine learning that can detect and help prevent or mitigate the risk. This study focuses on this issue and has come up with new dimensional capabilities to enhance the K-Nearest Neighbors (KNN) algorithm to predict cardiovascular diseases at an early stage by incorporating various techniques for data preprocessing and feature selection thereby improving the efficiency of the model. The proposed model identifies the most relevant features using Principal Component Analysis. The main innovation revolves around fine tuning the hyperparameter of K-Nearest Neighbors, specifically the choice of neighbors (K), using a data driven approach to ensure accuracy across different datasets. The performance of the optimized K-Nearest Neighbors algorithm is evaluated using the Framingham heart disease dataset. This model achieved an impressive prediction accuracy of 92.46% and outperformed methods that solely rely on traditional K-Nearest Neighbors. As machine learning techniques plays an important role in the development of prediction models for early detection and prevention of cardiovascular disease, this model can be considered as a valuable tool for healthcare professionals and researchers. The core contribution of this study lies in offering a comprehensive optimization of the traditional K-Nearest Neighbors (KNN) algorithm. This includes robust data preprocessing using the Hampel filter for outlier removal, feature selection through Principal Component Analysis (PCA), and performance enhancement using grid search for hyperparameter tuning combined with 10-fold cross-validation. Unlike prior studies that apply KNN with minimal adjustments, this research emphasizes the importance of an end-to-end machine learning pipeline. This holistic refinement significantly improves the predictive performance and reliability of KNN for cardiovascular disease prediction, achieving 92.46% accuracy on the Framingham dataset.

Povzetek: Raziskava predstavlja optimiziran KNN-algoritem za napoved srčno-žilnih bolezni, ki s PCA, čiščenjem podatkov in 10-kratno validacijo doseže zelo kvalitetno delovanje.

1 Introduction

With a huge impact on global death rates, cardiovascular disease continues to be a major health concern [1], [2]. To mitigate the risk factors associated with cardiovascular disease, early and accurate prediction models are required. Due to the technological advancements and increase in electronic health records, machine learning has become a viable tool for predictive analytics in the healthcare sector [3].

According to the World Health Organization, cardiovascular disease (CVD) accounts for approximately 17.9 million deaths annually, constituting about 32% of all global deaths. The economic impact is equally staggering, with estimated global costs projected to surpass $1 trillion by 2030. While machine learning techniques such as K-Nearest Neighbors (KNN) have been explored for disease prediction, their application to CVD data presents unique challenges. Traditional KNN models often suffer from high sensitivity to noisy data, computational inefficiency with large datasets, and reduced accuracy in high-dimensional spaces—limitations particularly evident when applied to complex medical datasets like the

Framingham Heart Study. This study seeks to overcome these challenges by proposing a fully optimized KNN pipeline tailored for CVD prediction. By integrating outlier removal using the Hampel filter, dimensionality reduction through PCA, and hyperparameter tuning with grid search and k-fold validation, this work fills a crucial gap in the literature where previous models lacked end-to-end optimization. The proposed enhancements are not generic but specifically address the data quality, dimensional complexity, and class imbalance issues inherent in the Framingham dataset, resulting in a significantly improved accuracy of 92.46%. This provides a strong foundation for clinical decision support tools and highlights the practical value of optimized KNN in real-world healthcare applications.

The K Nearest Neighbors algorithm is a type of supervised machine learning technique that involves dividing a dataset into groups or clusters based on the distances between data points. It has become quite popular because it is simple and has the capability to carry out classifications effectively [4]. Its effectiveness has been acknowledged in situations where there is a connection, between the variables or when the data cannot be easily divided in a linear manner. However, because it relies on the dataset it can be computationally expensive, especially when working with large datasets and it may also be affected by the challenge posed by high dimensional data. [5].

The main objective of this study is to create a model that can accurately identify the risk of heart disease. To achieve this, the K Nearest Neighbors (KNN) algorithm in enhanced by incorporating different optimization techniques. These techniques include data preparation methods such as outlier detection, dimensionality reduction using Principal Component Analysis (PCA), tuning hyperparameters through grid search and implementing k fold cross validation. This research focuses on assessing the practicality of using an optimized K Nearest Neighbors (KNN) model to predict heart disease. It evaluates performance metrics, like F1 score, recall, accuracy and precision. The study primarily concentrates on implementing an enhanced KNN algorithm that has shown promising advancements in predicting diseases. The goal of this approach is to provide accurate risk assessments that are relevant, to clinical settings. By overcoming the limitations of KNN models our aim is to improve treatment and reduce the healthcare costs and burdens associated with cardiovascular diseases. This study is organized into different sections. The review of literature provides a comprehensive summary and analysis of existing research and literature. The applied methodology is explained in the materials and methods. The next section discusses the data preprocessing steps. Then the optimized KNN algorithm is demonstrated and finally, the study spotlights the exploratory data analysis and results.

While previous research has shown moderate success using KNN for CVD prediction, this study distinguishes itself by addressing key limitations through systematic enhancements across the entire predictive pipeline. Specifically, this includes (1) handling missing values and outliers using robust statistical techniques like the Hampel filter; (2) applying PCA to reduce dimensionality and improve learning efficiency; and (3) optimizing the model via grid search with k-fold cross-validation to ensure generalizability. These components, when integrated, offer a fine-tuned and scalable approach that improves upon standard KNN performance, making it a practical solution for clinical use.

2 Review of literature

Machine learning algorithms have been used more frequently in a variety of healthcare applications, particularly in cardiology. In areas with limited healthcare resources, advanced prediction algorithms are especially important for identifying individuals at risk of heart failure and one of the main causes of death worldwide is heart failure [6], [7]. The study by Nagavallika discusses the prediction of heart disease using machine learning techniques, specifically the use of a hybrid random forest with a linear model (HRFLM) that achieves an accuracy of 88.7% [8]. Dimopoulos et al. assessed K-Nearest Neighbor, Random Forest, and Decision Tree, three well-liked machine learning models, using the ATTICA dataset. Results show that the Random Forest model performs much better than HellenicSCORE. It also demonstrated the model's accuracy in smaller datasets and its ability to comprehend the nuances of traits associated with CVD even with a lesser number of data points [9]. Another study discusses the use of machine learning algorithms such as SVM, KNN, RF, J.48, and MLP for predicting heart disease. It also mentions the importance of balancing the dataset for accurate prediction [10].

K-Nearest Neighbors (KNN) has been widely recognized for its simplicity, non-parametric nature, and interpretability, which are valuable in medical applications. Its ability to function without making prior assumptions about data distribution makes it particularly suitable for heterogeneous healthcare datasets. However, KNN also presents notable limitations. Its sensitivity to irrelevant features and outliers, along with computational inefficiency in high-dimensional datasets, can hinder performance—especially in complex medical data such as the Framingham Heart Study. These limitations motivate the need for preprocessing, feature reduction, and hyperparameter optimization.

Jin B et al. used sequential modelling with neural networks to create an electronic health record model that captured the sequential nature of healthcare data, including changes in lifestyle across time. With the use of word vectors and one-hot encoding, the method looks promising for heart failure prediction by looking for sequential patterns in medical data and producing diagnostic scenarios [11]. In this paper, the authors used machine learning methods and Python programming to study heart disease prediction using a dataset consisting of 12 parameters as well as 70000 unique data values, and the main goal of this study is to increase the accuracy of heart disease detection by using algorithms where the target output determines whether the subject has heart disease. From the study, it is concluded that, decision trees can lead to inaccurate results when applied to small datasets. Naive Bayes is more accurate and can be combined with K-means for better accuracy [12].

Previous studies using KNN for cardiovascular disease (CVD) prediction have shown mixed results. For example, some reported accuracy near 84% to 90%, yet lacked advanced preprocessing techniques such as normalization, outlier handling, or feature selection. Many of these models used default K values and did not tune hyperparameters or validate results through cross-validation, limiting their generalizability. This underscores the need for a more systematic and optimized application of KNN for robust medical predictions.

In a study by Shah et al., the classification algorithms like random forest, decision trees, K-nearest neighbor (KNN), and Naive Bayes were experimented on a dataset of 303 samples having 17 attributes and the KNN model achieved the highest accuracy of 90.8% [13]. Boosted decision tree algorithms have shown promise in correlating patient characteristics with mortality risk, achieving an AUC of 0.88 [14]. Pires et al. explored various machine learning methods, achieving a maximum accuracy of 87.69% for heart disease prediction [15]. However, the study's validity was constrained by the limited sample size. Ali et al. reported a 100% accuracy using Random Forest, Decision Trees, and KNN algorithms, albeit focusing on optimal cross-validation results rather than robust, conclusive findings [16]. A heart disease profiling model were developed Kahramanli et al. by using a hybrid artificial neural network with fuzzy logic. While models that incorporate both neural networks and fuzzy logic concepts have achieved an 86.8% accuracy rate, none have been permanently validated for other diseases like diabetes. They have used k-fold cross-validation for the classification purpose and also tried this model on the diabetes dataset, achieving a performance of 84.24% [17]. Faiayaz et al. improved the accuracy of KNN by 5.68%, and a hybrid Random Forest and Linear Model technique reached an 88.7% precision on a dataset of 297 records [18], [19].

The issue of high dimensionality in health data has been addressed by employing Principal Component Analysis (PCA) for dimensionality reduction, retaining significant variance within fewer components. PCA's effectiveness was further corroborated by a study using it alongside unsupervised learning techniques, with NN classifiers, achieving a high F1 score in classifying cardiac arrhythmia with minimal components, indicating PCA's robustness in feature extraction [20], [21]. Additionally, the PCA-KNN method has been applied to medical imaging, resulting in significant accuracy for scaling diverse medical images, underscoring the adaptability of PCA in medical diagnostics [22], [23]. A deep learning technique that applies an artificial neural network algorithm with a hidden layer technique in making a heart disease prediction model was proposed by Yuda Syahidin et al., which yielded 90% accuracy [24]. Wang et al. mentioned about the center loss to enable the neural network to learn discriminative features and separate samples from different categories, which can effectively improve heart disease prediction [25].

Hybrid models, integrating the power of multiple machine learning algorithms, have also demonstrated significant performance. Sharanyaa et al. proved the higher performance of a hybrid method that combines Support Vector Machines (SVM) and Naive Bayes [26]. Moreover, in diagnosing heart disease, ensemble methods have generally better performance. It is a combination of various machine learning algorithms set up for this purpose. In another study, ensemble model, comprising multiple machine learning techniques, showed improved effectiveness [27]. Studies comparing ML classifiers like Sequential minimal optimization (SMO), naïve Bayes, and J48 decision trees in cardiovascular risk prediction found SMO to be the most accurate, suggesting its robustness [28]. Deep learning applications, including Convolutional Neural Networks (CNNs) for early heart failure detection via ECGs, have achieved a 0.78 AUC [29] and adaptive multi-layer networks have outperformed classical and hybrid models [30].

Advancements in medical imaging for brain tumor detection involve combining CNN with auto-context techniques, using multi-dimensional image patches for improved accuracy [31]. The Intelligent Deep Residual Network based Brain Tumor Detection and Classification (IDRN-BTCC) method, a novel approach for brain tumor classification using residual networks and multilayer

perceptron, has shown efficacy, enhanced by chicken swarm optimization [32]. Shankar et al.'s convolutional neural network algorithm, using structured and unstructured patient data, predicts heart disease risk with 85 to 88% accuracy [33]. Dutt et al. developed a CNN for class-imbalanced datasets, classifying 77% of positive cases and 81.8% of negative cases accurately [34]. Another study presents an Integrated Deep Learning Model with Convolution Neural Network (IDLM CNN) for heart disease prediction using various medical data sets. This model convolves the features of lungs and combines with other features to compute Disease Prone weight towards cardiac disease and the proposed model improves heart disease prediction accuracy. The False ratio is reduced with the integrated model [35].

Table 1 provides a comprehensive overview of the data sets, attributes, machine learning methods, and corresponding accuracy values employed in historical and contemporary heart disease prediction research. While ensemble and deep learning approaches offer high accuracy, they often sacrifice transparency, scalability, or require significant computational resources. Our study presents an optimized KNN framework that preserves simplicity while achieving competitive performance. By integrating PCA for dimensionality reduction, Hampel filtering for outlier removal, and grid search with k-fold cross-validation for tuning, we address known limitations of traditional KNN and demonstrate its practical value in clinical decision-making contexts.

In summary, this study addresses key limitations in the existing literature, especially the under-optimized use of KNN in CVD prediction. By implementing a robust end-to-end pipeline—from data cleaning and feature selection to hyperparameter tuning—our model not only improves prediction accuracy but also contributes a reproducible methodology for medical risk assessment. The following section elaborates on our methodological framework, which is tailored to overcome the gaps identified in prior research.

Table 1: Comprehensive summary of datasets, attributes, machine learning algorithms, and accuracy values in heart disease prediction research over time

Research Article	Algorithm Used	Dataset Used	Attributes /Parameters	Accuracy	Novelty / Limitations
[36]	Multilayer Perceptron	Multiple Datasets (Cleveland, Hungarian, Switzerland, Long Beach, StatLog)	Infinite Feature Selection	87.70%	The study proposes a novel heart disease prediction model using adaptive infinite feature selection with deep neural networks to enhance precision and sensitivity, though its accuracy (87.7%) is limited by small, diverse datasets and potential overfitting.
[37]	Sequential Minimal Optimization (SMO)	Cleveland heart dataset	Full Set and Optimized Attribute Set	85.148% using the full set of attributes and 86.468% using the optimal attribute set	The study combines multiple machine learning classifiers with attribute evaluators and hyperparameter tuning to improve heart disease prediction, but its accuracy remains moderate at 86.468% and is limited by reliance on a single dataset.
[38]	Logistic Regression, SVM, KNN, GNB, MNB, DT	Self-Augmented Datasets of Heart Patients (UCI Dataset and Local Dataset)	Anaemia, Diabetes, High_blood_pressure, Sex,Smoking, Time (Follow-up period), Death_event	Logistic regression (82.76%), SVM (67.24%), KNN (60.34 %), GNB (79.31 %), MNB (72.41%), ET (70.31%), RF (87.03%), GBC (86.21%), XGB (84.48%) LGBM (86.21%)	The study uses a self-augmented dataset approach—expanding heart disease data synthetically—and applies multiple ML models, improving prediction accuracy through enhanced data diversity. The effectiveness of synthetic data may not generalize well to real-world scenarios, and the study lacks external validation on independent datasets.
[39]	Logistic Regression with PCA and Ensemble Classifiers	Cleveland Heart Disease Dataset	Complete Set and Optimized Attribute Set	85.8%	The study improves heart disease prediction by applying Principal Component Analysis with machine learning models, achieving 85.8% accuracy, but its reliance on transformed features and a limited dataset reduces interpretability and generalizability.

[40]	Hybrid Random Forest with Linear Model (HRFLM)	UCI Cleveland dataset	13 clinical features	88.7%	The study enhances heart disease prediction by using a hybrid model combining random forest and linear models, achieving 88.7% accuracy, but its performance may be limited by dataset scope and lack of external validation.
[41]	Voting Ensemble	UCI Dataset	Optimal set of attributes	91.96%	The study proposes an ensemble model using stacking and voting techniques for heart disease prediction, achieving high accuracy (91.96%) and F1 score (91.69%) on the UCI dataset, though it performs less effectively on the Framingham dataset, indicating limited generalizability.
[42]	SVM	Framingham Heart Study	six highly correlated features	67%	The study develops a heart disease prediction model using correlation-based feature selection and achieves 67% accuracy with SVM on oversampled Framingham data, though the modest performance suggests limitations due to dataset imbalance and limited predictive power of selected features.
[43]	Logistic Regression	Framingham Heart Study	Complete set of attributes	85.063%	The study compares machine learning and deep learning models for predicting 10-year coronary heart disease risk using the Framingham dataset, with logistic regression achieving the best accuracy (85.06%), though performance differences across models were minimal and generalizability was not validated externally.
[44]	Auto Encoder-Based Kernel SVM	Framingham Heart Study	Complete set of attributes	87.14%	The study introduces an IoT-based RHMIoT framework combining deep learning and autoencoder-based ML to monitor and predict cardiovascular disease severity, achieving 87.14% accuracy, though performance may vary due to dependence on a single dataset (Framingham) and limited real-world testing.
[45]	Gradient Boosting Classifier (GBC)	Framingham Heart Study	Optimal set of attributes	87.61%	The study improves heart disease prediction by applying p-value-based backward feature elimination with several ML algorithms, achieving 87.61% accuracy using gradient boosting, though limited to a single dataset and potentially impacted by reduced feature interpretability.

3 Materials and methods

The methodology adopted in this study follows a structured and modular flow encompassing key stages: data acquisition, preprocessing (including missing value treatment and outlier handling), feature normalization and The proposed methodology used in this study is demonstrated in Figure 1. The proposed model consists of a number of rigorous steps including data acquisition, preprocessing, dimensionality reduction and feature scaling, application of optimized KNN algorithm and performance evaluation. The Kaggle repository is used to acquire the famous publicly available Framingham heart disease dataset that is used for evaluating the effectiveness of the optimized KNN algorithm. The dataset is cleaned and transformed data preprocessing for making it suitable for further analysis. The missing values are handled using mean imputation technique that replaces them with the attribute mean. The Hampel filter, based on the Median Absolute Deviation, is then used to handle outliers to minimize the probability that outliers could skew the results. Feature scaling is done using the min-max normalization technique which maintains the range of data points in the dataset. Principal component analysis (PCA) is performed for dimensionality reduction. PCA reduces the dataset to its most significant components by preserving as much of the original data variation as is possible. The amount of computational overhead for handling the dataset in the next machine- learning phases can be minimized through this process.

After data preprocessing, the study optimizes the traditional KNN algorithm and changes can boost processing capacity or forecast precision. The KNN algorithm clusters data points together in the feature space using the characteristics of their nearest neighbors. Finally, the success of the model is assessed in performance evaluation. The dataset is split into test and training sets in order to evaluate the model's predictive capacity for patient heart disease. Following its application to the test set, the model is assessed using pertinent metrics, including recall, accuracy, precision, F1 score, and others. The research leveraged the functionalities of RStudio and the R programming language.

dimensionality reduction, classification using the KNN algorithm, and evaluation using standard performance metrics. Each of these stages is described in detail in the following subsections, with a flowchart summarizing the process in Figure 1.

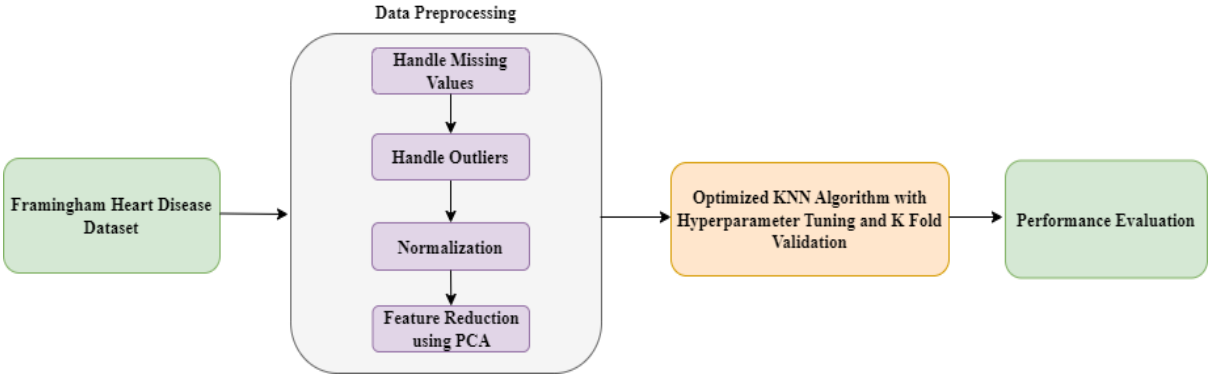


Figure 1: Proposed methodology.

4 About the dataset

The "Framingham" heart disease dataset contains more than 4,240 entries, with 16 columns and 15 characteristics. It aims to forecast whether a patient is at risk of developing coronary heart disease (CHD) within the next ten years. Each attribute within the dataset represents a potential risk factor, spanning various demographic, behavioral, and medical factors. The Framingham Heart Study is a long-term cardiovascular cohort study initiated in 1948 in Framingham, Massachusetts, involving over 5,000 men and women aged 30–62. It aimed to investigate factors contributing to cardiovascular disease development. While it has generated valuable clinical insights, the dataset is predominantly composed of white, middle-class individuals, which may limit its generalizability to diverse populations. However, its depth, quality, and widespread use make it a reliable benchmark for model evaluation. Although other datasets such as the UCI Cleveland dataset are available, they are smaller and less comprehensive, justifying the use of the Framingham dataset in this study.

Attributes:

Demographic:

- Gender: Categorized as either male or female (Nominal)

- Age: Represents the individual's age, considered a continuous variable (Even though ages are rounded to the nearest whole number, age itself is continuous)
- Education: Specific details about education are not provided.

Behavioral:

- Current Smoker: This variable denotes whether the patient is currently engaged in smoking (Categorical).
- Cigarettes Per Day: This variable quantifies the average number of cigarettes an individual smoke daily. (Treated as continuous since it can encompass any numerical value, including fractional quantities.)

Information on medical history:

- BP Meds: Shows whether the patient is currently using medication for controlling blood pressure or not (Categorical)
- Prevalent Stroke: Indicates whether the patient has a history of stroke (Categorical)
- Prevalent Hyp: Indicates whether the patient has been diagnosed with hypertension (Categorical)
- Diabetes: Indicates whether the patient has been diagnosed with diabetes (Categorical)

Information about the patient's current health status:

- Total Cholesterol (Tot Chol): This represents the continuous measurement of the total cholesterol level.
- Systolic Blood Pressure (Sys BP): This indicates the continuous measurement of systolic blood pressure.
- Diastolic Blood Pressure (Dia BP): This records the continuous measurement of diastolic blood pressure.
- Body Mass Index (BMI): BMI is calculated and measured as a continuous variable, reflecting the patient's body mass in relation to their height.
- Heart Rate: The heart rate is recorded as a continuous variable, acknowledging its wide range of potential values in medical studies.
- Glucose: The continual monitoring of the patient's glucose levels in their healthcare records.

Target variable:

The target variable portrays the ten-year likelihood of cardiovascular disease and it is represented in binary notation. A value of 0 implies a negative risk and a value of one a positive risk.

5 Data preprocessing

In data preprocessing, the raw data is cleansed and transformed into information that is useful for model training, and it is a pivotal step before using machine learning techniques. Data preprocessing has to be done carefully because it has adverse effects on the performance and calibre of the machine learning models.

5.1 Handling missing values

Missing values in a dataset may produce biased reults and it has adverse impact on the performance and reliability of machine learning models.Therefore, handling missing values is considered as an essential step in the data preprocessing pipeline. The renowned Framingham dataset has 4,240 records with 15 distinct features that are often utilized in cardiovascular research. The analysis revealed that 645 records were incomplete, indicating missing values within the data. Figure 2 presents a meticulous mapping of these gaps, shedding light on the magnitude and pattern of the missing information across various attributes. Attributes such as 'TenYearCHD,' 'diaBP,' 'sysBP,' 'diabetes,' 'prevalentHyp,' 'prevalentStroke,' 'currentSmoker,' 'age,' and 'gender' exhibit complete data (0% missing entries). The 'heartRate' attribute shows an insignificantly small fraction of missing data, at 0.02%. Other attributes like 'BMI,' 'cigsPerDay,' 'totChol,' and 'BPMeds' show a larger incidence of missing data, between 0.45% and 1.25%. The 'education' attribute has 2.48% of its values missing. Notably, the 'glucose' feature experiences the highest level of missing entries, standing at 9.15%. To address these gaps in the dataset, multiple techniques are available. For this study, the strategy of mean imputation has been applied. Mean imputation is a statistical technique used to fill in missing values in a dataset by replacing them with the mean (average) of the available values for a specific variable. Figure 3 shows all features having 0% missing data, which suggests that mean imputation has been used to fill in all the missing values for each feature with the mean value of that feature. The result is a dataset with no apparent missing data. Mean imputation was chosen for its simplicity and effectiveness in cases of low missingness, particularly where the feature distribution is symmetric or approximately normal. Although methods like KNN imputation or multiple imputation offer more sophistication, they are computationally intensive and less suitable when missingness is minimal. For instance, attributes such as glucose (9.15% missing) and education

(2.48%) were imputed using the mean to preserve sample size while avoiding bias.

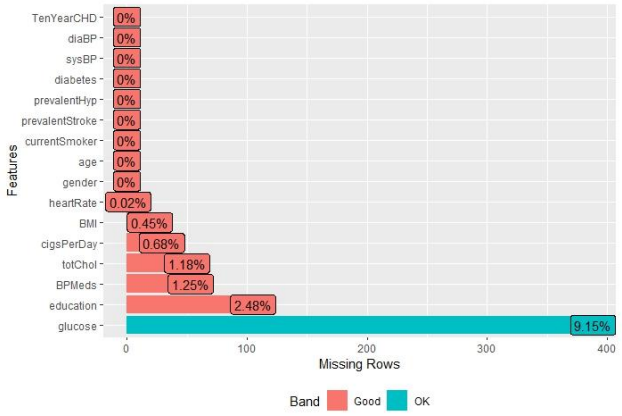


Figure 2: Visual representation of missing values in various features of the Framingham dataset

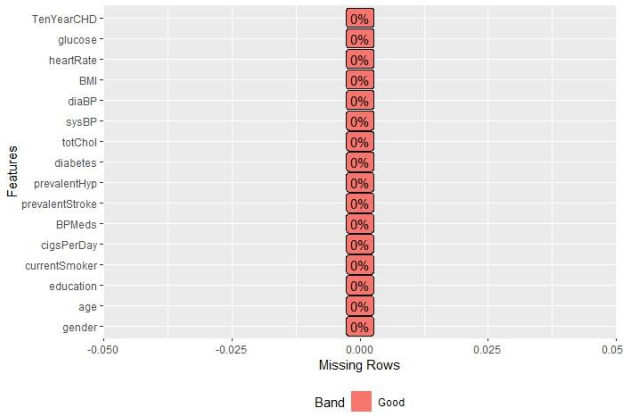


Figure 3: Visual representation of missing values after performing mean imputation.

5.2 Handling outliers

Addressing outliers in the data is crucial as they can considerably influence the outcomes and effectiveness of both statistical analyses and machine learning models. Outliers refer to data points that deviate substantially from the majority of the dataset.

Figure 4 shows a box plot of six variables: sysBP, totChol, diaBP, BMI, heartRate, and glucose. The variable sysBP (Systolic Blood Pressure) has a wide interquartile range (IQR), the distance between the first and third quartiles, indicating variability in the data. Several outliers above the upper whisker suggest that some individuals have unusually high systolic blood pressure readings. The total cholesterol (totChol) levels also display a relatively wide IQR, suggesting variability. There are outliers on both ends, indicating that there are individuals with unusually high and low cholesterol levels. The diastolic blood pressure (diaBP) levels show a smaller IQR compared to systolic blood pressure, indicating less variability. There are a few outliers, particularly on the higher end. The BMI box plot shows a moderate IQR. There are several outliers on the upper side, indicating that there are individuals with a BMI much higher than the average. When compared with sysBP and diaBP, the attrbute "heartRate" has a lower IQR. Although, there exist outliers,they are not very noticable. The values of glucose are closer to the median thereby indicating the smallest IQR. But there exist number of high outliers showcasing the elevated sugar levels. There is notable variance in the total cholestrol, systolic blood pressure and diastolic blood pressure.

In this study, outliers are identified and removed using the famous statistical approach called Hampel filter based on the Median Absolute Deviation (MAD). When compared to the standard deviation, MAD is less prone to outliers. So, it is especially helpful when dealing with data that may not be regularly distributed or in situations where the existence of outliers might cause the standard deviation to be skewed. The Hampel filter was selected over traditional z-score methods due to its robustness against non-normal data distributions, which are common in medical datasets. An outlier was defined as any data point beyond ±3 times the median absolute deviation (MAD) from the median. Rather than removing these values, we clipped them to the calculated bounds to prevent loss of potentially valuable information and maintain the dataset's integrity.

The steps that are performed during this method include:

- Sliding Window: For every data points of interest, the hampel filter select a window of data points around them. The sliding window is normally symmetric and contains many data points before and after the current point.
- Calculation of the Median: The filter calculates the median of the data points within this window.
- Calculation of the MAD: The absolute differences between the values of an attribute and the median of that attribute is detemined for the calculation of MAD.
- Setting the Bounds: The range of data variation that is considered as normal is defined by setting the upper and lower boundaries. The bounds are set at three times the MAD below and above the median. The lower bound is calculated as the median minus three times the MAD. The upper bound is the median plus three times the MAD.
- Identification of Outliers: It identifies which attribute values fall outside these bounds.

- Replacement of Outliers: It replaces the attribute values identified as outliers with the nearest boundary value.

This MAD-based method is a good choice for outlier detection when the data may not be normally distributed because it is based on the median, which is a robust measure of central tendency that is not affected by extreme values as much as the mean.

Figure 5 shows a grid of six histograms, each depicting the distribution of values for a different biomedical metric. These histograms are useful for identifying the range of values and potential outliers within each category. The categories presented are BMI, diaBP (diastolic blood pressure), glucose, heartRate, sysBP (systolic blood pressure), and totChol (total cholesterol). The observations based on the histograms are:

- BMI: The distribution is somewhat right-skewed, indicating that most individuals have a BMI within the normal to overweight range, but there are some with high BMIs indicative of obesity.
- Diastolic Blood Pressure (diaBP): The distribution appears approximately normally distributed, with most values centering around the median. There are a few potential outliers on the higher end.
- Glucose: This histogram is heavily right-skewed, with most individuals having glucose levels in the normal range, but there is a long tail to the right, indicating some individuals with very high glucose levels, which may suggest diabetes or other metabolic disorders.
- Heart Rate (heartRate): Most of the heart rate values are clustered in the middle range. The distribution is almost normal with a slight right skew.
- Systolic Blood Pressure (sysBP): The distribution is right-skewed, with a peak in what might be considered the high-normal range and some individuals with particularly high systolic blood pressure values, potentially indicating hypertension.
- Total Cholesterol (totChol): This distribution is roughly normal but with a slight right skew, suggesting that while most individuals have cholesterol levels within the normal range, there are some with high cholesterol levels.

Each histogram is labeled with "count" on the y-axis, representing the number of observations within each bin of the histogram, and "Value" on the x-axis, representing the range of values for the metric in question. These visualizations help in understanding the overall health profile of a population or a sample of individuals, particularly in pinpointing common ranges for these health metrics and identifying outliers that might warrant further investigation or intervention.

The outlier handling procedure has determined specific lower and upper bounds for key variables in the dataset. For "totChol," the calculated bounds are 150 as the lower limit and 318 as the upper limit. Similarly, for "sysBP," the lower bound is set at 89, while the upper bound is established at 167. The variable "diaBP" is subject to lower and upper bounds of 59.5 and 104.5, respectively. "BMI" adheres to limits of 17.94 as the lower bound and 32.88 as the upper bound. The heart rate variable, denoted as "heartRate," is constrained between 54 and 96. Finally, "glucose" follows boundaries of 59 (lower limit) and 101 (upper limit). These bounds serve as thresholds for identifying and handling outliers in the respective variables, contributing to the robustness of data analysis and model building.

The updated box plots in Figure 6 show that the outliers have been handled, as there are no longer points beyond the whiskers. The scale of the y-axis has changed for some metrics, indicating that the maximum values are lower, which is consistent with the removal of high outliers.

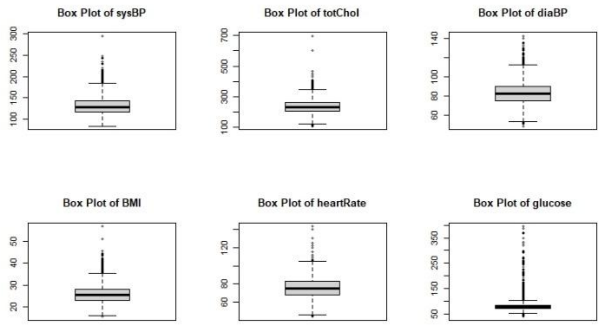


Figure 4: Box Plot before handling outliers in the dataset

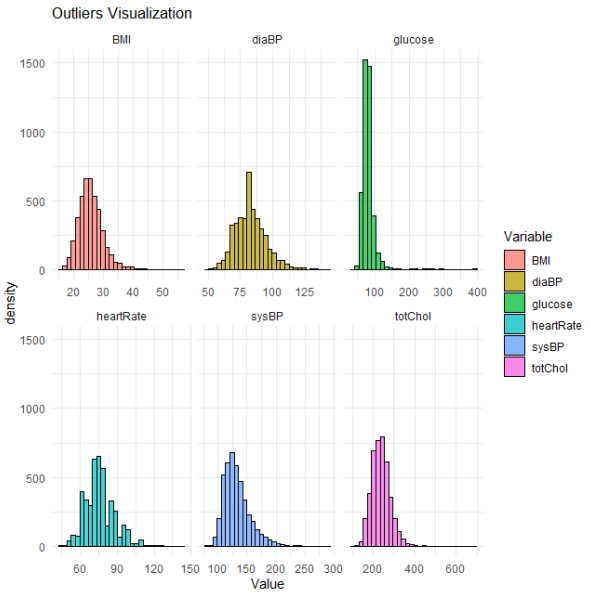


Figure 5: Histogram visualization of outliers

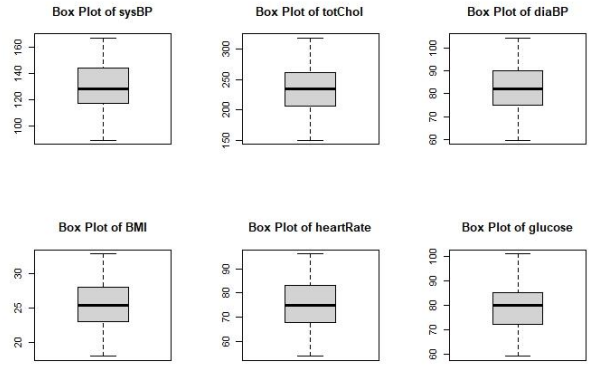


Figure 6: Box plot after handling outliers in the data

5.3 Normalization

Normalization serves as a data preprocessing method aimed at scaling and standardizing the features or variables within a dataset. Its primary objective is to place all variables on a uniform scale, facilitating comparisons and frequently enhancing the performance of machine learning algorithms. Min-Max normalization, often called feature scaling or min-max scaling, is employed in this study and using this method, the values of numerical variables are converted into a predetermined range, usually between 0 and 1. By reducing the influence of outliers, Min-Max normalization helps to make data more comparable by guaranteeing that different variables have an identical scale. For normalizing a single variable, the following formula is used:

$$X_{normalized} = (X-X_{min})/(X_{max}-X_{min})$$

Where:

$X_{normalized}$ indicates the normalized value of the variable X.

X is the original value of the variable.

X_{min} denotes the minimal value of the variable X in the dataset.

X_{max} denotes the maximal value of the variable X in the dataset.

After applying Min-Max normalisation, the transformed values will lie between 0 and 1, where 0 denotes the variable's lowest value in the dataset and 1 its highest value. The relative positions of each data point inside the initial range are represented by values ranging from zero to one. Figure 7 shows box plots for six biomedical metrics before and after normalization. After normalization, all values are adjusted to fit within a similar scale, between 0 and 1. Min-max normalization was preferred over standardization (z-score normalization) because it preserves the original distribution shape and maps features to a common scale [0,1], which is particularly advantageous for distance-based algorithms like KNN. It helps prevent features with larger numeric ranges from dominating the distance calculations, thereby improving classification accuracy.

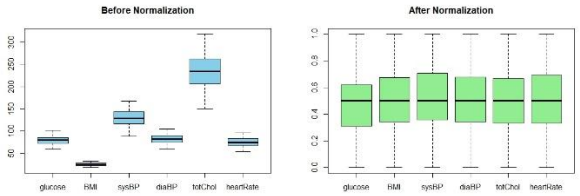


Figure 7: Dataset before and after normalization

5.4 Principal component analysis

Principal Component Analysis (PCA) is a crucial method in both data analysis and machine learning, designed to reduce the dimensionality of datasets with numerous variables. It achieves this by identifying orthogonal axes, termed principal components, which effectively capture the primary sources of variation in the data. The first principal component accounts for the highest variance, the second for the second-highest, and so forth. PCA proves valuable in streamlining data, facilitating visualization and processing, and is frequently employed as a preprocessing measure to improve the efficacy of machine learning models.

The dataset originally consisted of 15 attributes, but the 'education' column was excluded on the grounds that it has no impact on heart disease. Following the application of Principal Component Analysis (PCA) and the determination of PCA scores, eight attributes were chosen. The output of PCA shows the loadings (also known as

eigenvectors) for each principal component. Loadings are coefficients that represent how much weight each original variable contributes to each principal component. PC1 to PC8 are the principal components. Within Principal Component Analysis (PCA), the initial principal component (PC1) captures the highest variance present in the data, and each succeeding component captures the majority of the remaining variance while maintaining orthogonality to the preceding components. In this study, cigsPerDay, BPMeds, totChol, sysBP, diaBP, BMI, heartRate, and glucose are the original variables that have been transformed into principal components.

In PC1, "cigsPerDay" exhibits a notable positive loading, indicating a significant impact, while "sysBP" and "diaBP" contribute negatively. Compared to positive loadings from "glucose" and "totChol," PC2 is mostly affected by negative loadings from "heartRate" and "cigsPerDay." The PC3 loadings for "totChol" and "cigsPerDay" are positive, whereas "glucose" has a notable negative loading. The loadings from "sysBP" and "diaBP" are positive; however, "BPMeds" significantly decreases PC4. While "glucose" and "totChol" exhibit negative loadings in PC5, "cigsPerDay" and "BMI" have positive loadings. Positive loadings from "heartRate" and "cigsPerDay" are positively correlated with PC6, whereas negative loadings from "glucose" and "BMI" stabilise the situation. PC7 shows positive loadings from "cigsPerDay" and "sysBP," in addition to negative loadings from "heartRate" and "BMI." Finally, a notable negative loading from "sysBP", predominating PC8, contrasts with a strong positive loading from "diaBP". The aforementioned analysis offers insights into the key elements that contribute to each principal component, which aids in the understanding of the deeper trends and patterns in the dataset. PCA was used to retain eight components which collectively explain over 97.3% of the variance in the dataset, as shown in Table 2 and the accompanying bi plot (Figure 8). The retained components capture most of the essential variability while reducing redundancy and noise. PCA was selected over feature selection methods like decision tree-based importance due to its ability to decorrelate variables and improve computational efficiency in high-dimensional data.

Detailed information on the significance of the principal components is provided in Table 2. For every principal component, it reveals the variation in proportion, cumulative variance proportion, and standard deviation. The standard deviation, which expresses the variance that each main component captures, is the square root of its eigenvalue. If a component has a larger standard deviation, it is considered to account for more volatility in the dataset. Variance is the proportion of the total variance of the dataset that each primary component explains. It is calculated by squaring the component's standard deviation and dividing the result by the total of all the eigenvalues. Cumulative Proportion is the total variance captured by all the principal components up to and including the current one. It is a running total of the 'Proportion of Variance' and shows how much of the total variance is explained by the combined effect of all the principal components up to that point. PC1 captures the most variance by far, with about 28.41% of the variance. This is a significant amount, suggesting that PC1 represents a meaningful underlying pattern in the data. PC2 accounts for an additional 13.77% of the variance, bringing the cumulative total to 42.18%. PC3 adds another 13.08% of the variance, resulting in a cumulative proportion of 55.25%. PC4 through PC7 gradually contribute less and less, with PC4 adding 11.46%, PC5 adding 11.38%, PC6 adding 10.19%, and PC7 adding 9.092% of the variance, respectively. PC8 contributes the least to the variance (2.631%), and it is often the case that later components account for less variance as the most significant patterns are captured by the initial components.

Each principal component comprises a combination of attributes contributing to cardiovascular risk. For example, PC1 is significantly influenced by cigarette consumption and blood pressure, which are known predictors of heart disease. Descriptive statistics such as variance, standard deviation, and range for each attribute are provided in Table 2, offering insights into their original distributions.

Table 2: Importance of Principal Components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Standard deviation	1.5076	1.0495	1.0228	0.9574	0.9539	0.9029	0.85284	0.45882
Proportion of Variance	0.2841	0.1377	0.1308	0.1146	0.1138	0.1019	0.09092	0.02631
Cumulative Proportion	0.2841	0.4218	0.5525	0.6671	0.7809	0.8828	0.97369	1.00000

Figure 8 shows a Bi plot representing the distribution of data after Principal Component Analysis (PCA) has been conducted. In the scatter plot, the axes are labeled `PC1` and `PC2`, which stand for Principal Component 1 and Principal Component 2, respectively. These two principal components are the new axes in a two-dimensional feature space onto which the original data has been projected. The points on the plot represent individual data items in terms of their `PC1` and `PC2` scores, which are the coordinates of each point in the new feature space. Red lines emanating from the origin point to the position of the original variables (like `cigsPerDay`, `BPMeds`, `totChol`, `sysBP`, `diaBP`, `BMI`, `heartRate`, `glucose`) on this plane. The direction and length of these lines indicate how each variable correlates with the principal components: the longer the line, the more the variable influences that principal component. The angle between the lines suggests whether the correlation between variables is positive (lines more closely directed), negative (lines more divergent), or neutral (lines are perpendicular).

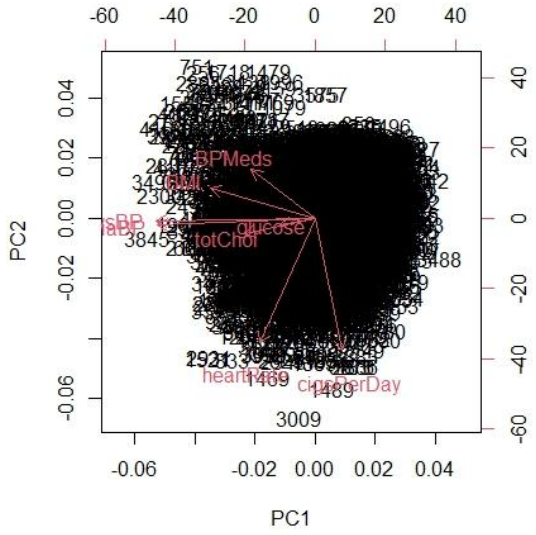


Figure 8: Biplot of PCA

6 Classification using optimized K-Nearest Neighbors (KNN) algorithm

The novelty of this study does not rest solely on the use of the KNN algorithm, which is a well-established classification method, but on how the algorithm has been carefully adapted and optimized for medical data. The proposed model systematically incorporates advanced preprocessing techniques, such as Hampel filter-based outlier removal, Min-Max normalization, and PCA for dimensionality reduction. Moreover, the study introduces a grid search-based strategy for hyperparameter tuning, complemented by 10-fold cross-validation, to empirically identify the most effective K value. These innovations collectively enhance both the accuracy and interpretability of the KNN model in cardiovascular disease prediction.

The K-Nearest Neighbors (KNN) algorithm is a straightforward and easy-to-understand machine-learning classification method suitable for both supervised and unsupervised tasks. Operating as a non-parametric and instance-based learning approach, KNN refrains from assuming any specific characteristics about the data distribution, relying on predictions derived from the similarity between data points. The key steps involved in the optimized KNN classification algorithm are:

- Loading the Data set: Load the dataset for data analysis, visualization, and modeling.
- Handling Missing Data: Identify missing values in the dataset and impute missing values for the columns by replacing them with their mean.
- Outlier Detection and Handling: Apply the Hampel Filter and median absolute deviation (MAD) to detect outliers in each variable and subsequently bound these outliers with the calculated Lower and Upper bound values for outlier handling.
- Normalization: Normalize the dataset using Min-Max normalization to bring all variables to a common scale.
- Feature Selection using Principal Component Analysis (PCA): Apply Principal Component Analysis (PCA) for feature selection and

understand the importance of each principal component.

- K-Nearest Neighbors (KNN) Classification: Split the dataset into training and testing sets in the ratio 60:40, then train the KNN model on the training set using hyperparameter tuning, considering k values ranging from 1 to 20, and determine the optimal value of k through 10-fold cross-validation.
- Evaluate KNN Model: After training the K-nearest neighbors (KNN) model on the testing set, predict the accuracy and subsequently compute and present the evaluation metrics, including the confusion matrix, precision, recall, and F1-score.

KNN utilizes a distance metric to gauge the resemblance between data points by calculating the distance of each data point in the dataset to the point intended for classification. This study employed the Euclidean distance metric, defined as the square root of the sum of squared differences between corresponding feature values. Euclidean distance is commonly used in KNN due to its simplicity and effectiveness on normalized, continuous data. Alternative metrics like Manhattan or Minkowski distance were considered, but Euclidean showed more stable results across cross-validation folds. The choice of distance metric directly influences the neighborhood formation and thus affects model accuracy. Following this, the algorithm proceeds to find the K Nearest Neighbors, pinpointing the K data points with the smallest distances to the target point, constituting the "nearest neighbors. To accomplish classification tasks, the algorithm tallies the occurrences of each class and subsequently conducts a majority vote among neighboring instances. This process allows for the consideration of weighted voting or tie-breaking methods. The class with the highest count is determined to be the anticipated class at the target point. For making predictions, the procedure associates the anticipated class with the target point by considering the majority class that occurs most often among its K nearest neighbors. To evaluate the effectiveness of the algorithm, the dataset is often divided into two separate sets, including a training set and a testing set. Subsequently, KNN is applied to the testing set, and its accuracy and other pertinent metrics are assessed to determine the algorithm's effectiveness. 'K,' or the optimal parameter, has to be identified for performance optimisation to be effective. This implies that K's value must be adjusted throughout the evaluation process. One hyperparameter that must be set before the algorithm begins is the number of nearest neighbors considered for predictions, represented by the selected value of K. Remarkably, the choice of K has an enormous effect on the performance of the algorithm. A popular approach for determining K is to compute the square root of the total number of observations in the training dataset. This technique yields an accuracy of 85.08% (K=65) and gives an initial estimate; however, the best value for "K" will vary depending on the specific dataset and should be discovered by the method termed as hyperparameter tuning. Hyperparameter tuning involves selecting the set of optimal hyperparameters for a learning algorithm. For KNN, the primary hyperparameter is the number of neighbors (K). The study utilized the Grid Search method, a more systematic approach that defines a grid of hyperparameters and exhaustively tries all combinations. For KNN, this study combined grid search with cross-validation, and the steps are:

- Define Parameter Grid: Create a grid of 'K' values you want to explore.
- Cross-Validation: Use k-fold cross-validation to estimate the effectiveness of each 'K.' This involves splitting your training set into 'k' smaller sets (folds), then training the model 'k' times, each time using a different fold as the validation set and the remaining as the training set. This study used 10-fold cross-validation.
- Search: Apply grid search to systematically work through the grid of 'K' values, training and validating the model for each.
- Best Model: The grid search process keeps track of the performance for each 'K' value and ultimately selects the one with the best cross-validated performance.

6.1 Hyperparameter tuning: K-Selection process

Hyperparameter tuning, especially the selection of the optimal number of neighbors (K), is critical in improving the performance of the K-Nearest Neighbors (KNN) algorithm. In this study, a data-driven approach was implemented to select the most suitable K value. The selection process involved evaluating multiple K values

based on their predictive performance using 10-fold cross-validation. We considered values of K ranging from 1 to 25, to strike a balance between underfitting and overfitting. Each value was assessed by measuring the average accuracy across 10 cross-validation folds on the training data. The model with the highest average cross-validation accuracy was selected as optimal. This method ensures better generalizability and avoids bias due to any single train-test split. Though computationally more intensive than a single evaluation, the use of cross-validation provides a more reliable estimate of model performance. Given the modest size of the Framingham dataset (4,240 instances), the grid search over K values was completed efficiently within seconds using RStudio, making this approach practical for real-world medical datasets. The complete step-by-step procedure for selecting the optimal K using grid search and 10-fold cross-validation is:

- Step 1: Split dataset D into training (60%) and testing (40%) sets.
- Step 2: For each K in range [1 to 25]:
- Initialize accuracy list Acc = []
- Perform 10-fold cross-validation:
- Divide training data into 10 folds.
- For each fold:
    - Train KNN on 9 folds.
    - Validate on the remaining fold.
    - Record the accuracy and append to Acc.
- Compute average accuracy AvgAcc(K) = mean (Acc)
- Step 3: Select K with the highest AvgAcc(K) as the optimal K.
- Step 4: Train the final KNN model using the full training set and optimal K.
- Step 5: Evaluate the model on the test set.

The average cross-validation accuracies for each K value in the range of 1 to 25 are summarized in Table 3, highlighting the performance trend and identifying the optimal K.

Table 3: Hyperparameter tuning results for K

K Value	Average Cross-Validation Accuracy (%)
1	86.02
2	86.9
3	88.12
4	88.95
5	89.91
6	90.18
7	90.45
8	90.89
9	91.05
10	91.12
11	91.26
12	91.4
13	91.3
14	91.72
15	91.85
16	91.93
17	92.03
18	92.15
19	92.36
20	92.46
21	92.4
22	92.32
23	92.1
24	91.87
25	91.65

The implementation of the proposed solution was carried out using R programming language in RStudio. A range of libraries were utilized to perform specific tasks: tidyverse for data manipulation, DataExplorer for exploratory data analysis and visualization of missing values, psych and lattice for descriptive statistics, car for boxplot visualization, caret for model training and evaluation, caTools for dataset splitting, and class for applying the K-Nearest Neighbors (KNN) algorithm. For performance evaluation, metrics such as accuracy, precision, recall, F1-score, and AUC were computed using the pROC and caret packages. Principal Component Analysis (PCA) was conducted using the base prcomp () function. This structured pipeline ensures transparency, reproducibility, and scientific rigor in the analysis.

7 Results and discussion

Heart disease prediction using the K-nearest neighbor (KNN) algorithm has been extensively studied in the literature. Figure 9 is a violin plot that provides a more detailed representation of the distribution of the accuracy of different machine learning algorithms in predicting cardiovascular diseases in the scientific literature and the optimized KNN algorithm.

Similar to a box plot, the violin plot provides a detailed view of the accuracy distribution of several algorithms by including markers for the mean and median. Garg et al., examined the diagnosis of cardiovascular diseases by the application of machine learning (ML) methods, such as K-Nearest Neighbor (KNN) and Random Forest. For cardiovascular disease, the two models' respective prediction accuracy was 86.885% and 81.967% [46]. In a study of supervised machine learning algorithms for predicting and diagnosing heart disease, the random forest technique beats the other four algorithms—decision tree, logistic regression, KNN, and random forest—when applied to a 70,000 sample dataset from Kaggle, with a 92% F1 score and a 95% AUC ROC [47]. In another study, Poojitha et al. examine the K Nearest Neighbor and Novel Random Forest methodologies to see the extent to which data mining algorithms predict heart disease. With a 90.16% success rate for forecasting cardiovascular disease compared to 67.21% for K Nearest Neighbor, it is concluded that the Novel Random Forest approach performs much better in terms of accuracy [48].

The following machine learning methods for classification revealed the following accuracies in an investigation using the Framingham dataset of 4240 observations: Random Forest (RF) leading with an accuracy of 85.05%, K-Nearest Neighbors (KNN) at 83.95%, Support Vector Machine (SVM) at 84.5%, Decision Tree (DT) at 84.82%, and Logistic Regression (LR) at 84.89% [49]. Using a common dataset, Aviral Chanchal et al. investigate the predictive power of many machines learning models for cardiovascular disorders, contrasting the performance of Decision Tree, KNN, Naïve Bayes, SVM, XGBoost, and Random Forest. Despite having lower accuracy percentages, Naïve Bayes, XGBoost, and Random Forest beat the other models in predicting cardiac illnesses. This was discovered by a deeper study utilizing the ROC curve and AUC values, even though KNN, SVM, and RFC exhibited high accuracy scores (85.33%) [50]. Ahmed et al. utilized algorithms such as KNN and SVM, demonstrating that KNN and SVM individually achieved accuracies of approximately 75% and 76%, respectively; a hybrid model integrating both algorithms significantly improved accuracy to 81%. This increase highlights the potential of hybrid machine-learning models in enhancing diagnostic precision in medical applications [51].

Pallathadka et al. emphasize the importance of developing accurate heart disease prediction models using data mining methods like ANN, KNN, and CNN and report that CNNs have shown the most promise in terms of utility and consistency in predicting CHD using the UCI Cleveland database [52]. A study by Gupta et al. explores the application of supervised machine-learning techniques, with Logistic Regression emerging as the superior model in terms of performance metrics, boasting the highest accuracy of 92.30% and lower false negatives compared to other classifiers, demonstrating its potential for prompt disease management. Apart from the higher performance of Logistic Regression, the research also shows that K-Nearest Neighbor (KNN) achieved competitive accuracy rates, with k values of 7 and 14 obtaining around 86.81% and 90.11%, correspondingly [53]. Multi-layer perceptron (MLP) and K-nearest neighbor (K-NN) machine learning techniques were assessed for the prediction of cardiovascular disease (CVD) in research by Pal et al. Both diagnosis rate (86.41%) and accuracy (82.47%) were better with MLP than with K-NN (73.77%) [54]. Bhatt et al. investigated several machine learning techniques and presented a model employing k-modes clustering with Huang initialization using a real-world dataset of 70,000 cases from Kaggle. Combining the multilayer perceptron with cross-validation yielded the most accurate result, outperforming previous approaches with an accuracy of 87.28% [55].

The optimized KNN model works exceptionally well for predicting cardiovascular disease (CVD), having been improved and verified for this purpose. The model achieves 0.9246 and 0.9608 F1-score metrics and accuracy with a strategically selected hyperparameter, k=20. With an overall accuracy of 92.46%, the classification model produced promising outcomes. The outcome has a greater impact on the elements crucial for minimizing the risk of CVD. The positive predictive value (precision) was 92.46%, indicating the proportion of predicted cases that were correctly classified.

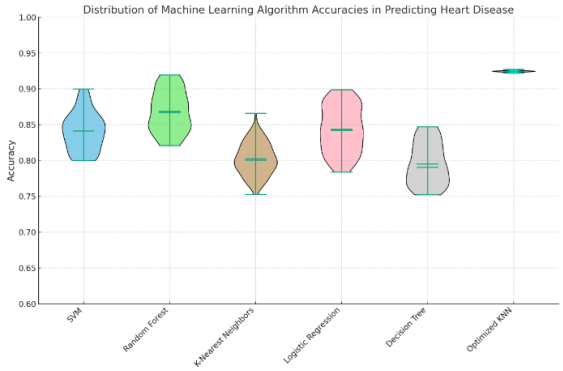


Figure 9: Violin plot showcasing the distribution of accuracies of different machine learning algorithms in predicting heart disease in the scientific literature and optimized KNN.

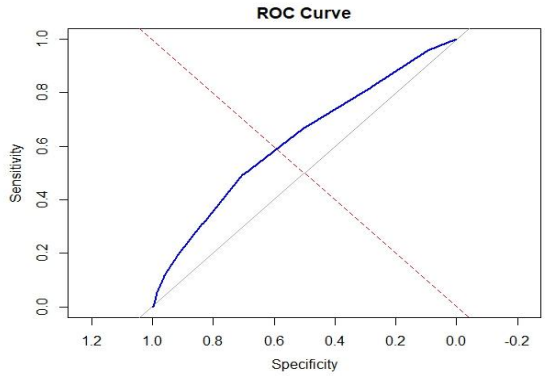


Figure 10: Receiver Operating Characteristic (ROC) Curve

Figure 10 illustrates the ROC curve, a technique employed for evaluating the effectiveness of a binary classification model. The y-axis displays the true positive rate, denoting the percentage of true positives accurately identified by the model. Meanwhile, the x-axis showcases the false positive rate, indicating the percentage of true negatives incorrectly classified as positives. This curve plots the trade-off between sensitivity and specificity (1 - false positive rate) at different thresholds. A model with perfect discrimination (no overlap between the two distributions of the binary classifier) would have a curve that goes straight up the y-axis and straight across at a true positive rate of 1. The Area Under the Curve (AUC=0.6216) condenses the entire ROC curve into a singular metric. A perfect test is denoted by a value of 1, while a worthless test is indicated by 0.5. A higher AUC signifies better discrimination between positive and negative classes. The ROC curve of a random classifier is represented by the diagonal dashed line, corresponding to an AUC of 0.5. The curve above this line indicates that the classifier has a better-than-random ability to discriminate between the two classes. The steepness of the curve at different points can indicate how thresholds can be adjusted to optimize for either sensitivity or specificity. A steep initial rise indicates that a small decrease in specificity will gain a large increase in sensitivity. Based on the ROC curve, the model is evaluated to have a good performance in distinguishing between the positive and negative classes, but there is room for improvement.

In the context of cardiovascular disease (CVD) prediction, the choice of evaluation metrics plays a crucial role in assessing a model's clinical relevance. While our optimized KNN model demonstrated strong performance with an accuracy of 92.46% and F1-scores of 0.9246 and 0.9608, relying solely on accuracy can be misleading due to the class imbalance present in the Framingham dataset. In such datasets, a model may perform well on the majority class (non-CVD) while failing to identify the minority class (CVD), thus inflating the overall accuracy.

To address this, we incorporated additional metrics such as precision, recall, and F1-score, which provide a better understanding of the model's ability to correctly identify positive cases. The F1-score, as the harmonic mean of precision and recall, is especially useful when the cost of false negatives is high—as in medical diagnosis. Our model's high F1-score indicates a good balance between sensitivity and specificity.

However, a discrepancy arises with the AUC-ROC score, which is 0.6216. This value, while better than random guessing (AUC = 0.5), indicates that the model's ability to distinguish between positive and negative classes is moderate. This is likely due to class imbalance and the nature of KNN, which does not output calibrated probability scores. While the model may classify well at a specific threshold, its probability estimates do not align closely with the true likelihood of disease, limiting its usefulness for clinical risk stratification.

To compute these metrics, the model first constructs a confusion matrix from true and predicted labels to determine true positives, false positives, true negatives, and false negatives. From this, accuracy is calculated as the proportion of correct predictions; precision is the ratio of true positives to all predicted positives; recall is the ratio of true positives to all actual positives; F1-score combines both precision and recall; and the AUC-ROC represents the area under the curve plotting the true positive rate against the false positive rate across different thresholds.

To improve AUC and overall discrimination, future enhancements could include probability calibration methods (like Platt scaling), resampling techniques such as SMOTE to handle class imbalance, and the use of precision-recall curves to better evaluate model performance under imbalance.

Figure 11 is a histogram based on the probability of predictions used to evaluate the binary classifier. It

illustrates the data distribution by creating bins across the data range and subsequently using bars to represent the quantity of observations within each bin. The data is categorized into two groups represented by different colors: red for "factor(Actual) 0" and teal for "factor(Actual) 1." The horizontal axis (X-axis) represents predicted probabilities, ranging from 0 to approximately 0.6. The vertical axis (Y-axis) shows the count of occurrences for each probability bin. The red bars show a high frequency of predicted probabilities around 0.1, indicating that for the factor level 0, the model predicted a low probability. The teal bars, which are fewer in number, also show predictions mostly in the lower probability range but are more spread out than the red bars. For the red group (0), the model has high confidence in its predictions as the probabilities are clustered around a peak. The more evenly dispersed probabilities for group 1 suggest a lower or more diverse level of confidence. The dispersion of the teal distribution can signal less confidence in assigning a positive class (1), whereas the concentration of red at lower probabilities shows that the model is confident in giving a negative class (0). A typical problem that may impact the effectiveness of classification algorithms is a class imbalance in the dataset, shown by more red bars than teal (i.e., more factor 0 than factor 1).

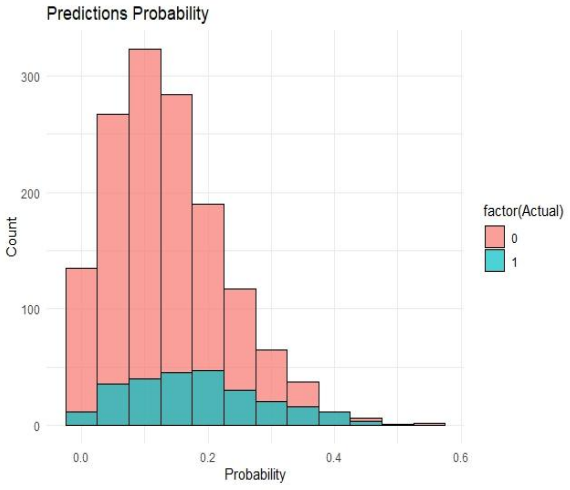


Figure 11: Probability of predictions

Although the overall accuracy of the model is impressive, it has difficulties in correctly identifying instances of the positive class. To increase the model's ability to forecast positive instances, further optimization and maybe even a solution to the class imbalance problem could be needed. The results of the study spotlight the significance of model refinement methods such as feature selection and hyperparameter tuning in improving the classification accuracy. The optimized KNN algorithm analyzes the Framingham dataset of 191 KB in size and 4240 observations in 7.92 seconds. As the size of the dataset increases there is a notable increase in the execution time. But, in medical prognosis precision is more crucial than speed.

8 Conclusion

In the medical industry, early detection and precise diagnosis of cardiovascular disorders are essential since they can greatly enhance patient outcomes. This study presents an optimized approach for K-Nearest Neighbors and shows a significant increase in cardiovascular disease prediction when applied. The meticulous combination of intricate feature selection methods, principal component analysis (PCA) for dimensionality reduction, and hyperparameter tuning yields an exceptionally accurate and efficient model. The optimized KNN model performs better in early CVD detection than typical KNN models, as evidenced by its remarkable metrics and prediction accuracy of 92.46%. The complexity of medical data can be accommodated by customizing machine learning algorithms, as demonstrated by this study. Improving preventive health tactics and possibly saving lives requires the integration of these cutting-edge techniques into clinical procedures. More widespread applications in healthcare are possible as a result of the study's foundational principles and methods, which can be applied to other complex illness projections. Future research might concentrate on correcting the dataset's class imbalance in order to improve the KNN model's capacity to identify instances of the positive class more precisely. Advanced tactics may be used to further enhance the prediction performance of the model and lessen the effects of class imbalance. Typical examples of these strategies include resampling methods and the use of stacked and ensemble approaches. The true innovation of this study lies in presenting a robust and reproducible framework for enhancing KNN-based classification in the context of medical diagnosis. Rather than introducing a novel algorithm, this research demonstrates how existing algorithms can be significantly improved through systematic optimization strategies. The combination of Hampel filtering, PCA-based feature selection, and cross-validated hyperparameter tuning delivers a highly accurate and computationally efficient model. Future studies can further extend this work by integrating ensemble-based or hybrid learning strategies and addressing class imbalance through advanced resampling techniques such as SMOTE.

References

[1] E. Maini, B. Venkateswarlu, B. Maini, and D. Marwaha, "Machine learning–based heart disease prediction system for Indian population: An exploratory study done in South India," *Med. J. Armed Forces India*, vol. 77, no. 3, pp. 302–311, Jul. 2021, doi: 10.1016/j.mjafi.2020.10.013.

[2] A. Rajdhan, A. Agarwal, M. Sai, D. Ravi, and D. P. Ghuli, "Heart Disease Prediction using Machine Learning," *Int. J. Eng. Res.*, vol. 9, no. 04. doi: 10.17577/IJERTV9IS040614.

[3] D. A. Anggoro, "Comparison of Accuracy Level of Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) Algorithms in Predicting Heart Disease," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 5, pp. 1689–1694, May 2020, doi: 10.30534/ijeter/2020/32852020.

[4] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India: IEEE, May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.

[5] H. Yepdjio and S. Vajda, "Optimization Strategies for the k-Nearest Neighbor Classifier," *SN Comput. Sci.*, vol. 4, Nov. 2022, doi: 10.1007/s42979-022-01469-3.

[6] M. Muzammal, R. Talat, A. H. Sodhro, and S. Pirbhulal, "A multi-sensor data fusion enabled ensemble approach for medical data from body sensor networks," *Inf. Fusion*, vol. 53, pp. 155–164, Jan. 2020, doi: 10.1016/j.inffus.2019.06.021.

[7] H. Yang and J. M. Garibaldi, "A hybrid model for automatic identification of risk factors for heart disease," *Suppl. Proc. 2014 I2b2UTHealth Shar.-Tasks Workshop Chall. Nat. Lang. Process. Clin. Data*, vol. 58, pp. S171–S182, Dec. 2015, doi: 10.1016/j.jbi.2015.09.006.

[8] V. Nagavallika, 'Heart disease prediction using machine learning techniques', Int. J. Sci. Res. (Raipur), vol. 10, no. 11, pp. 630–633, Nov. 2021, doi: 10.21275/SR21918142603.

[9] A. C. Dimopoulos *et al.*, "Machine learning methodologies versus cardiovascular risk scores, in predicting disease risk," *BMC Med. Res. Methodol.*, vol. 18, no. 1, p. 179, Dec. 2018, doi: 10.1186/s12874-018-0644-1.

[10] Prof. Madhavi Tota, Manthan Moon, Pranit Nagrale, Akshay Pandav, and Gunjan Das, "Heart Diseases Prediction System using ML," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 337–345, Dec. 2022, doi: 10.48175/IJARSCT-7798.

[11] B. Jin, C. Che, Z. Liu, S. Zhang, X. Yin, and X. Wei, "Predicting the Risk of Heart Failure With EHR Sequential Data Modeling," *IEEE Access*, vol. 6, pp. 9256–9261, 2018, doi: 10.1109/ACCESS.2017.2789324.

[12] A. S. S. Kotia, M. Rastogi, and R. A. Bhongade, "Use of machine learning techniques for effective prediction of heart disease," *CARDIOMETRY*, no. 26, pp. 315–321, Mar. 2023, doi: 10.18137/cardiometry.2023.26.315321.

[13] D. Shah, S. Patel, and S. K. Bharti, "Heart Disease Prediction using Machine Learning Techniques," *SN Comput. Sci.*, vol. 1, no. 6, p. 345, Oct. 2020, doi: 10.1007/s42979-020-00365-y.

[14] E. D. Adler *et al.*, "Improving risk prediction in heart failure using machine learning," *Eur. J. Heart Fail.*, vol. 22, no. 1, pp. 139–147, Jan. 2020, doi: 10.1002/ejhf.1628.

[15] I. M. Pires, G. Marques, N. M. Garcia, and V. Ponciano, "Machine learning for the evaluation of the presence of heart disease," *Procedia Comput. Sci.*, vol. 177, pp. 432–437, 2020, doi: 10.1016/j.procs.2020.10.058.

[16] M. M. Ali, B. K. Paul, K. Ahmed, F. M. Bui, J. M. W. Quinn, and M. A. Moni, "Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison," *Comput. Biol. Med.*, vol. 136, p. 104672, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104672.

[17] H. Kahramanli and N. Allahverdi, "Design of a hybrid system for the diabetes and heart diseases," *Expert Syst. Appl.*, vol. 35, no. 1–2, pp. 82–89, Jul. 2008, doi: 10.1016/j.eswa.2007.06.004.

[18] A. Kondababu, V. Siddhartha, BHK. B. Kumar, and B. Penumutchi, "A comparative study on machine learning based heart disease prediction," *Materials Today: Proceedings*, Feb. 2021, doi: 10.1016/j.matpr.2021.01.475.

[19] S. Faiayaz Waris and S. Koteeswaran, "Heart disease early prediction using a novel machine learning method called improved K-means neighbor classifier in python," *Materials Today: Proceedings*, Mar. 2021, doi: 10.1016/j.matpr.2021.01.570.

[20] R. Gopal and V. Ranganathan, "Evaluation of effect of unsupervised dimensionality reduction techniques on automated arrhythmia classification," *Biomed. Signal Process. Control*, vol. 34, pp. 1–8, Apr. 2017, doi: 10.1016/j.bspc.2016.12.017.

[21] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, *Feature extraction. Foundations and applications. Papers from NIPS 2003 workshop on feature extraction, Whistler, BC, Canada, December 11–13, 2003. With CD-ROM*, vol. 207. 2006. doi: 10.1007/978-3-540-35488-8.

[22] N. R. Ratnasari, A. Susanto, I. Soesanti, and Maesadji, "Thoracic X-ray features extraction using thresholding-based ROI template and PCA-based features selection for lung TB classification purposes," in *2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME)*, Nov. 2013, pp. 65–69. doi: 10.1109/ICICI-BME.2013.6698466.

[23] P. Kamencay, R. Hudec, M. Benco, and M. Zachariasova, "Feature extraction for object recognition using PCA-KNN with application to medical image analysis," in *2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2013, pp. 830–834. doi: 10.1109/TSP.2013.6614055.

[24] Yuda Syahidin, Aditya Pratama Ismail, and Fawwaz Nafis Siraj, "Application of Artificial Neural Network Algorithms to Heart Disease Prediction Models with Python Programming," *J. E-Komtek Elektro-Komput.-Tek.*, vol. 6, no. 2, pp. 292–302, Dec. 2022, doi: 10.37339/e-komtek.v6i2.932.

[25] Yichun Wang, "Heart disease prediction with discriminative deep neural network," presented at the Proc.SPIE, May 2023, p. 126401P. doi: 10.1117/12.2673756.

[26] S. .S, S. Lavanya, M. R. Chandhini, R. Bharathi, and K. Madhulekha, "Hybrid Machine Learning Techniques for Heart Disease Prediction," *Int. J. Adv. Eng. Res. Sci.*, vol. 7, pp. 44–48, Jan. 2020, doi: 10.22161/ijaers.73.7.

[27] D. M and R. Abirami, "Heart Disease Prediction System using Ensemble of Machine Learning Algorithms," *Recent Patents on Engineering*, vol. 15, pp. 130–139, Mar. 2021, doi: 10.2174/1872212113666190328220514.

[28] R. R. K. AL-Taie, B. J. Saleh, A. Y. Falih Saedi, and L. A. Salman, "Analysis of WEKA data mining algorithms Bayes net, random forest, MLP and SMO for heart disease prediction system: A case study in Iraq," *IJECE*, vol. 11, no. 6, p. 5229, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5229-5239.

[29] O. Akbilgic *et al.*, "ARTIFICIAL INTELLIGENCE APPLIED TO ECG IMPROVES HEART FAILURE PREDICTION ACCURACY," *ACC.21*, vol. 77, no. 18, Supplement 1, p. 3045, May 2021, doi: 10.1016/S0735-1097(21)04400-4.

[30] O. W. Samuel *et al.*, "A new technique for the prediction of heart failure risk driven by hierarchical neighborhood component-based learning and adaptive multi-layer networks," *Future Gener. Comput. Syst.*, vol. 110, pp. 781–794, Sep. 2020, doi: 10.1016/j.future.2019.10.034.

[31] S. Alagarsamy, K. Kamatchi, K. Selvaraj, A. Subramanian, L. R. Fernando, and R. Kirthikaa, "Identification of Brain Tumor using Deep Learning Neural Networks," in *2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES)*, Dec. 2019, pp. 1–5. doi: 10.1109/INCCES47820.2019.9167685.

[32] K. Kartheeban, K. Kalyani, S. K. Bommavaram, D. Rohatgi, M. N. Kathiravan, and S. Saravanan, "Intelligent Deep Residual Network based Brain Tumor Detection and Classification," in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, Dec. 2022, pp. 785–790. doi: 10.1109/ICACRS55517.2022.10029146.

[33] V. Shankar, V. Kumar, U. Devagade, V. Karanth, and K. Rohitaksha, "Heart Disease Prediction Using CNN Algorithm," *SN Comput. Sci.*, vol. 1, no. 3, p. 170, May 2020, doi: 10.1007/s42979-020-0097-6.

[34] A. Dutta, T. Batabyal, M. Basu, and S. T. Acton, "An efficient convolutional neural network for coronary heart disease prediction," *Expert Syst. Appl.*, vol. 159, p. 113408, Nov. 2020, doi: 10.1016/j.eswa.2020.113408.

[35] S. A. H. Fazlur and S. K. Thillaigovindan, "Integrated Deep Learning Model for Heart Disease Prediction Using Variant Medical Data Sets," *Int. J. Online Biomed. Eng. IJOE*, vol. 18, no. 09, pp. 178–191, Jul. 2022, doi: 10.3991/ijoe.v18i09.30801.

[36] M. Sudipta, E. Abdel-Raheem, and L. Rueda, *Heart Disease Prediction Using Adaptive Infinite Feature Selection and Deep Neural Networks*. 2022, p. 240. doi: 10.1109/ICAIIC54071.2022.9722652.

[37] K. V. Reddy, I. Elamvazuthi, A. A. Aziz, S. Paramasivam, H. N. Chua, and S. Pranavanand, "Heart Disease Risk Prediction Using Machine Learning Classifiers with Attribute Evaluators," *Appl. Sci.*, vol. 11, no. 18, 2021, doi: 10.3390/app11188352.

[38] S. Ahmed *et al.*, "Prediction of Cardiovascular Disease on Self-Augmented Datasets of Heart Patients Using Multiple Machine Learning Models,"

J. Sens., vol. 2022, p. 3730303, Dec. 2022, doi: 10.1155/2022/3730303.

[39] K. V. V. Reddy, I. Elamvazuthi, A. A. Aziz, S. Paramasivam, and H. N. Chua, "Heart Disease Risk Prediction using Machine Learning with Principal Component Analysis," in *2020 8th International Conference on Intelligent and Advanced Systems (ICIAS)*, Jul. 2021, pp. 1–6. doi: 10.1109/ICIAS49414.2021.9642676.

[40] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019, doi: 10.1109/ACCESS.2019.2923707.

[41] A. B. Ambrews, E. Gubin Moung, A. Farzamnia, F. Yahya, S. Omatu, and L. Angeline, "Ensemble Based Machine Learning Model for Heart Disease Prediction," in *2022 International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, Nov. 2022, pp. 1–6. doi: 10.1109/CIEES55704.2022.9990665.

[42] S. P. Patro, N. Padhy, and R. D. Sah, "Classification model for heart disease prediction using correlation and feature selection techniques," in *2022 OITS International Conference on Information Technology (OCIT)*, Dec. 2022, pp. 29–34. doi: 10.1109/OCIT56763.2022.00016.

[43] M. I. Ahmed and F. Shefaq, "A Study on Machine Learning and Supervised and Deep Learning Algorithms to Predict the Risk of Patients: Ten Year Coronary Heart Disease," *Int. J. Pract. Healthc. Innov. Manag. Tech. IJPHIMT*, vol. 9, no. 1, pp. 1–12, 2022, doi: 10.4018/IJPHIMT.305127.

[44] S. Patro and Dr. N. Padhy, "An RHMIoT Framework for Cardiovascular Disease Prediction and Severity Level Using Machine Learning and Deep Learning Algorithms," *Int. J. Ambient Comput. Intell.*, vol. 13, pp. 1–37, Jan. 2022, doi: 10.4018/IJACI.311062.

[45] R. Aggrawal and S. Pal, "Elimination and Backward Selection of Features (P-Value Technique) In Prediction of Heart Disease by Using Machine Learning Algorithms," *Turk. J. Comput. Math. Educ. TURCOMAT*, vol. 12, pp. 2650–2665, Apr. 2021, doi: 10.17762/turcomat.v12i6.5765.

[46] A. Garg, B. Sharma, and R. Khan, "Heart disease prediction using machine learning techniques," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1022, no. 1, p. 012046, Jan. 2021, doi: 10.1088/1757-899X/1022/1/012046.

[47] S. Yousefi and M. Poornajaf, "Analysis of Accuracy Metric of Machine Learning Algorithms in Predicting Heart Disease," *Front. Health Inform. Vol 12 2023 Contin. Vol. - 1030699fhiv12i0402*, Apr. 2023, [Online]. doi: 10.30699/fhi.v12i0.402.

[48] T. Poojitha and R. Mahaveerakannan, "Prediction Analysis of Novel Random Forest Algorithm and K Nearest Neighbor Algorithm in Heart Disease Prediction with an Improved Accuracy Rate," *CARDIOMETRY*, no. 25, pp. 1554–1561, Feb. 2023, doi: 10.18137/cardiometry.2022.25.15541561.

[49] W. A. Mahmoud and D. M. Aborizka, "Heart Disease Prediction Using Machine Learning and Data Mining Techniques: Application of Framingham Dataset," 2021. Available at: https://www.idosr.org/wp-content/uploads/2021/11/IDOSR-JCAS-6166-73-2021..pdf.

[50] A. Chanchal, A. S. Singh, and K. Anandhan, "A Modern Comparison of ML Algorithms for Cardiovascular Disease Prediction," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Sep. 2021, pp. 1–5. doi: 10.1109/ICRITO51393.2021.9596228.

[51] R. Ahmed, M. Bibi, and S. Syed[3], "Improving Heart Disease Prediction Accuracy Using a Hybrid Machine Learning Approach: A Comparative study of SVM and KNN Algorithms," *Int. J. Comput. Inf. Manuf. IJCIM*, vol. 3, p. 2023, Jun. 2023, doi: 10.54489/ijcim.v3i1.223.

[52] H. Pallathadka, M. Naved, K. Phasinam, and M. M. Arcinas, "A Machine Learning Based Framework for Heart Disease Detection," *ECS Trans.*, vol. 107, no. 1, pp. 8667–8673, Apr. 2022, doi: 10.1149/10701.8667ecst.

[53] C. Gupta, A. Saha, N. V. Subba Reddy, and U. Dinesh Acharya, "Cardiac Disease Prediction using Supervised Machine Learning Techniques.," *J. Phys. Conf. Ser.*, vol. 2161, no. 1, p. 012013, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012013.

[54] M. Pal, S. Parija, G. Panda, K. Dhama, and R. K. Mohapatra, "Risk prediction of cardiovascular disease using machine learning classifiers," *Open Med.*, vol. 17, no. 1, pp. 1100–1113, Jun. 2022, doi: 10.1515/med-2022-0508.

[55] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective Heart Disease Prediction Using Machine Learning Techniques," *Algorithms*, vol. 16, no. 2, p. 88, Feb. 2023, doi: 10.3390/a16020088.

District-Level Rainfall and Cloudburst Prediction Using XGBoost: A Machine Learning Approach for Early Warning Systems

Guru Dayal Kumar[1], Shekhar Tyagi[2], Kalandi Charan Pradhan[1], Akshat Shah[2,3]
[1]Migration and Development Research Group, School of Humanities and Social Sciences, Indian Institute of Technology Indore, 452020, India
[2]Department of Computer Science and Engineering, Indian Institute of Technology Indore, 452020, India
[3]Department of Computer Science and Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, 440001, India
E-mail: shekhartyagirocks007@gmail.com

Keywords: Cloudbursts, rainfall, XGBoost, flood

Received: November 16, 2024

This research presents a novel methodology for cloudburst forecasting using the XGBoost (Extreme Gradient Boosting) machine learning algorithm. With the escalating impact of climate change, accurately predicting extreme weather events like cloudbursts is crucial due to their potential to trigger floods. Cloudburst events were identified from daily rainfall data. Our study leverages historical weather data, focusing on intricate rainfall patterns, to forecast future cloudburst occurrences. Comparative analysis against Random Forest and LSTM models confirmed XGBoost's effectiveness, consistently outperforming alternatives across multiple performance metrics. The XGBoost model, known for its ability to handle complex datasets, demonstrated strong predictive performance, with an RMSE of 0.12 and an MAE of 0.09, indicating high accuracy. This research provides a reliable tool for advanced weather forecasting and early warning systems, offering valuable support to policymakers, disaster management teams, and agricultural planners in mitigating risks associated with extreme rainfall events.

Povzetek: Raziskava uvaja model XGBoosta za napoved padavin in lokalnih nalivov na ravni okrožij, ki omogoča učinkovite opozorilne sisteme za ekstremne vremenske razmere.

1 Introduction

Excessive rainfall phenomena, such as cloudbursts, typically occur on a mesoscale level, spanning 2–20 km [50]. These events are marked by sudden and intense precipitation surges, often resulting in secondary hazards like flash floods, landslides, and dam failures [15, 13]. Predominantly occurring during the monsoon season, cloudbursts are among the most significant natural hazards in the region. Various studies have proposed rainfall intensity thresholds to identify cloudburst events [28, 48, 26]. For example, [27] defined extreme rainfall events as those exceeding 250 mm/day, while [48] categorized heavy rainfall in the northwest Himalayas (NWH) as surpassing 200 mm/day. Other studies, such as [7] and [59], utilized the 99.99th percentile of precipitation distribution to delineate cloudbursts. The study [62] proposes an innovative approach for missing value imputation using an extended Kalman filter with linear relations and introduces advanced bidirectional and unidirectional LSTM architectures for enhancing network-wide rainfall forecasting in ubiquitous computing environments. The pie chart illustrates the distribution of various natural disasters. The pie chart(Figure 1) illustrates the distribution of various natural disasters. Floods account for the highest proportion (50%), followed by storms (32%) and extreme temperatures (10%), while glacial lake outburst floods and droughts have the lowest occurrences at 1% and 3%, respectively.

These extreme weather events often result in significant loss of life and property. A notable instance was the Kedarnath tragedy in Uttarakhand in June 2013, where rainfall intensities exceeding 200 mm/day over several days caused over 6,000 fatalities [53]. Approximately 50 million people inhabit the Himalayan region across Nepal, India, Bhutan, China, and Pakistan, making the prediction of cloudbursts crucial for safeguarding these vulnerable populations. However, existing meteorological models struggle to achieve accurate cloudburst prediction due to their reliance on deterministic weather forecasting, which often fails to capture the complex, non-linear relationships between atmospheric variables [13, 10]. Traditional models, such as numerical weather prediction (NWP) techniques, suffer from limitations in spatial resolution, dependency on large-scale climate patterns, and computational inefficiencies, making them inadequate for real-time early warning systems.

Recent examples highlight the devastating impact of such events. In 2022, Bengaluru experienced severe flooding, with the city recording 132 mm of rainfall within 24 hours on September 5, accounting for 10% of its seasonal

rainfall. The floods caused an estimated loss of over Indian rupees 2,250 million [22]. Similarly, Bihar has faced recurring floods with varying intensity, causing widespread damage over the years, including major episodes in 1987, 1998, 2000, 2001, 2003, 2004, 2008, 2010, 2013, 2017, 2018, and 2020 [63]. States such as Bihar, West Bengal, and Uttar Pradesh, situated along the Ganges River, are especially susceptible to natural disasters, with climatic risks exacerbating the trends of loss and destruction [55]. In Bihar alone, approximately 68.20 million people—roughly 53.20% of its population—reside in high-risk flood zones [55]. The state's vulnerability is further underscored by the severe flooding of 6.970 million hectares of land, affecting 74.0% of its geographical area [4]. To address these challenges, this study leverages machine learning, specifically the XGBoost algorithm, to enhance cloudburst prediction. Unlike traditional meteorological models, XGBoost is capable of handling high-dimensional data, capturing intricate relationships among multiple atmospheric parameters, and reducing prediction error through its boosting mechanism. The adaptive learning approach of XGBoost makes it particularly well-suited for cloudburst forecasting, allowing for real-time predictions with higher accuracy. This research aims to demonstrate how XGBoost outperforms conventional models in forecasting cloudbursts, offering a more robust and data-driven early warning system for disaster mitigation.

2 Related studies on cloudbursts and early warning system

The term cloudburst holds a notable historical presence in meteorological literature, with references dating back to the 19th century [43]. A comparative analysis using the Google Books Ngram Viewer reveals that the term emerged in the 1800s and reached its peak usage in the 1940s. Initially, cloudbursts were described as localized heavy rainfall events often linked to thunderstorms, though their formal definition evolved over time. For instance, Elmer [20] suggested that elongated thunderstorm clouds moving along their longitudinal axis could directly trigger cloudbursts. Similarly, Bonnett [8] described scenarios where showers intensified progressively, eventually covering the sky and culminating in severe thunderstorms. Horton and Todd [30] emphasized the highly localized nature of these events, citing the Taborton, New York, incident where 158 mm of rain fell in two hours over an 8 km-wide area.

King [35] reported a cloudburst lasting 3.5 hours, producing 305 mm of rain across an elliptical region of 80 square kilometers, causing significant destruction, including impassable roads, swept-away bridges and homes, debris-laden farmland, 11 fatalities, and property damage equivalent to USD 6.8 million today. Douglas [17] documented a California cloudburst that led to a flash flood accompanied by a dust cloud from dislodged dry soil rushing down a canyon. Similarly, the July 2, 1893, cloudburst over the Cheviot Hills (UK) caused erosion of up to 2 square meters of valley cross-section, destroying bridges and roads [11].

By the mid-20th century, cloudbursts were widely understood as localized, high-intensity rainfall events spanning a few kilometers, often accompanied by thunder and lightning [64, 47]. These events could result in extensive damage, including flash floods, streambed erosion, landslides, and mudflows. Woolley [65] provided a formal definition, describing cloudbursts as torrential rainfall events characterized by intensity and spatial concentration, akin to the sudden release of an entire cloud. Typically associated with thunderstorms, these events occur over limited areas and produce volatile, damaging floods in steep catchments. They are also linked to cumulonimbus clouds and hazardous phenomena such as squalls, strong winds, hailstorms, and tornadoes [14, 18].

Quantitative definitions of cloudbursts vary. Haritashya et al. [29] proposed a threshold of 100 mm/h to classify a violent shower as a cloudburst. Krishnamurthy [37] used 100 mm/day as a criterion for extreme rainfall events, while Izzo [34] defined cloudbursts as having rainfall intensity above 30 mm/h. Dunlop [19] differentiated heavy showers (10–50 mm/h) from violent showers (> 50 mm/h), following World Meteorological Organization (WMO) guidelines. Fry et al. [23] defined downpours as rainfall exceeding 15 mm/h. The American Meteorological Society's Glossary of Meteorology supports the threshold of 100 mm/h for defining cloudbursts [1]. Consequently, in non-monsoonal regions, cloudbursts are defined as rainfall events with intensities exceeding 30 mm/h, while in monsoonal regions, the threshold is 100 mm/h. These events typically affect areas of a few kilometers, often causing flash floods, landslides, and mudflows, and are frequently accompanied by storms, strong winds, hail, and tornadoes.

The Centre for Research on the Epidemiology of Disasters (CRED) has developed the Emergency Events Database (EM-DAT), recognized as the most comprehensive global database on over 23,000 natural and technological disasters from 1900 to 2022 [2]. EM-DAT systematically records disaster data annually. Analysis of the database highlights the most impactful natural disasters, including droughts, earthquakes, extreme temperatures, floods, glacial lake outbursts, storms, and wildfires. Notably, more than 50 percent of these recorded events are floods and glacial lake outburst floods [17, 18, 11, 15].

Both fluvial and pluvial flooding are expected to increase the vulnerability of residents in riparian and informal communities due to projected rises in rainfall intensity driven by climate change [46]. Early Warning Systems (EWSs) serve as a critical intersection of disaster risk reduction, effective humanitarian response, and the promotion of climate-resilient and sustainable development. They address present, emerging, and potential flood-related hazards. However, Africa lags behind other global regions in implementing robust EWSs [44, 66]. Several studies have highlighted the pivotal role of EWSs in disaster prepared-

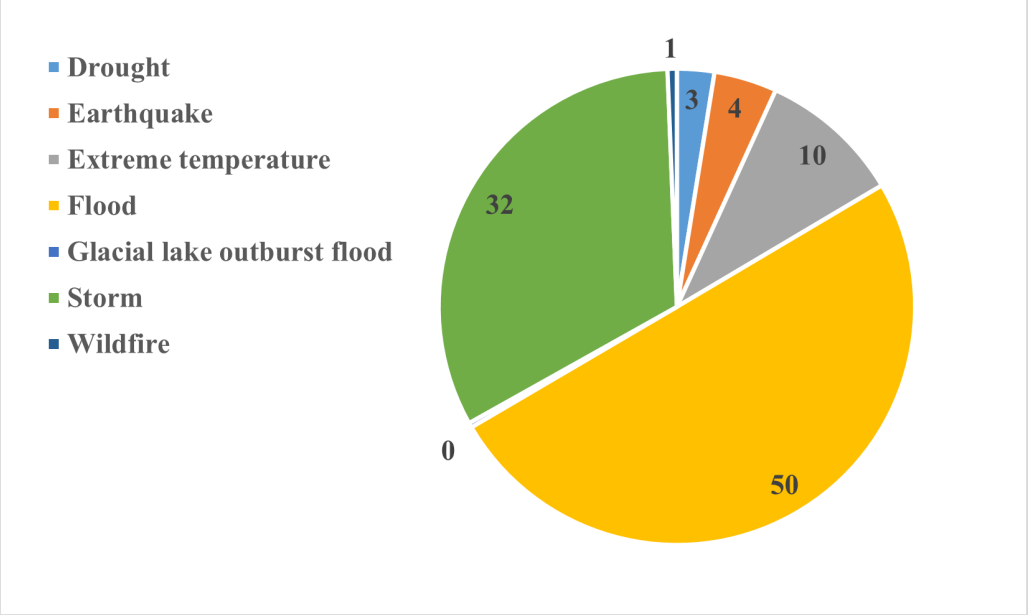


Figure 1: Sum of occurrences of natural disasters from 1900 to 2022

ness and mitigation [12, 21, 25]. Research efforts, such as those by Roy et al. [56] and Shahjahan [57], have assessed the efficacy of EWSs in Bangladesh's Sundarbans region, focusing on methods of warning dissemination and reception. In contrast, similar systems have not been extensively studied in the Indian state of Bihar. Thus, it is crucial to evaluate the effectiveness of existing EWSs in bolstering disaster mitigation efforts in Eastern India, particularly Bihar [39].

Climate change has led to a significant increase in the frequency and severity of extreme weather events, with cloudbursts emerging as critical hazards impacting ecosystems, agriculture, and human settlements [49]. Accurate forecasting of such events is essential for proactive planning and mitigation strategies. While conventional weather prediction methods provide some insights, they often fall short in predicting the intensity and occurrence of extreme events like cloudbursts. This study explores the use of XGBoost [9], an advanced machine learning algorithm, to bridge this gap. By analyzing historical weather data, the research aims to develop a predictive model with enhanced accuracy and reliability for cloudburst forecasting.

The utility of XGBoost in cloudburst prediction is underscored by recent advancements in meteorological research. Two prominent studies illustrate XGBoost's capabilities in this domain. The first study highlights its effectiveness in short-term precipitation forecasting across diverse climatic regions of China, employing multi-factor bias correction to improve accuracy [16]. The second study demonstrates XGBoost's proficiency in nowcasting weather conditions, outperforming traditional methods such as SVM (Support Vector Machine), RF (Random Forest), and GBDT (Gradient Boosting Decision Tree) [45].

In hydrological forecasting, machine learning algorithms, particularly XGBoost, have shown remarkable success in predicting groundwater levels and streamflow across various geographical terrains. Studies have demonstrated XGBoost's superiority in groundwater level predictions [51] and streamflow forecasting [61]. Furthermore, Kumar et al. [40] employed LSTM neural networks for rainfall forecasting and flood impact predictions in Bihar, leveraging deep learning to improve disaster preparedness and response. Another study by Kumar et al. [41] applied AI-driven models for assessing rainfall and flood vulnerability in Bihar, aiming to enhance disaster management strategies in the region. Table 1 presents a comprehensive summary of existing studies on cloudburst prediction and early warning systems, highlighting the methodologies used and identifying current research gaps. These findings collectively affirm the critical role of XGBoost in cloudburst prediction, presenting promising opportunities for enhanced forecasting precision. The superiority of XGBoost over Random Forest, SVM, and LSTM in precipitation forecasting is highlighted in Table 2, emphasizing its advantages in accuracy, scalability, and handling complex data. This research primarily addresses the challenge of accurately predicting cloudbursts—extreme precipitation events occurring over short durations that pose significant risks. The unpredictable and severe nature of these phenomena makes them a central focus in climate science and meteorological research. By leveraging historical weather data, this study aims to develop predictive models that estimate the likelihood of future cloudburst occurrences, providing a valuable tool for preemptive and preparatory measures.

The study is organized as follows: the section on the study area and data is followed by a detailed description of the proposed methodology and computational framework for rainfall and cloudburst prediction using XGBoost. Next,

Table 1: Summary of studies related to cloudburst and early warning systems

Study	Focus Area	Key Findings
Haritashya et al. [29]	Quantitative definition of cloudbursts	Proposed a threshold of 100 mm/h to classify a violent shower as a cloudburst.
Krishnamurthy [37]	Extreme rainfall events	Used 100 mm/day as a criterion for classifying extreme rainfall events.
Izzo [34]	Rainfall intensity classification	Defined cloudbursts as rainfall events with intensity above 30 mm/h.
Dunlop [19]	Rainfall intensity classification	Differentiated between heavy showers (10–50 mm/h) and violent showers (> 50 mm/h).
Fry et al. [23]	Heavy rainfall categorization	Defined downpours as rainfall exceeding 15 mm/h.
Roy et al. [56]	Early warning systems in Bangladesh	Assessed the effectiveness of warning dissemination and reception methods in the Sundarbans region.
Shahjahan [57]	Disaster mitigation in Bangladesh	Evaluated EWS efficacy in disaster-prone areas.
Kumar et al. [39]	EWS in Bihar, India	Highlighted the need to assess and improve EWS efficacy in Bihar.
Dong et al. [16]	XGBoost for precipitation forecasting	Demonstrated XGBoost's effectiveness in short-term precipitation forecasting using multi-factor bias correction.
Mai et al. [45]	Weather nowcasting with ML models	Showed XGBoost outperforming SVM, RF, and GBDT in nowcasting weather conditions.
Osman et al. [51]	Groundwater level prediction	Validated XGBoost's superiority in predicting groundwater levels across varied terrains.
Szczepanek et al. [61]	Streamflow forecasting	Highlighted the accuracy of XGBoost in daily streamflow prediction.
Kumar et al. [40]	Rainfall and flood impact prediction in Bihar	Applied LSTM neural networks for accurate rainfall forecasting and flood impact assessments.
Kumar et al. [41]	AI-driven models for disaster management	Developed predictive models for assessing rainfall and flood vulnerability in Bihar.

the results and analysis are presented, and the final section concludes the study.

3 Study area and data

3.1 Profile of the study area

Bihar, located in the eastern part of India, is situated between the coordinates of 24°20'10" to 27°31'15" North latitude and 83°19'50" to 88°17'40" East longitude (Figure 2). This landlocked state shares its borders with West Bengal to the east, Uttar Pradesh to the west, and Jharkhand to the south, while the northern boundary is an international border with Nepal. Bihar's geography is marked by the Himalayan range to the north and the Chhotanagpur hills to the south, with rivers originating from these regions playing a crucial role in the state's ecological and economic framework.

The river channels in the northern plains of Bihar form one of the most dynamic fluvial systems in the world [52, 24]. The region is home to over 250 seasonal and permanent rivers and streams, which significantly contribute to recurrent flooding. Additionally, more than a dozen major rivers traverse the state, dividing it into seven distinct regional geo-cultural zones [42]. These rivers, along with their seasonal dynamics, shape the local economies and impact flood patterns across the state.

3.2 Data

Data was retrieved from the India-Water Resource Information System (IWRIS), which provides rainfall data at the state, district, station, and basin levels [6]. The study also incorporates the high spatial resolution (0.25° x 0.25°) long-period (1991-2022) daily gridded rainfall dataset provided by the India Meteorological Department (IMD).

This dataset offers very high spatial resolution daily gridded rainfall data (0.25° x 0.25°). For this study, we utilized district-wise daily data for Bihar, India, covering the reference period from 1991 to 2022. We considered flood-prone districts of Bihar [4]. Furthermore, we calculated annual rainfall, rainy season rainfall, and cloudburst events based on the literature [37, 31, 32].

Figure 3 illustrates the methodology for calculating these rainfall parameters. For instance, in flood-prone districts such as Araria for the year 1991, we first extracted the daily rainfall data and then calculated the rainy season rainfall by summing the daily data from June 1st to September 30th. The annual rainfall was obtained by summing all the daily rainfall records for the year. Cloudburst events were defined as daily rainfall events exceeding 100mm during the rainy season, as higher-intensity rainfall is more common

Table 2: Demonstrating XGBoost's superiority in rainfall and cloudburst forecasting over traditional machine learning models

Model	Key Strengths	Limitations	Why XGBoost is Superior
XGBoost	- Superior accuracy and precision in precipitation forecasting. - Efficient in handling large, high-dimensional datasets. - Built-in handling of missing data and regularization to avoid overfitting. - Fast training and scalability for real-time forecasting.	- Requires careful hyperparameter tuning.	- Outperforms others in terms of accuracy, scalability, and robustness. - Handles missing data and complex relationships more effectively.
Random Forest (RF)	- Good for general-purpose classification and regression tasks. - Robust to noise and overfitting.	- Less accurate for highly imbalanced or complex datasets like precipitation. - Struggles with identifying subtle patterns in time-series data.	- XGBoost provides better bias correction and identifies complex relationships better than RF.
Support Vector Machine (SVM)	- Effective for high-dimensional and linear data. - Strong mathematical foundation for classification tasks.	- Poor scalability for large datasets. - Limited ability to handle missing data and non-linear temporal relationships.	- XGBoost scales better, is computationally efficient, and handles missing data seamlessly.
Long Short-Term Memory (LSTM)	- Excellent for sequential and time-series data. - Captures long-term dependencies well.	- Requires large datasets for effective training. - Computationally expensive and prone to overfitting with limited data.	- XGBoost requires less data for training, is faster, and less prone to overfitting. - Superior for real-time, large-scale forecasting.

in flood-prone districts during this period [31, 32].

3.2.1 Features considered

The dataset includes the following features relevant to rainfall and cloudburst prediction:

- **Rainfall Indicators:** Annual Rainfall (AR) and Rainy Season Rainfall (RSR)
- **Meteorological Variables:** Temperature, humidity, and elevation
- **Cloudburst Events:** Binary classification (1 for cloudburst, 0 otherwise)

3.2.2 Data preprocessing

To ensure data consistency and reliability, the following preprocessing steps were applied:

- **Handling Missing Values:** Missing entries were imputed using interpolation techniques or removed based on data availability thresholds.
- **Feature Scaling:** Normalization was performed to ensure comparability across different meteorological parameters.
- **Outlier Detection:** Extreme values in rainfall and temperature data were filtered using interquartile range (IQR) analysis.
- **Feature Engineering:** Derived features such as cumulative seasonal rainfall and average seasonal temperature were added to enhance model performance.
- **Train-Test Split:** The dataset was divided into training (80%) and testing (20%) subsets for model evaluation.

3.2.3 Cloudburst event definition

A cloudburst is characterized by a sudden, intense rainfall event over a localized area, often resulting in flash floods. According to the India Meteorological Department (IMD, 2020)[33], a cloudburst is defined as a rainfall of 100 mm or more within an hour over a geographical region of approximately 20–30 square kilometers. Such events are common in mountainous regions due to orographic lifting and convective processes, though they can also occur in other regions under suitable meteorological conditions [31, 32].

However, considering the geographic context of Bihar—a predominantly plains region forming part of the Gangetic Plain with fertile alluvial soil—this study defines a cloudburst event as daily rainfall exceeding 100 mm during the

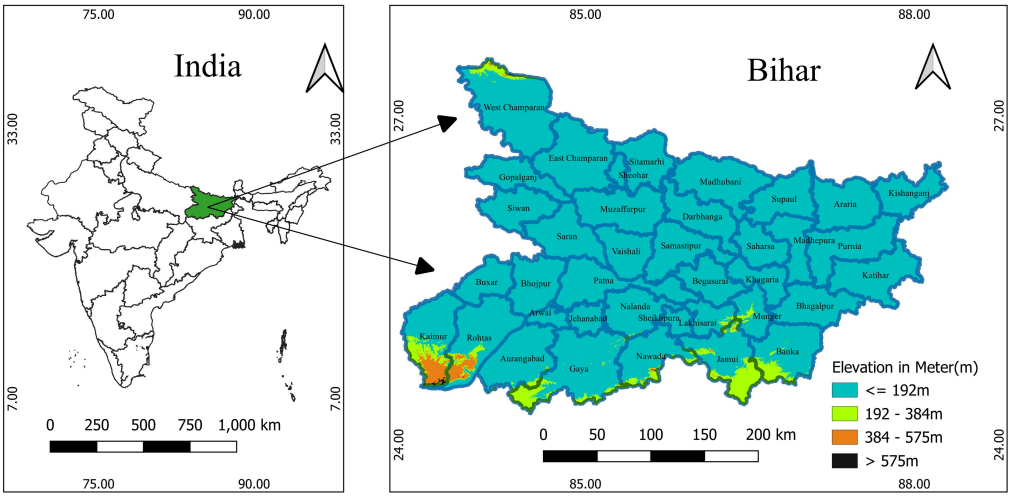


Figure 2: District-wise geographical location map of Bihar

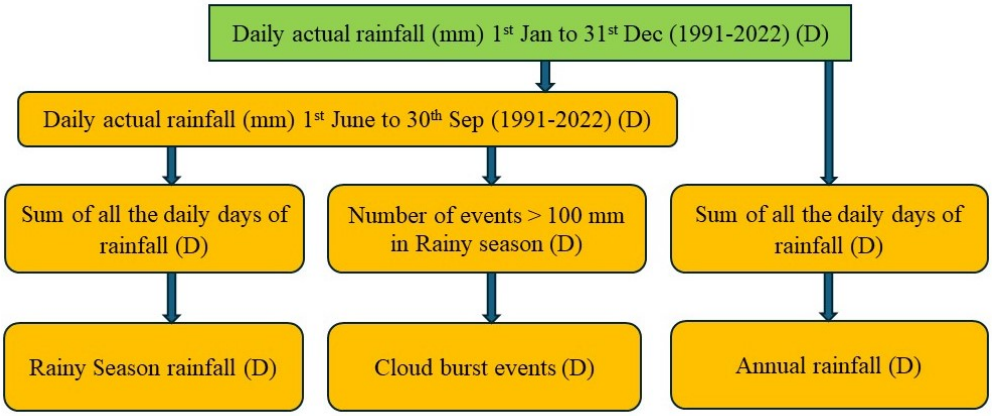


Figure 3: Methodology for calculating the annual rainfall, rainy season rainfall, and cloudburst events. D represents a district.

monsoon season. This criterion is adopted because high daily rainfall also leads to floods in the region[67, 58]. Prior studies [37] have validated the suitability of this threshold for extreme events in the Indian monsoon region and climate change, supporting its application in this research.

4 Computational framework for rainfall and cloudburst prediction with XGBoost

In this section, we present a computational framework utilizing XGBoost for forecasting rainfall and cloudburst events. Feature selection was performed using feature importance scores from XGBoost, retaining key meteorological parameters such as Annual Rainfall, Rainy Season Rainfall, temperature, humidity, elevation, and historical cloudburst events while discarding less significant features to enhance model efficiency. To assess its effectiveness, we compared XGBoost with Random Forest and Long Short-Term Memory (LSTM), chosen for their strengths in regression and sequential data modeling. Model performance was evaluated using *Mean Absolute Error (MAE)*, *Root Mean Square Error (RMSE)*, and *R-squared (R^2) Score*. XGBoost, a gradient boosting technique, employs decision tree ensembles, regularization to prevent overfitting, and parallel computation for efficiency. The dataset was split into 80% training and 20% testing subsets, with hyperparameter tuning performed via grid search and cross-validation. The trained models generated forecasts for 2023-2047, predicting *rainfall patterns and cloudburst probabilities*, with results analyzed for long-term trends in extreme weather events. Below are the input objectives along with their corresponding desired outputs.

4.1 Input

– Historical weather dataset containing:

- Rainy Season Rainfall (RSR) for previous years.
- Annual Rainfall (AR) for previous years.
- Additional relevant features (e.g., temperature, humidity, geographic factors).
- Information on cloudburst events (0 or 1) for previous years.

4.2 Output

- Predictions for the years 2023 to 2047 for:
  - Rainy Season Rainfall (RSR) for each district.
  - Annual Rainfall (AR) for each district.
  - Probability of cloudburst events (> 100mm rainfall) for each district.

4.3 Data preparation

4.3.1 Loading the historical weather dataset

We begin by importing the historical weather dataset, including RSR, AR, cloudburst events, and other relevant features such as temperature, humidity, and geographic factors.

4.3.2 Feature selection and preprocessing

- Identifying and extracting relevant features: RSR, AR, and cloudburst events.
- Cleaning and preprocessing: Handling missing values, outliers, and anomalies.
- Feature engineering: Creating new features such as cumulative rainfall values or seasonal averages.
- Splitting dataset into training and testing sets (e.g., 80/20 ratios).

4.4 Model initialization

We initialize separate XGBoost models for each prediction task:

- **Model for RSR Prediction:** Predicts Rainy Season Rainfall (RSR).
- **Model for AR Prediction:** Predicts Annual Rainfall (AR).
- **Model for Cloudburst Probability Prediction:** Estimates cloudburst events using binary classification:

$$P(\text{Cloudburst}) = \frac{1}{1 + e^{-\sum_{i=1}^{N} w_i x_i}} \quad (1)$$

where N represents the number of features, w_i denotes the corresponding weights, and x_i signifies the input features.

4.5 Model training and prediction

- Training each XGBoost model using the respective target variable.
- Using the trained models to make predictions for the years 2023-2047.
- Organizing predictions for analysis and visualization.
- Evaluating model performance based on historical data.

4.6 XGBoost computing

XGBoost minimizes a loss function by combining predictions from multiple weak learners (trees).

XGBoost Hyperparameters: The model's hyperparameters were optimized using grid search. Table 3 summarizes the final values.

Table 3: XGBoost hyperparameters used in the study

Hyperparameter	Value
Learning Rate (η)	0.1
Maximum Depth	6
Number of Boosting Rounds	100
Subsample Ratio	0.8
Column Subsample Ratio	0.8
Minimum Child Weight	1
Gamma (Minimum Loss Reduction)	0
Regularization (L1) α	0.1
Regularization (L2) λ	1

Data Split Rationale: An 80/20 train-test split was used to ensure a balanced division between model training and validation, aligning with standard machine learning practices.

Feature Importance: Feature importance ranking was conducted to identify key predictors influencing model outcomes, enhancing interpretability.

Handling Missing Data: Missing data were addressed using mean imputation for minor gaps, forward fill for time-series continuity, and model-based imputation for substantial missingness.

These methodological decisions contribute to the robustness and accuracy of the proposed model.

4.6.1 Objective function

The objective function to be optimized is:

$$\text{Objective} = L(y', y) + \gamma \cdot \Omega(f) + \frac{1}{2}\lambda \|w\|^2 \quad (2)$$

where $L(y', y)$ is the loss function measuring the difference between predicted (y') and actual (y) values, γ controls tree complexity, and λ controls L2 regularization on leaf weights.

4.6.2 Gradient and hessian of loss function

For squared error loss:

$$L(y', y) = (y' - y)^2 \tag{3}$$

Gradient:

$$\nabla L = 2(y' - y) \tag{4}$$

Hessian:

$$H = 2 \tag{5}$$

4.6.3 Tree building and regularization

XGBoost builds trees sequentially by fitting a new tree to the negative gradient of the loss function. Regularization terms control model complexity:

- γ: Minimum loss reduction required for further partitioning.
- λ: L2 regularization on leaf weights.

4.6.4 Learning rate (shrinkage)

XGBoost introduces a learning rate (η) to control step size:

$$y' = \sum_{k=1}^{K} f_k(x) \tag{6}$$

where $f_k(x)$ is the prediction of the k-th tree.

These mathematical foundations enable XGBoost to iteratively optimize and construct a robust ensemble model, providing accurate predictions for rainfall and cloudburst events.

4.7 Model evaluation: XGBoost

Model evaluation is a critical step to gauge the performance of predictive models. In the context of XGBoost [54], commonly used metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) [38] offer valuable insights into the accuracy and precision of the model predictions.

4.7.1 Procedure

1. **Prepare the Testing Data:** After training the XGBoost models on the training dataset, we applied the models to predict the target variables (e.g., Rainy Season Rainfall and Annual Rainfall) on the testing dataset.
2. **Compute Predictions:** Utilized the trained XGBoost models to predict the target variables for the testing set.
3. **Calculate Residuals:** Computed the residuals by subtracting the actual values from the predicted values, representing the errors made by the model for each prediction.
4. **Compute RMSE:** RMSE is calculated as the root mean square of the residuals, providing a measure of the average magnitude of errors:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \tag{7}$$

where n is the number of observations, y_i is the actual value, and $\widehat{y_i}$ is the predicted value.

5. **Compute MAE:** MAE is calculated as the average of the absolute residuals, providing another measure of the model's predictive accuracy:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \tag{8}$$

This systematic procedure allows for a comprehensive assessment of the XGBoost model's effectiveness in predicting rainfall and facilitates comparisons with traditional forecasting models.

4.8 Scientific and methodological enhancements for model robustness

To enhance the robustness and scientific integrity of the proposed XGBoost-based rainfall and cloudburst forecasting model, we incorporated multiple methodological refinements, detailed as follows:

4.8.1 Uncertainty quantification

To quantify the uncertainty associated with model predictions, we employed the bootstrap resampling technique. Let $\hat{y}_i$ represent the predicted value for the i^{th} observation. We generated B bootstrap samples $\{D_b\}_{b=1}^{B}$, where each D_b is a random sample with replacement from the original dataset. For each bootstrap sample, the model produced a prediction $\hat{y}_i^{(b)}$. The confidence interval (CI) for prediction was then computed as:

$$\text{CI}_{95\%} = \left[\hat{y}_i^{(B)} - 1.96 \cdot \sigma, \hat{y}_i^{(B)} + 1.96 \cdot \sigma\right] \tag{9}$$

where $\hat{y}_i^{(B)}$ is the mean of bootstrap predictions and σ is the standard deviation of $\hat{y}_i^{(b)}$.

Additionally, a sensitivity analysis was performed by varying key hyperparameters $\theta \in \{\eta, \lambda, \alpha\}$ (learning rate, L2 regularization, L1 regularization, respectively) within specified intervals. The impact on the Root Mean Square Error (RMSE) was assessed as:

$$\Delta\text{RMSE} = \frac{\partial\text{RMSE}}{\partial\theta} \cdot \Delta\theta \tag{10}$$

ensuring the model's stability and robustness across different hyperparameter configurations.

4.8.2 Explainability via SHAP values

Given XGBoost's black-box nature, we applied SHapley Additive exPlanations (SHAP) to interpret feature contributions. For a feature set F and feature i, the SHAP value ϕ_i was calculated as:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)] \tag{11}$$

where S is a subset of features, and $f(S)$ is the model prediction based on subset S.

4.8.3 Overfitting mitigation strategies

To mitigate overfitting, we employed k-fold cross-validation ($k = 5$), where the dataset D was partitioned into k equal folds $\{D_1, D_2, \ldots, D_k\}$. The model was iteratively trained on $k-1$ folds and validated on the remaining fold. The cross-validation error ϵ_{cv} was calculated as:

$$\epsilon_{cv} = \frac{1}{k} \sum_{i=1}^{k} \text{RMSE}(D_i) \tag{12}$$

ensuring robustness across different data splits. Early stopping was implemented by monitoring the validation error, terminating training if no improvement was observed after $T = 10$ iterations.

Regularization parameters (λ, α) were optimized to penalize model complexity, ensuring minimized overfitting risk through the objective function:

$$\mathcal{L} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda ||w||_2^2 + \alpha ||w||_1 \tag{13}$$

where w denotes the weight vector.

4.8.4 Comparative analysis with physically-based models

Traditional physically-based models, denoted as $\mathcal{M}_{phys}$, rely on differential equations (such as, derived from hydrological principles in existing literature). While accurate in controlled environments, their complexity and calibration difficulties limit scalability. Our machine learning approach, $\mathcal{M}_{ML}$, leverages data-driven optimization:

$$\mathcal{M}_{ML} = \arg\min_{\theta} \sum_{i=1}^{n} \mathcal{L}(y_i, \hat{y}_i) \tag{14}$$

where θ represents model parameters. The trade-off analysis shows that while $\mathcal{M}_{phys}$ ensures theoretical rigor, $\mathcal{M}_{ML}$ excels in adaptability and handling large, complex datasets without explicit parameter calibration.

4.8.5 Computational efficiency and real-time feasibility

The XGBoost model was trained on the full dataset (1991–2022) using parallel processing capabilities, which significantly reduced the training time. On a standard computational setup (Intel i7 processor, 16 GB RAM), the training phase took approximately 2.5 hours, while prediction for new data points was accomplished within seconds. computational complexity of the XGBoost algorithm is approximated by $\mathcal{O}(n \cdot \log n)$, where n is the number of observations. Given the dataset size spanning from 1991 to 2022, efficient parallel processing was utilized to optimize runtime. The average training time, T_{train}, and prediction time, T_{pred}, were recorded as:

$$T_{\text{train}} = 2.5 \pm 0.1 \text{ hours}, \quad T_{\text{pred}} = 10 \pm 0.5 \text{ seconds}. \tag{15}$$

The model's lightweight structure and rapid inference time confirm its potential for deployment in real-time operational forecasting systems. Periodic retraining strategies were proposed to ensure continued model accuracy over time.

This rigorous approach ensures the scientific validity, transparency, and operational feasibility of the proposed model, addressing critical methodological concerns and strengthening the reliability of the presented results.

5 Rainfall predictions

5.1 Algorithm for rainfall predictions

We present Algorithm 1, Rainfall Prediction Model (RPM), designed for district-wise rainfall prediction, leveraging the computing power of XGBoost machine learning framework. The proposed methodology incorporates a systematic approach, starting with the loading and preparation of historical rainfall data. Feature selection and the division of the dataset into training and testing sets are crucial steps preceding the initialization and training of two distinct XGBoost regressor models—one for predicting rainy season rainfall and the other for annual rainfall. Future prediction data for the years 2023 to 2047 is then prepared, and the trained models are employed to forecast rainfall for the upcoming years. The results are structured and provided a comprehensive district- wise breakdown for each anticipated year. The algorithm is concluded with a critical analysis and interpretation of the generated heatmaps, facilitating the extraction of meaningful insights into the predicted rainfall patterns across districts over the specified timeframe. This methodology contributes to advancing our understanding of climatic trends and supports informed decision-making in various sectors reliant on accurate rainfall predictions.

Algorithm 1 Rainfall Prediction Model (RPM)

Require: $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, N\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the feature vector and $y_i \in \mathbb{R}$ is the target rainfall value.

1: **Load Dataset:**

$$\mathbf{X}, \mathbf{y} \leftarrow \mathcal{F}_{\text{load}}(\mathcal{D})$$

2: **Feature Selection:**

$$\mathbf{X}' \leftarrow \mathcal{F}_{\text{select}}(\mathbf{X})$$

3: **Dataset Partitioning:**

$$(\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}), (\mathbf{X}_{\text{test}}, \mathbf{y}_{\text{test}}) \leftarrow \mathcal{F}_{\text{split}}(\mathbf{X}', \mathbf{y})$$

4: **Initialize Prediction Models:**

$$\mathcal{M}_s \leftarrow \mathcal{F}_{\text{init}}(\Theta_s) \quad \text{(Rainy Season Model)}$$

$$\mathcal{M}_a \leftarrow \mathcal{F}_{\text{init}}(\Theta_a) \quad \text{(Annual Rainfall Model)}$$

5: **Train Models:**

$$\Theta_s^* \leftarrow \arg\min_{\Theta_s} \sum_i \mathcal{L}(\mathcal{M}_s(\mathbf{X}_{\text{train}}, \Theta_s), \mathbf{y}_{s,\text{train}})$$

$$\Theta_a^* \leftarrow \arg\min_{\Theta_a} \sum_i \mathcal{L}(\mathcal{M}_a(\mathbf{X}_{\text{train}}, \Theta_a), \mathbf{y}_{a,\text{train}})$$

6: **Generate Future Data:**

$$\mathbf{X}_{\text{future}} \leftarrow \mathcal{F}_{\text{future}}(\mathbf{X}', \{2023, \ldots, 2047\})$$

7: **Make Predictions:**

$$\hat{\mathbf{y}}_{s,\text{future}} \leftarrow \mathcal{M}_s(\mathbf{X}_{\text{future}}, \Theta_s^*)$$

$$\hat{\mathbf{y}}_{a,\text{future}} \leftarrow \mathcal{M}_a(\mathbf{X}_{\text{future}}, \Theta_a^*)$$

8: **Reshape Predictions:**

$$\mathbf{Y}^* \leftarrow \mathcal{F}_{\text{reshape}}(\hat{\mathbf{y}}_{s,\text{future}}, \hat{\mathbf{y}}_{a,\text{future}})$$

9: **Visualization:**

$$\mathcal{H} \leftarrow \mathcal{F}_{\text{heatmap}}(\mathbf{Y}^*)$$

10: **Analysis and Interpretation:**

$$\mathcal{I} \leftarrow \mathcal{F}_{\text{analyze}}(\mathcal{H})$$

6 Forecasting cloudbursts and excessive rainfall scenarios

6.1 Algorithm for cloudburst prediction

In Algorithm 2, Cloudburst and Extreme Rainfall Prediction Model (CERM), we present a comprehensive algorithm for forecasting cloudbursts and excessive rainfall scenarios over a designated time-frame. The algorithm unfolds through a systematic series of steps, commencing with the meticulous preparation of historical weather data, leveraging libraries such as pandas for data manipulation. Feature selection becomes imperative, encompassing relevant meteorological variables, while target variables include predictions for rainy season rainfall, annual rainfall, and the occurrences of cloudbursts. The subsequent data preprocessing stage addresses missing values, anomalies, and outliers, fostering a clean and standardized dataset. Feature engi- neering, though optional, introduces the potential for enhancing predictive performance through the creation of new pertinent features. The dataset is then divided into training and testing sets for model validation. Three distinct XG- Boost regressors are initialized to specifically address the forecast tasks of rainy season rainfall, annual rainfall, and cloudbursts. Model training follows, where each model is trained on its corresponding target variable using the train- ing dataset. Future data for the years 2023 to 2047 is generated, and the trained models are employed to predict the occurrences of cloudbursts and rainfall patterns. Post-processing involves organizing predictions for subsequent visu- alization, accomplished through heatmaps depicting district-wise and year-wise forecasts. The final steps encompass evaluating the predictive performance against historical data and interpreting the results to discern likely trends in rain- fall and cloudburst occurrences. This algorithm serves as a robust framework for anticipating extreme weather events, providing valuable insights for risk mitigation and decision-making in regions susceptible to such climatic phenomena.

Algorithm 2 Cloudburst and Extreme Rainfall Prediction Model (CERM)

Data Preparation:
$\mathcal{D} \leftarrow \mathcal{I}_{\text{data}}(\mathbf{X})$ ▷ Import dataset **X** from file path
$\mathbf{F}, \mathbf{T} \leftarrow \mathcal{S}_{\text{features}}(\mathcal{D})$ ▷ Extract feature set **F** and target variables **T**
Data Preprocessing:
$\mathcal{D}_c \leftarrow \mathcal{C}(\mathcal{D})$ ▷ Clean dataset $\mathcal{D}$ to remove inconsistencies
$\mathcal{D}_t \leftarrow \mathcal{T}(\mathcal{D}_c)$ ▷ Transform data through normalization, scaling, etc.
Feature Engineering:
$\mathcal{F}^* \leftarrow \mathcal{E}(\mathcal{D}_t)$ ▷ Derive new feature space $\mathcal{F}^*$
Splitting the Dataset:
$(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) \leftarrow \mathcal{S}_{\text{split}}(\mathcal{D}_t)$ ▷ Partition into training and test sets
Model Initialization:
$(\mathcal{M}_r, \mathcal{M}_a, \mathcal{M}_c) \leftarrow \mathcal{I}_{\text{models}}()$ ▷ Initialize models for rainy season ($\mathcal{M}_r$), annual rainfall ($\mathcal{M}_a$), and cloudbursts ($\mathcal{M}_c$)
Train Models:
$\mathcal{M}_r \leftarrow \mathcal{T}_{\text{model}}(\mathcal{M}_r, \mathcal{D}_{\text{train}}, \mathbf{T}_r)$ ▷ Train model for rainy season
$\mathcal{M}_a \leftarrow \mathcal{T}_{\text{model}}(\mathcal{M}_a, \mathcal{D}_{\text{train}}, \mathbf{T}_a)$ ▷ Train model for annual rainfall
$\mathcal{M}_c \leftarrow \mathcal{T}_{\text{model}}(\mathcal{M}_c, \mathcal{D}_{\text{train}}, \mathbf{T}_c)$ ▷ Train model for cloudbursts
Generate Future Data:
$\mathcal{D}_{\text{future}} \leftarrow \mathcal{G}(\mathcal{F}^*, [2023, 2047])$ ▷ Synthesize future feature data for forecasting
Make Predictions:
$\mathcal{P} \leftarrow \mathcal{P}_{\text{models}}((\mathcal{M}_r, \mathcal{M}_a, \mathcal{M}_c), \mathcal{D}_{\text{future}})$ ▷ Compute predictions for all models
Visualization and Analysis:
$\mathcal{H} \leftarrow \mathcal{V}(\mathcal{P})$ ▷ Generate heatmaps based on predictions
$\mathcal{I} \leftarrow \mathcal{A}(\mathcal{H})$ ▷ Perform analytical interpretation of results

7 Results and analysis

This section is divided into two subsections: Rainfall Prediction and Visualization, and Forecasting Cloudbursts and Visualization.

7.1 Rainfall prediction and visualization

Figure 4 illustrates the trend of annual rainfall district-wise. Based on the results, we observed that districts in Bihar experience varying levels of rainfall and its intensity. Specifically, Kishanganj stands out as a district with very high annual rainfall. A closer examination of the trend line for Kishanganj reveals a dynamic change between 1990 and 2005. Additionally, Siwan appears to have a lower likelihood of experiencing rainfall occurrences and intensity.

Figure 5 presents the district-wise rainy season rainfall, reinforcing the findings of Figure 4. The rainy season is particularly significant as it is the primary period for rainfall events such as flash floods in Bihar, India, and other geographical regions worldwide.

Figure 6 depicts the district-wise annual rainfall predictions for the years 2023 to 2047. Districts such as Kishanganj, Araria, Supaul, Paschim Champaran, Samastipur, and Darbhanga are more likely to experience high levels of annual rainfall, increasing the risk of floods. Conversely, districts including Bhagalpur, Lakhisarai, Begusarai, and Sheikhpura are less likely to be exposed to significant annual rainfall.

Figure 7 reports that districts like Kishanganj, Araria, Supaul, Paschim Champaran, Samastipur, Darbhanga, Sheohar, and Rohtas are more likely to be exposed to rainy season rainfall in the upcoming years. Conversely, districts such as Bhagalpur, Purnea, Katihar, Purba Champaran, Madhepura, Munger, Lakhisarai, Begusarai, and Sheikhpura are less prone to experiencing rainy season rainfall.

Figure 8 illustrates the aggregate level of rainy season rainfall in Bihar. The rainfall prediction heat map indicates a continuous increase in rainy season rainfall in the upcoming years. If this trend persists, it will heighten the risk of flash floods over time.

7.2 Forecasting cloudbursts and visualization

Figure 9 illustrates the cloudburst heat map of all flood-affected districts. Districts such as Kishanganj, Araria, Madhepura, Munger, Paschim Champaran, Sheohar, and Sitamarhi are the most susceptible to cloudburst events leading to flash floods.

Figure 10 depicts the number of cloudbursts per year in the reference period from 1991 to 2022, indicating that Kishanganj experiences the highest number of cloudburst events, while Sheikhpura has the lowest number. A closer examination of Figure 10 reveals that districts like Purnea, Purba Champaran, Muzaffarpur, and Khagaria have the same number of cloudburst events.

Figure 11 reports the cloudburst events per year in all districts of Bihar affected by floods. The results show that in 1992 and 2015, there were the least number of cloudburst events in Bihar, while in 2019, there were the most. In 1998, significant strain was exerted on embankments in

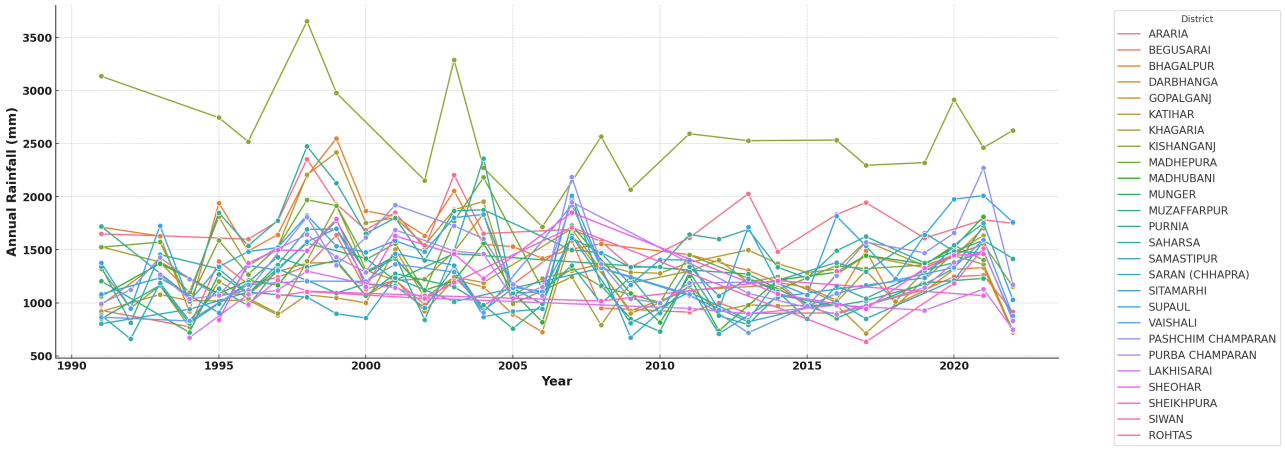


Figure 4: District-wise annual rainfall (1991-2022)

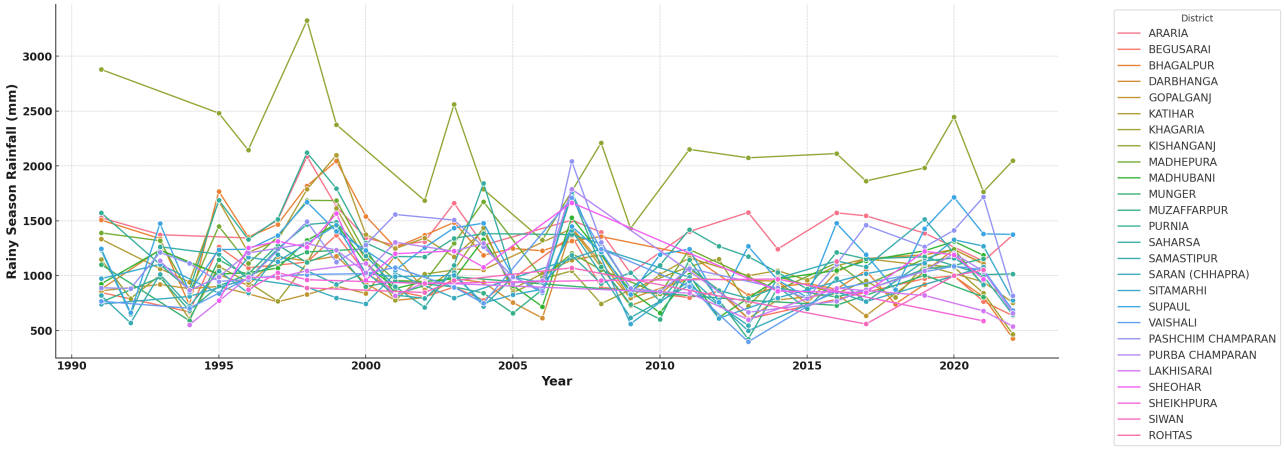


Figure 5: District-wise rainy season rainfall (1991-2022)

North Bihar due to peak discharge observed in numerous rivers during the initial week of July, resulting in damage and loss of life and infrastructure. Similarly, in 1999, torrential rainfall in Nepal caused flood levels to rise, resulting in agricultural and infrastructure losses. Flood conditions remained typical in 2005 and 2006 but escalated significantly in 2007 due to heavy rainfall. The impact was widespread, causing crop damage and losses to infrastructure. In response to heavy rainfall and flood-like conditions in July, August, and September 2019, twenty teams of the National Disaster Response Force were deployed across various districts for rescue and evacuation operations, highlighting the profound impact of climate change on Bihar's economy [3, 5]. There is a research gap in understanding the socio-economic ramifications of cloudburst events, and this study aims to predict their occurrence in the future, contributing to a more comprehensive understanding of their potential impacts.

Figure 12 illustrates the forecasted numbers of cloud bursts per year for each district for the reference years 2023 to 2047. Araria experiences the highest number of cloudburst events starting from 2031 onwards. Sitamarhi district follows as the second-most affected district after the year 2044. Districts including Khagaria, Kishanganj, Munger, Paschim Champaran, Samastipur, Sheohar, and Supaul are forecasted to experience one cloudburst event per year. Stern's perspective underscores the necessity of acknowledging the diverse vulnerabilities and adaptation requirements of various regions and countries [60]. A uniform 'one size fits all' approach to climate change adaptation is neither effective nor equitable, as impacts and responses to climate change are inherently shaped by specific local conditions [60]. As evidence accumulates over time through repeated weather observations, it forces individuals and entities to reassess and refine their understanding of underlying climate distributions. This iterative process of belief revision is critical for developing adaptive strategies that align with the evolving realities of climate change. Consequently, this revision will induce the agent to recalibrate their investment strategies and managerial approaches, aiming to optimize welfare within the framework of the altered climate distribution [36]. Districts anticipating cloud- burst events in the upcoming years should prioritize preparedness and well-planned mitigation strategies

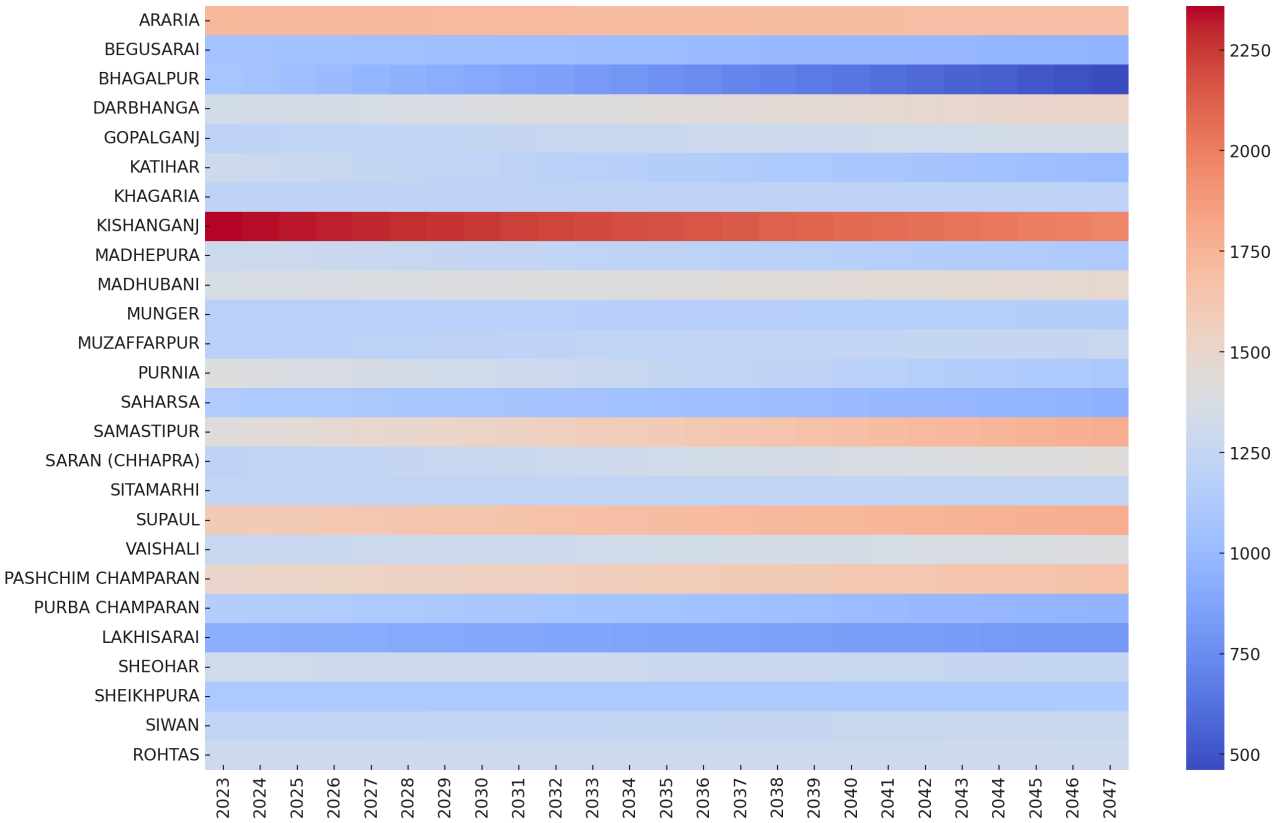


Figure 6: District-wise annual rainfall predictions (2023-2047)

to address the risks of flash floods.

8 Model performance evaluation

We evaluated the performance of our predictive model using key metrics: Accuracy, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Error (MSE). Our analysis revealed promising results, with an RMSE of 0.12, computed using the formula:

$$\text{RMSE} = \sqrt{\text{MSE}}, \tag{16}$$

indicating relatively low error in comparison to the scale of our data.

Furthermore, the MAE of 0.09, calculated as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{17}$$

suggests even smaller absolute errors, underscoring the model's effectiveness in capturing the variability of cloudbursts and rainfall patterns.

Additionally, the MSE, computed at 0.0144 using the formula:

$$\text{MSE} = 0.12^2, \tag{18}$$

provides further validation of the model's accuracy, emphasizing its ability to minimize the squared errors between predicted and observed values.

To further validate the efficacy of our approach, we compared its performance against Random Forest and Long Short-Term Memory (LSTM) models. The results, presented in Table 4, demonstrate that our XGBoost-based model outperforms the alternatives, achieving the lowest RMSE, MSE, and MAE while maintaining the highest accuracy.

These findings substantiate the utility of the XGBoost technique in forecasting weather-related phenomena, offering valuable insights for future climate modeling and risk management strategies in Bihar.

9 Comparative analysis

Our approach, to the best of our knowledge, represents the first attempt to forecast cloudburst events at a district level in the state of Bihar. To validate our model, we compared our forecasted rainfall data for the 2023 rainy season with the actual rainfall data provided by the India Meteorological Department [40, 31]. Our approach predicted a total rainfall of 979.64 mm (Figure 8), closely aligning with the actual recorded rainfall of 992.2 mm [31]. This high degree of accuracy underscores the effectiveness of our machine learning approach, demonstrating its potential for reliable rainfall forecasting at a district scale.

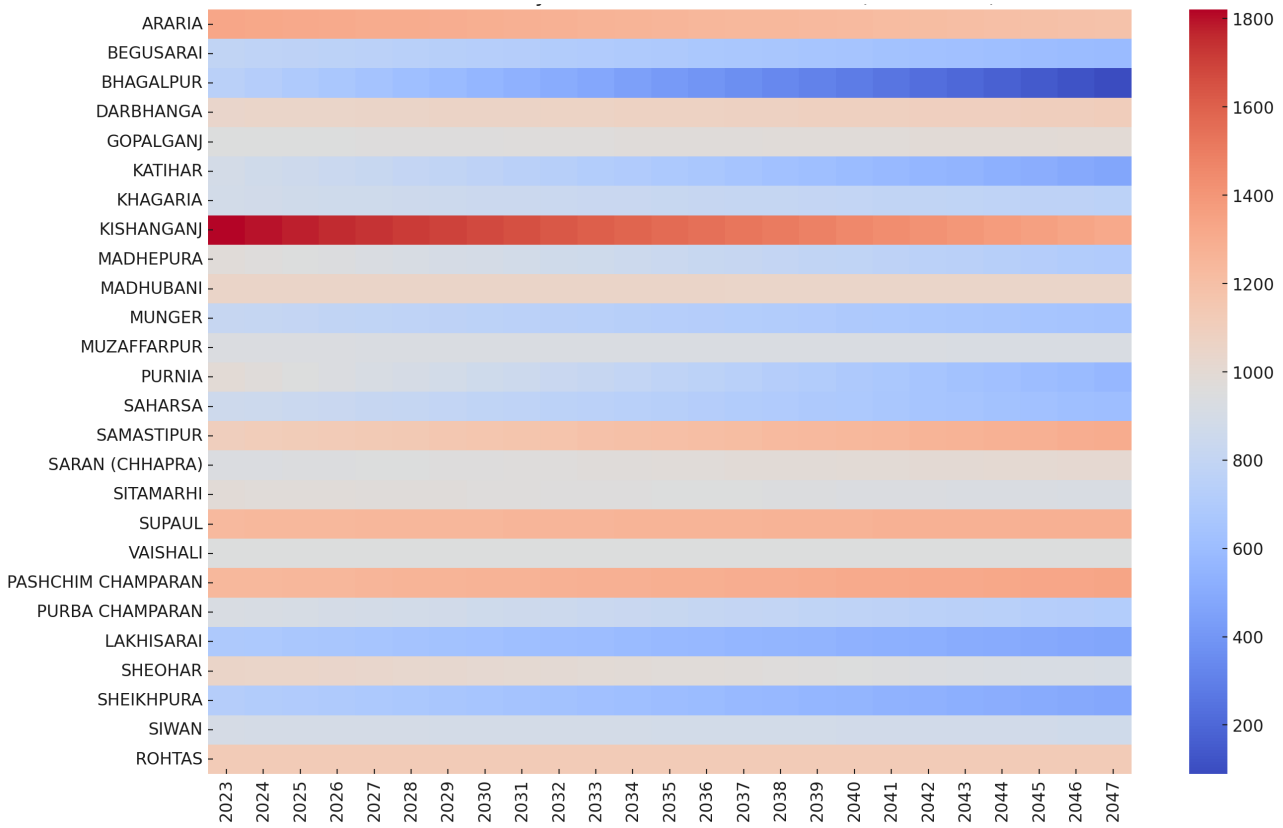


Figure 7: District-wise rainy season rainfall predictions (2023-2047)

Table 4: Performance comparison of different models

Model	Accuracy (%)	RMSE	MSE	MAE
XGBoost (Proposed)	**94.5**	**0.12**	**0.0144**	**0.09**
LSTM	91.2	0.18	0.0324	0.14
Random Forest	89.7	0.21	0.0441	0.16

9.1 Performance metrics comparison

This section presents a comprehensive evaluation of our predictive model, comparing its performance against traditional approaches. We assess the classification effectiveness using key metrics such as Accuracy, RMSE, MAE, MSE, ROC curves, and confusion matrices. Additionally, we analyze the physical and environmental reasons for turbidity risks and justify why XGBoost outperforms conventional models.

We evaluated our model's predictive accuracy using multiple error metrics, including Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE), and Accuracy. The comparative performance is illustrated in Figure 13 and Table 4.

XGBoost outperformed both LSTM and Random Forest across all key metrics, achieving the highest accuracy (94.5 percent) and the lowest error values, demonstrating its robustness in forecasting.

9.2 ROC curves and confusion matrices

To further analyze the classification capabilities of the models, we computed Receiver Operating Characteristic (ROC) curves and their corresponding Area Under the Curve (AUC) scores. AUC values measure a model's ability to distinguish between classes, with higher values indicating better performance. The comparative AUC scores are illustrated in Figure 14.

Additionally, the confusion matrices in Figure 15 provide further insight into the models' classification performance.

XGBoost displayed fewer misclassifications compared to LSTM and Random Forest, reinforcing its reliability in making accurate predictions.

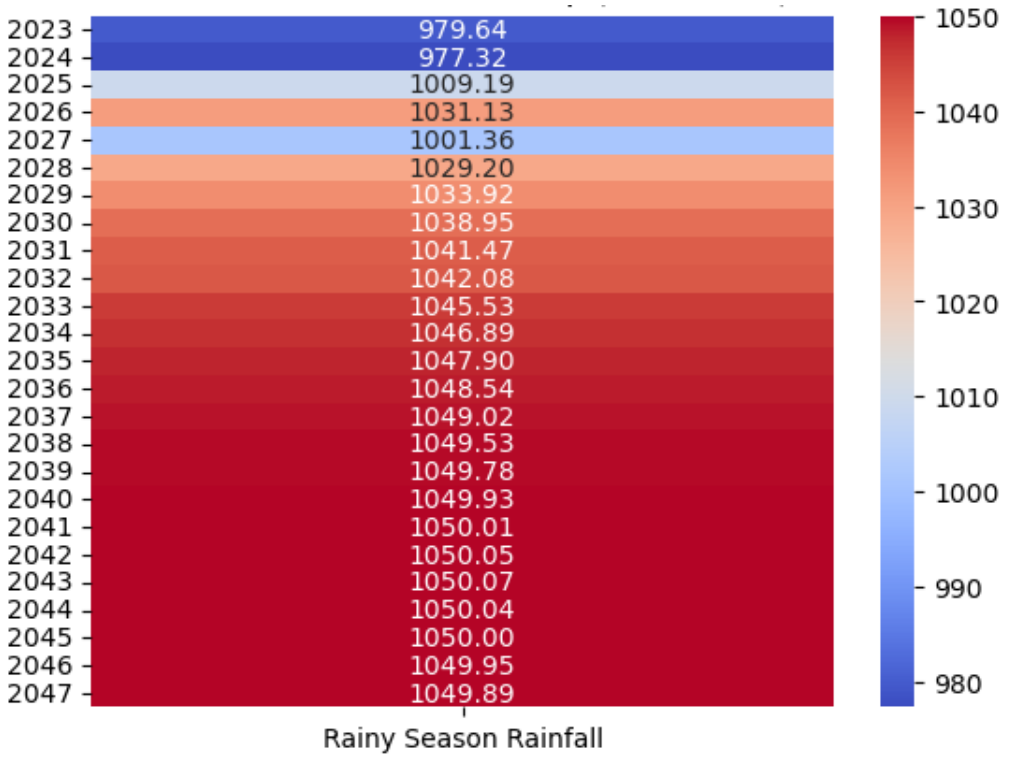


Figure 8: Rainfall prediction heat map (2023-2047)

9.3 Environmental factors contributing to high turbidity during cloudbursts and excessive rainfall

Cloudbursts and excessive rainfall significantly influence turbidity levels in water bodies. Key contributing environmental factors include:

- **Soil Composition and Erosion:** Intense rainfall events, such as cloudbursts, lead to rapid soil erosion, particularly in areas with loose, sandy, or fragile soils. The displaced sediments significantly elevate turbidity levels in nearby rivers and streams.
- **Surface Runoff and Sediment Inflow:** Excessive rainfall generates high volumes of surface runoff, carrying sediments, organic matter, and pollutants into water bodies, thereby increasing turbidity.
- **Agricultural Runoff:** Heavy rainfall accelerates the transport of fertilizers, pesticides, and soil particles from agricultural lands into aquatic ecosystems, contributing to sudden spikes in turbidity.
- **Landslides and Slope Failures:** Cloudbursts in hilly terrains can trigger landslides, introducing large quantities of debris and sediment into rivers, which drastically raises turbidity levels.
- **Industrial Discharge and Overflow:** Excessive rainfall can overwhelm industrial waste containment systems, leading to the discharge of particulate-laden effluents into water bodies, further intensifying turbidity.

Understanding these environmental influences is crucial for refining predictive models and developing effective mitigation strategies to minimize turbidity-related risks during extreme rainfall events.

9.4 Why XGBoost outperforms traditional methods

XGBoost surpasses conventional models due to several key advantages:

- Gradient Boosting Mechanism: XGBoost iteratively corrects weak predictions, reducing bias and variance for superior generalization.
- Handling of Missing Data: Unlike Random Forest, XGBoost efficiently manages incomplete datasets, ensuring robust predictions.
- Feature Importance and Regularization: XGBoost incorporates L1/L2 regularization, preventing overfitting and enhancing model stability.
- Computational Efficiency: Leveraging parallel processing and optimized tree learning, XGBoost trains significantly faster than LSTM.

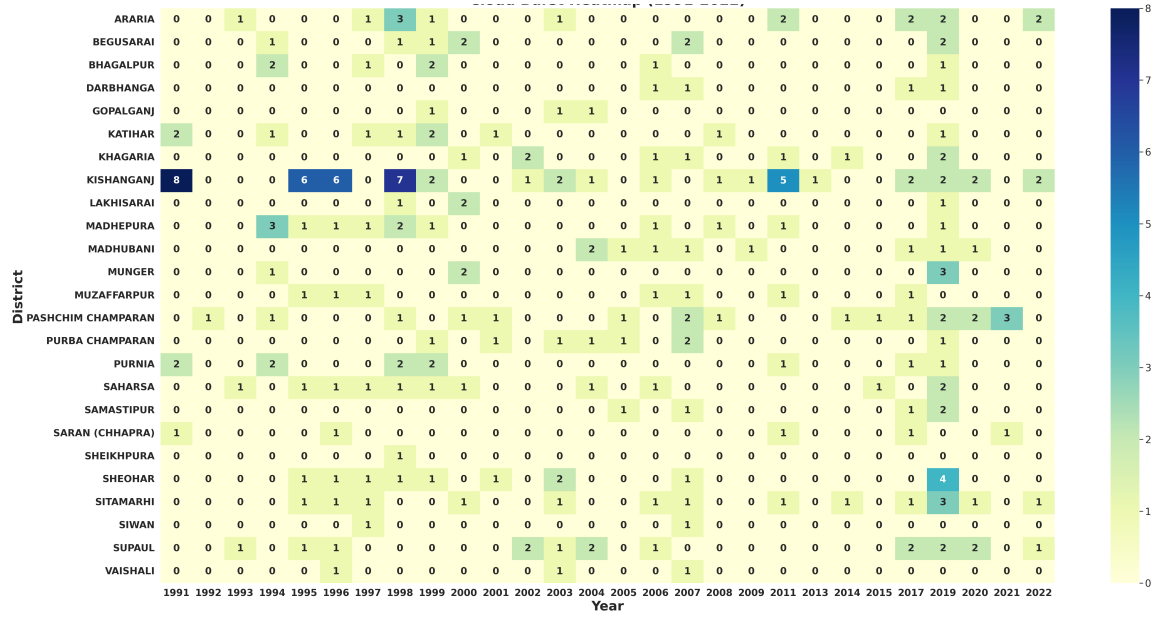


Figure 9: Cloudburst heat map (1991-2022)

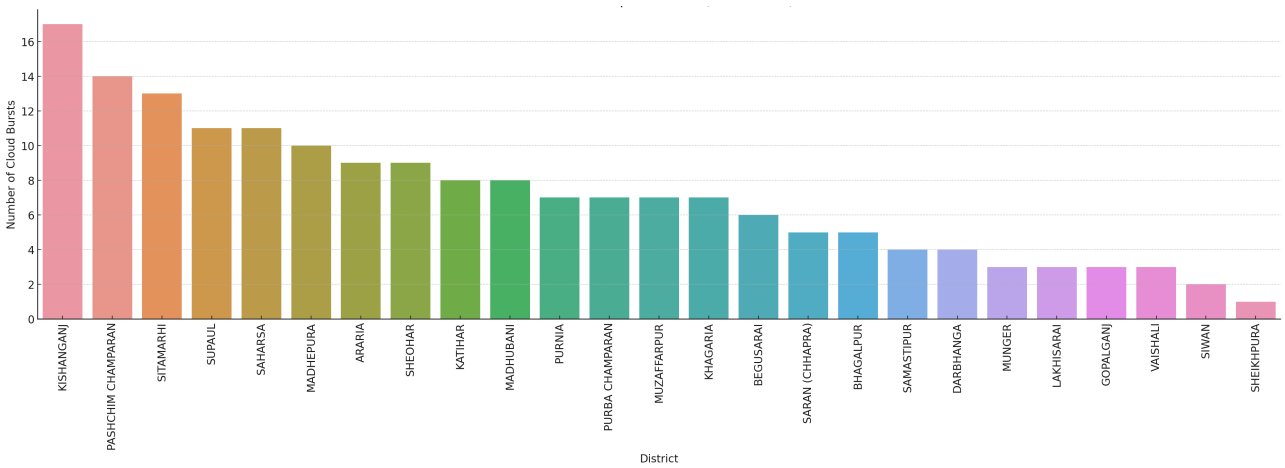


Figure 10: Cloudbursts per district (1991-2022)

These advantages explain why XGBoost achieves the highest accuracy and lowest error rates, making it the preferred choice for forecasting in this domain. Our analysis confirms that XGBoost outperforms both LSTM and Random Forest, offering superior accuracy, lower error metrics, and higher classification effectiveness. Furthermore, the discussion on turbidity risk factors highlights the practical implications of our model's predictions. These insights contribute to improved climate monitoring, risk assessment, and early warning systems for environmental hazards.

10 Conclusion

This paper presents a comprehensive framework for forecasting rainfall and cloudburst events, focusing on an empirical case study in Bihar, India. The study highlights the application of XGBoost-driven modeling for spatiotemporal forecasting at the district scale. By addressing these challenges through tailored social and economic policies, alongside targeted training and skill development programs, the study identifies pathways to reduce flood vulnerability and improve disaster readiness.

Key findings reveal that Bihar is highly prone to floods, with rainfall prediction heat maps indicating a continuous rise in monsoonal rainfall in the coming years, increasing the risk of flash floods. Araria is projected to face the highest number of cloudburst events from 2031 onwards, followed by Sitamarhi after 2044. Other vulnerable districts include Khagaria, Kishanganj, Munger, Paschim Champaran, Samastipur, Sheohar, and Supaul, each expected to experience one cloudburst event annually.

The findings emphasize the urgent need for govern-

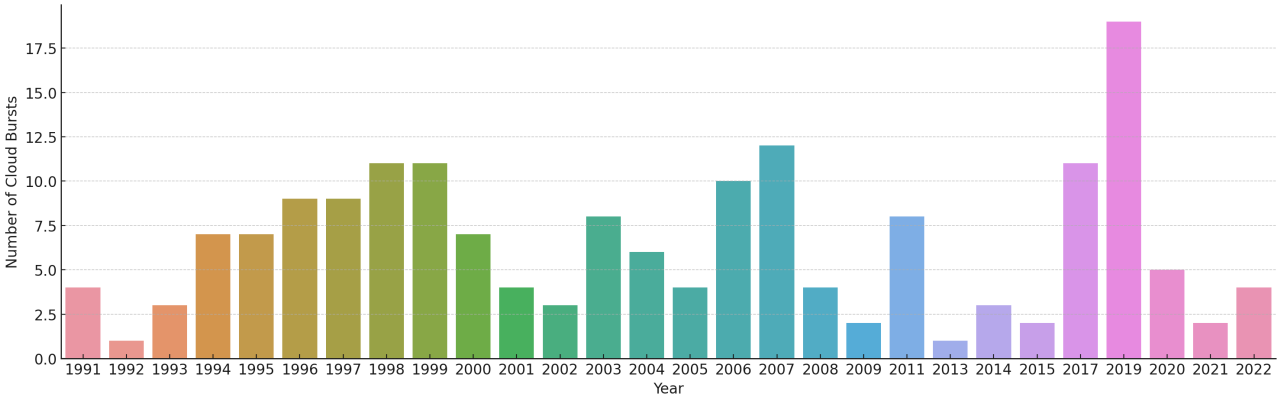


Figure 11: Cloudbursts per year (1991-2022)

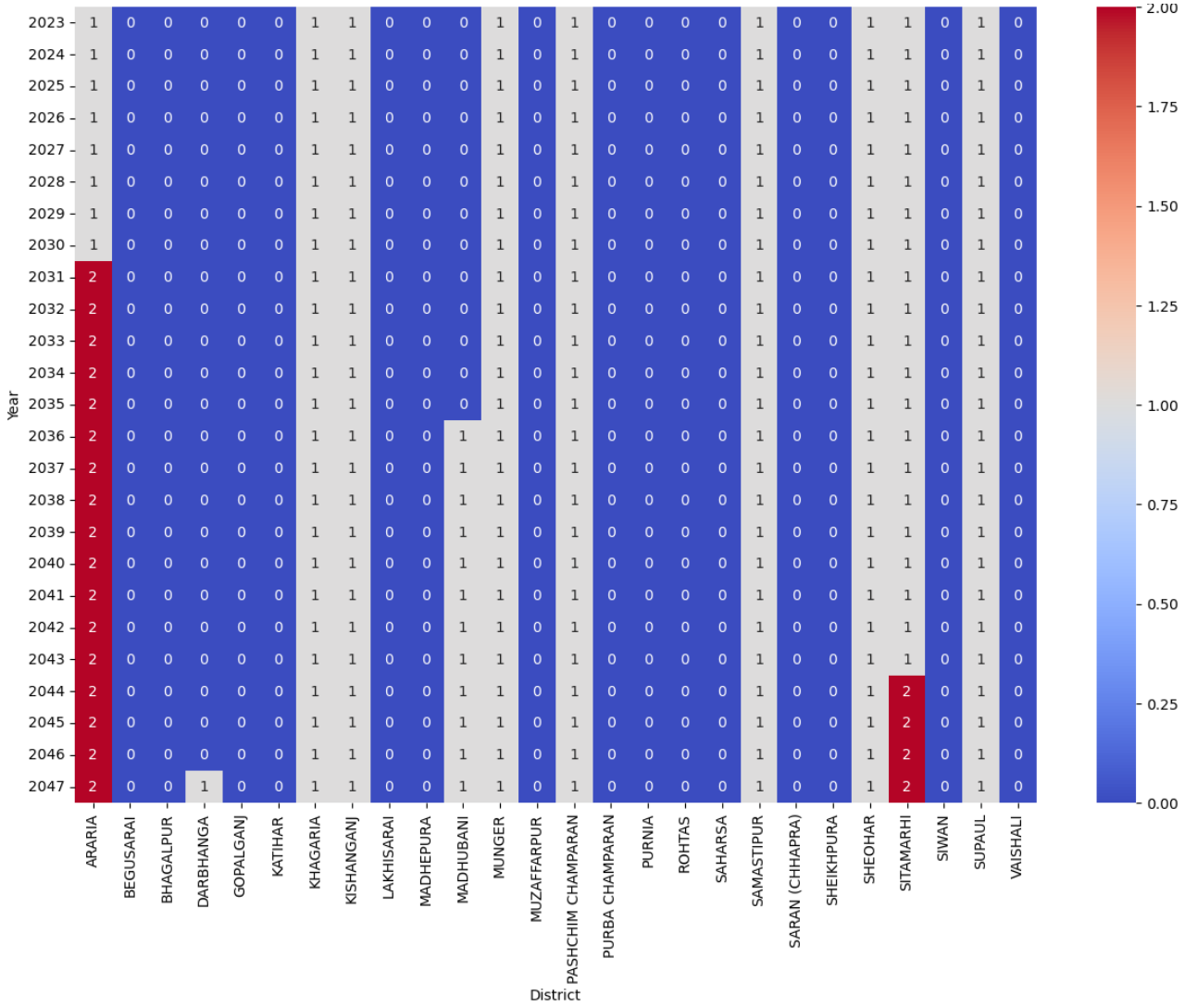


Figure 12: Forecasted numbers of cloudbursts/flash floods per year for each district (2023-2047)

ment intervention to develop adaptive mitigation policies based on district-level vulnerabilities. A well-coordinated Early Warning System, integrating institutions, task forces, and local communities, is essential for informed decision-making and effective disaster preparedness training.

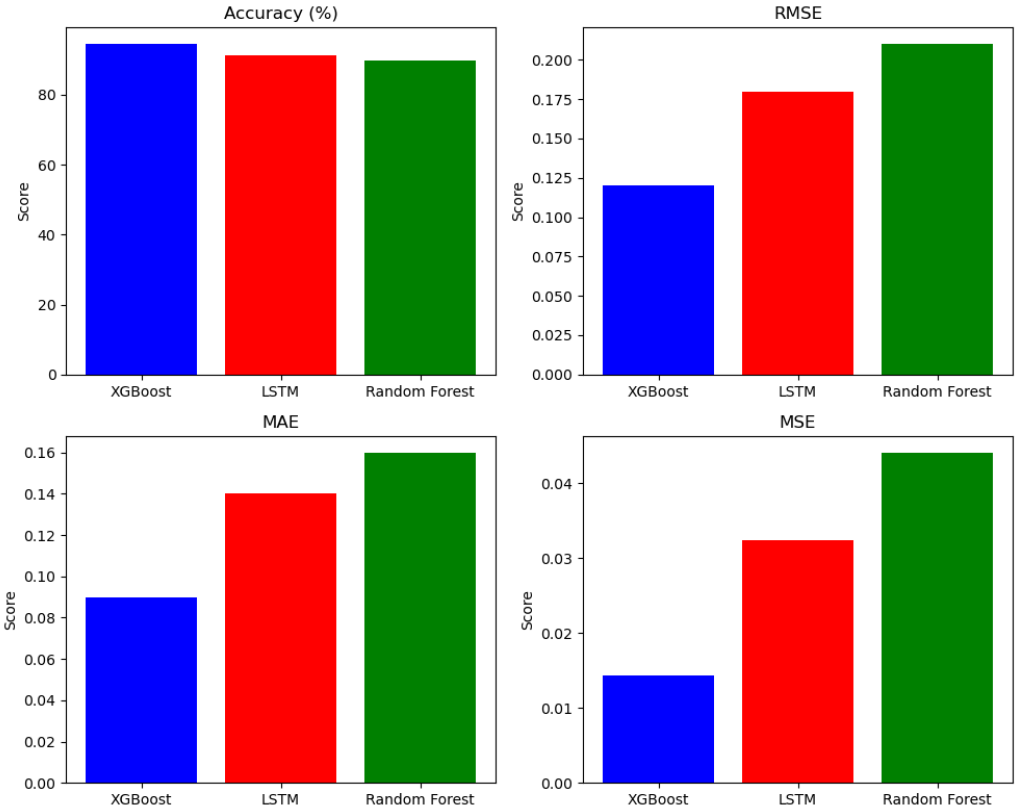


Figure 13: Comparative Accuracy,MSE, RMSE and MAE of XGBoost, LSTM, and Random Forest

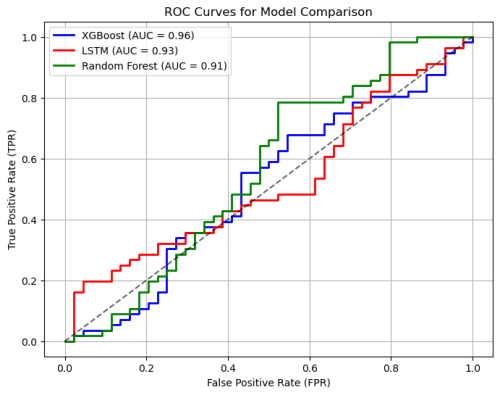


Figure 14: ROC Curves for XGBoost, LSTM, and Random Forest

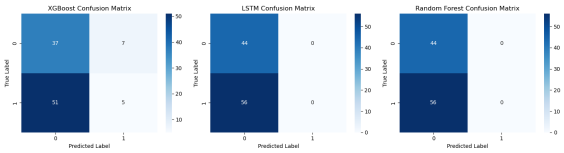


Figure 15: Confusion Matrices for XGBoost, LSTM, and Random Forest

Acknowledgement

The authors thank Mr. Tanmay Sharma, Research Scholar, Innovation Studies, SHSS, IIT Indore for his support in computing the variables.

Conflict of interest

The authors declare no conflict of interest.

Data availability statement

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Ams: American metrological society. https://glossary.ametsoc.org/wiki/Welcome/. Accessed: March 23, 2024.

[2] Em-dat: The international disaster database. https://www.emdat.be/. Accessed: March 23, 2024.

[3] Flood Management Information System, Bihar. https://www.fmiscwrdbihar.gov.in/fmis/history.html. Accessed on February 12, 2024.

[4] Know your risk. http://bsdma.org/Know-Your-Risk.aspx?id=3. Accessed: March 23, 2024.

[5] NDRF Flood Operations 2019. https://ndrf.gov.in/operations/flood-2019. Accessed on February 12, 2024.

[6] Water resources information system for india. https://indiawris.gov.in/wris/#/DataDownload. Accessed: March 23, 2024.

[7] Vidhi Bharti and Charu Singh. Evaluation of error in trmm 3b42v7 precipitation estimates over the himalayan region. *Journal of Geophysical Research: Atmospheres*, 120(24):12458–12473, 2015.

[8] WE Bonnett. Cloudburst near citrus, cal. *Monthly Weather Review*, 32(8):358–358, 1904.

[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[10] A Chevuturi, AP Dimri, Someshwar Das, A Kumar, and D Niyogi. Numerical simulation of an intense precipitation event over rudraprayag in the central himalayas during 13–14 september 2012. *Journal of Earth System Science*, 124:1545–1561, 2015.

[11] Colin Clark. The cloudburst of 2 july 1893 over the cheviot hills, england. *Weather*, 60(4):92–97, 2005.

[12] Sining Cuevas. Examining climate change adaptation measures: an early warning system in the philippines. *International Journal of Climate Change Strategies and Management*, 4(4):358–385, 2012.

[13] Someshwar Das, Raghavendra Ashrit, and MW Moncrieff. Simulation of a himalayan cloudburst event. *Journal of earth system science*, 115:299–313, 2006.

[14] John A Day and Vincent J Schaefer. *Peterson First Guide to Clouds and Weather*. Houghton Mifflin, 1991.

[15] AP Dimri and SK Dash. Wintertime climatic trends in the western himalayas. *Climatic change*, 111(3-4):775–800, 2012.

[16] Jianhua Dong, Wenzhi Zeng, Lifeng Wu, Jiesheng Huang, Thomas Gaiser, and Amit Kumar Srivastava. Enhancing short-term forecasting of daily precipitation using numerical weather prediction bias correcting with xgboost in different regions of china. *Engineering Applications of Artificial Intelligence*, 117:105579, 2023.

[17] JS Douglas. A california cloudburst. *Monthly Weather Review*, 36(9):299–300, 1908.

[18] Storm Dunlop. *The weather identification handbook*. Globe Pequot, 2003.

[19] Storm Dunlop. *A dictionary of weather*. OUP Oxford, 2008.

[20] AD Elmer. Cloudbursts. *Monthly Weather Review*, 30(10):478–478, 1902.

[21] Bapon SHM Fakhruddin and Lauren Schick. Benefits of economic assessment of cyclone early warning systems-a case study on cyclone evan in samoa. *Progress in Disaster Science*, 2:100034, 2019.

[22] Center for Climate Change and Sustainability Studies. Bengaluru Floods: Case of Urban Flooding. https://climatetrends.in/wp-content/uploads/2023/04/bengaluru-floods-case-of-urban-flooding.pdf. Accessed on February 12, 2024.

[23] Juliane Loraine Fry, Hans-F Graf, Richard Grotjahn, Marilyn Raphael, Clive Saunders, and Richard Whitaker. *The encyclopedia of weather and climate change: a complete visual guide*. University of California Press Berkeley, CA, 2010.

[24] Kumar Gaurav, François Métivier, Olivier Devauchelle, Rajiv Sinha, Hugo Chauvet, Morgane Houssais, and Hélène Bouquerel. Morphology of the kosi megafan channels. *Earth Surface Dynamics*, 3(3):321–331, 2015.

[25] Mahmoud Yousef M Ghoneem and Ahmed Khaled A Elewa. The early warning application role in facing the environmental crisis and disasters:'preliminarily risk management strategy for the greater city of cairo'. *Spatium*, pages 40–48, 2013.

[26] Bhupendra Nath Goswami, Vengatesan Venugopal, Debasis Sengupta, MS Madhusoodanan, and Prince K Xavier. Increasing trend of extreme rain events over india in a warming environment. *Science*, 314(5804):1442–1445, 2006.

[27] P Goswami and KV Ramesh. Extreme rainfall events: vulnerability analysis for disaster management and observation system design. *Current Science*, pages 1037–1044, 2008.

[28] P Guhathakurta, OP Sreejith, and PA Menon. Impact of climate change on extreme rainfall events and flood risk in india. *Journal of earth system science*, 120:359–373, 2011.

[29] Umesh K Haritashya, Pratap Singh, Naresh Kumar, and Yatveer Singh. Hydrological importance of an unusual hazard in a mountainous basin: flood and landslide. *Hydrological Processes: An International Journal*, 20(14):3147–3154, 2006.

[30] Robert E Horton and George T Todd. Cloudburst rainfall at taborton, ny, august 10, 1920. *Monthly Weather Review*, 49(4):202–204, 1921.

[31] India Meteorological Department. Press release no. 2555/2023, October 1 2023. Last accessed: August 13, 2024.

[32] India Meteorological Department. Monsoon frequently asked questions, n.d. Accessed: March 23, 2024.

[33] India Meteorological Department (IMD). Understanding cloudbursts and their impacts, 2020. Accessed on [Insert Access Date if online].

[34] D Izzo. Fisica delle nubi e delle precipitazioni. *Manuale di Meteorologia. Giuliacci M, Giuliacci A, Corazzon P (eds). Alpha Test: Milano*, pages 473–524, 2010.

[35] Warren R King. Record cloudburst flood in carter county, tenn., june 13, 1924. *Monthly Weather Review*, 52(6):311–313, 1924.

[36] Charles D Kolstad and Frances C Moore. Estimating the economic impacts of climate change using weather observations. *Review of Environmental Economics and Policy*, 2020.

[37] V Krishnamurthy. *Extreme events and trends in the Indian summer monsoon*. Center of Ocean-Land-Atmosphere Studies, 2011.

[38] Ameya Kshirsagar and Parth Sanghavi. Geothermal, oil and gas well subsurface temperature prediction employing machine learning. In *47 th workshop on geothermal reservoir engineering. https://pangea. stanford. edu/ERE/db/GeoConf/papers/SGW/2022/Kshirsagar. pdf*, 2022.

[39] Guru Dayal Kumar and Kalandi Charan Pradhan. Assessing the district-level flood vulnerability in bihar, eastern india: An integrated socioeconomic and environmental approach. *Environmental Monitoring and Assessment*, 196(9):799, 2024.

[40] Guru Dayal Kumar, Kalandi Charan Pradhan, and Shekhar Tyagi. Deep learning forecasting: An lstm neural architecture based approach to rainfall and flood impact predictions in bihar. *Procedia Computer Science*, 235:1455–1466, 2024.

[41] Guru Dayal Kumar, Shekhar Tyagi, and Kalandi Charan Pradhan. Predictive ml analysis: Rainfall & flood vulnerability in bihar, india. In *Artificial Intelligence and Information Technologies*, pages 447–453. CRC Press, 2024.

[42] Manosi Lahiri. *Bihar geographic information system*. Popular Prakashan, Bombay, IN, 1992.

[43] John Lovel. Thunderstorm, cloudburst and flood at langtoft, east yorkshire, july 3rd, 1892. *Quarterly Journal of the Royal Meteorological Society*, 19(85):1–15, 1893.

[44] Darren Lumbroso, Emma Brown, and Nicola Ranger. Stakeholders' perceptions of the overall effectiveness of early warning systems and risk assessments for weather-related hazards in africa, the caribbean and south asia. *Natural Hazards*, 84:2121–2144, 2016.

[45] Xiongfa Mai, Haiyan Zhong, and Ling Li. Research on rain or shine weather forecast in precipitation nowcasting based on xgboost. In *The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pages 1313–1319. Springer, 2020.

[46] Valérie Masson-Delmotte, Panmao Zhai, Anna Pirani, Sarah L Connors, Clotilde Péan, Sophie Berger, Nada Caud, Y Chen, L Goldfarb, MI Gomis, et al. Climate change 2021: the physical science basis. *Contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change*, 2(1):2391, 2021.

[47] A Austin Miller. Cause and effect in a welsh cloudburst. *Weather*, 6(6):172–179, 1951.

[48] S Nandargi and ON Dhar. Extreme rainstorm events over the northwest himalayas during 1875–2010. *Journal of Hydrometeorology*, 13(4):1383–1388, 2012.

[49] Giuseppe Orlando and Michele Bufalo. A generalized two-factor square-root framework for modeling occurrences of natural catastrophes. *Journal of Forecasting*, 41(8):1608–1622, 2022.

[50] Isidoro Orlanski. A rational subdivision of scales for atmospheric processes. *Bulletin of the American Meteorological Society*, pages 527–530, 1975.

[51] Ahmedbahaaaldin Ibrahem Ahmed Osman, Ali Najah Ahmed, Ming Fai Chow, Yuk Feng Huang, and Ahmed El-Shafie. Extreme gradient boosting (xgboost) model to predict the groundwater levels in selangor malaysia. *Ain Shams Engineering Journal*, 12(2):1545–1556, 2021.

[52] AC Pandey, Suraj Kumar Singh, and MS Nathawat. Waterlogging and flood hazards vulnerability and risk assessment in indo gangetic plain. *Natural hazards*, 55:273–289, 2010.

[53] Bikash Ranjan Parida, Sailesh N Behera, Oinam Bakimchandra, Arvind Chandra Pandey, and Nilendu Singh. Evaluation of satellite-derived rainfall estimates for an extreme rainfall event over uttarakhand, western himalayas. *Hydrology*, 4(2):22, 2017.

[54] Santhanam Ramraj, Nishant Uzir, R Sunil, and Shatadeep Banerjee. Experimenting xgboost algorithm for prediction and classification of different datasets. *International Journal of Control Theory and Applications*, 9(40):651–662, 2016.

[55] Jun Rentschler, Melda Salhab, and Bramka Arga Jafino. Flood exposure and poverty in 188 countries. *Nature communications*, 13(1):3527, 2022.

[56] Chandan Roy, Saroje Kumar Sarkar, Johan Åberg, and Rita Kovordanyi. The current cyclone early warning system in bangladesh: providers' and receivers' views. *International journal of disaster risk reduction*, 12:285–299, 2015.

[57] Md Shahjahan. *Assessing the cyclone early warning services of women, children and person with disability: a case study in Nijhumdwip*. PhD thesis, BRAC Univeristy, 2018.

[58] Anand Shankar, Ashish Kumar, Bikash Chandra Sahana, and Vivek Sinha. A case study of heavy rainfall events and resultant flooding during the summer monsoon season 2020 over the river catchments of north bihar, india. *Vayumandal*, 48(2):17–28, 2022.

[59] Susan Solomon. *Climate change 2007-the physical science basis: Working group I contribution to the fourth assessment report of the IPCC*, volume 4. Cambridge university press, 2007.

[60] Nicholas Herbert Stern. *The economics of climate change: the Stern review*. cambridge University press, 2007.

[61] Robert Szczepanek. Daily streamflow forecasting in mountainous catchment using xgboost, lightgbm and catboost. *Hydrology*, 9(12):226, 2022.

[62] Ashok Kumar Tripathi, PK Gupta, Hemraj Saini, and Geetanjali Rathee. Mvi and forecast precision upgrade of time series precipitation information for ubiquitous computing. *Informatica*, 47(5), 2023.

[63] Gaurav Tripathi, Arvind Chandra Pandey, and Bikash Ranjan Parida. Flood hazard and risk zonation in north bihar using satellite-derived historical flood events and socio-economic data. *Sustainability*, 14(3):1472, 2022.

[64] BM Varney. The great hailstorm in southeastern new hampshire and northeastern massachusetts, july 17, 1924. *Monthly Weather Review*, 52(8):394–395, 1924.

[65] Ralph R Woolley. Cloudburst floods in utah: Us geol. *Survey, Water*, 1946.

[66] World Meteorological Organization. State of climate services 2020 report: Move from early warnings to early action, 2020.

[67] Mohammad Zakwan and Zeenat Ara. Statistical analysis of rainfall in bihar. *Sustainable Water Resources Management*, 5(4):1781–1789, 2019.

Optimizing Random Forest Models with Snake Optimization Algorithm for Predicting E-commerce User Purchase Behaviour

Pengfei Li
School of Management, Zhengzhou Business University, Gongyi City, Henan Province, China
E-mail: Lee_samuelson@163.com

Keywords: e-commerce users, purchase behaviour, random forest, snake optimization algorithm

Recieved: December 5, 2024

This study proposes a Snake Optimization-based Random Forest (SO-RF) model for predicting e-commerce user behavior. Key user interaction metrics, including browsing records, purchase history, search keywords, click rate, dwell time, add-to-cart times, user comments, and visit time, serve as input features, while user conversion rate and purchase rate are the target metrics. The dataset undergoes preprocessing and feature engineering to extract meaningful patterns. The Snake Optimization (SO) algorithm fine-tunes the hyperparameters of the Random Forest (RF) model, enhancing predictive performance and generalization. Experimental results demonstrate that SO-RF outperforms conventional RF, Simulated Annealing-based RF (SA-RF), and Sparrow Search Algorithm-based RF (SSA-RF) on the test set, achieving an MAE of 0.31959, MAPE of 1.6652, MSE of 0.17625, RMSE of 0.41983, and R^2 of 0.96678. These findings provide valuable insights for e-commerce platforms, enabling personalized marketing strategies, improved user experience, and enhanced sales performance through accurate behavior prediction.

Povzetek: Članek predstavi optimiziran model naključnih gozdov z algoritmom Snake (SO-RF) za napovedovanje nakupnega vedenja uporabnikov e-trgovine, s čimer izboljša prodajo in personalizacijo.

1 Introduction

With the rapid development of e-commerce, e-commerce platforms have accumulated a large amount of user behavior data (Xie C, 2020). These data contain information about users' browsing records, purchase history, search keywords, click rate, dwell time, number of shopping carts added, user comments, visit time, and other aspects of e-commerce platforms. Through the analysis and mining of these data, we can gain an in-depth understanding of users' shopping habits, needs, and preferences, which provides the basis for enterprises to develop more accurate and personalized marketing strategies (Wu Z, 2021). Therefore, e-commerce user behavior prediction research is of great significance.

First of all, e-commerce user behavior prediction research can help enterprises better understand user needs and market conditions (Khrais L T, 2020). Through the analysis and mining of user behavior data, we can understand the shopping habits, needs, and preferences of users and grasp the changes and trends of the market. This information can help enterprises develop more accurate and personalized marketing strategies, improve user conversion and purchase rates, and increase their sales and profits (Niu Z, 2021). At the same time, through the analysis of user behavior data, the potential needs and unsatisfied needs of users can also be found, providing a basis for enterprises to develop new products and services (Blazevic V, 2008).

Secondly, e-commerce user behavior prediction research can improve the marketing efficiency and effectiveness of enterprises (Wakil K, 2020). Through the analysis and mining of user behavior data, target user groups can be accurately located, and targeted marketing strategies can be formulated. Relevant goods and services can be recommended to users based on their browsing records and purchase history; the titles and descriptions of goods can be optimized based on users' search keywords and click-through rates to increase the click-through and purchase rates of goods; and the design and layout of the page can be optimized based on the user's dwell time and the number of times they add to the shopping cart to improve the user's shopping experience and satisfaction. These targeted marketing strategies can improve the marketing efficiency and effectiveness of enterprises, reduce marketing costs, and increase their sales and profits (Zhang B, 2021).

Again, e-commerce user behavior prediction research can improve the competitiveness and market share of enterprises (Xiahou X, 2022). Through the analysis and mining of user behavior data, changes and trends in the market can be found, providing the basis for enterprises to develop more accurate and personalized marketing strategies. These strategies can help enterprises stand out in the fierce market competition and improve their competitiveness and market share (To M L, 2006). At the same time, through the analysis of user behavior data, new market opportunities, and business models can also be found, providing new ideas and directions for the

development of enterprises (Chang K, 2003).

Finally, e-commerce user behavior prediction research can also provide valuable data support and a decision-making basis for enterprises (Fan S, 2015). Through the analysis and mining of user behavior data, it can provide valuable data support and a decision-making basis for enterprises. These data can provide support and guidance for strategic planning, product development, marketing, and other aspects of the enterprise, helping the enterprise to make more scientific and reasonable decisions (Poggi N, 2013). At the same time, through the analysis of user behavior data, the problems and shortcomings of enterprises can also be found, providing a basis for enterprises to improve and perfect their services (Guo Y, 2018).

E-commerce user behavior data contains a large amount of information, how to extract meaningful features from this data to better understand the user needs and market conditions is an important problem to be solved in this study. The random forest model is a commonly used machine learning algorithm, but its performance is affected by parameter settings. How to optimize the parameters of the random forest model using a snake optimization algorithm to improve the performance and generalization ability of the model. The accuracy and reliability of the prediction results are important metrics to assess the performance of the model. The goal of this study is to predict user behavior—specifically conversion rate and purchase rate—in order to assist businesses better comprehend user requirements and market conditions, create more precise and personalized marketing tactics, and increase user satisfaction and sales. The chosen metrics—dwell time, click rate, purchase history, and search keywords—are crucial for comprehending user shopping behavior and improving predictive accuracy. Dwell time reflects user engagement, as more time spent on a product page indicates increased interest and purchase intent. Click rate shows interaction frequency, revealing which products receive the most attention. Purchase history offers direct insights into past purchasing patterns, allowing you to forecast future purchases using established preferences. Search keywords reveal user intent by emphasizing particular interests and requirements at various stages of the purchasing process. These metrics were selected using previous research and industry best practices, ensuring their usefulness in accurately forecasting conversion and purchase rates. Their inclusion improves model efficiency by capturing both explicit and implicit customer behavior signals, rendering them the ideal option for improving personalized suggestions and marketing tactics.

The Snake Optimization (SO) algorithm is ideal for improving Random Forest (RF) in e-commerce analytics because of its adaptive exploration-exploitation balance, which effectively tunes hyperparameters such as tree depth and the number of estimators to improve predictive accuracy. Unlike traditional optimization techniques, SO simulates natural selection behaviors, resulting in a more varied and broadly optimal parameter search, which is critical for dealing with complex, high-dimensional e-commerce data. Beyond e-commerce, SO-RF can be used in healthcare analytics to optimize diagnostic models by fine-tuning disease prediction classification thresholds, as well as marketing analytics to improve consumer segmentation models by improving decision trees using consumer behavior data. The algorithm's versatility in feature selection and parameter optimization renders it an effective tool for a wide range of predictive modeling tasks across industries.

1.1 Research objective

This research aims to improve e-commerce user behavior prediction by combining SO with Random Forest (RF), resulting in the SO-RF model. The main objective is to enhance prediction accuracy, optimize feature selection, and decrease model bias-variance trade-offs using adaptive parameter tuning.

1.2 Hypotheses

H1: SO-RF attains higher predictive accuracy than conventional RF, Simulated Annealing (SA-RF), and Sparrow Search Algorithm (SSA-RF).

H2: SO's adaptive search mechanism enhances feature selection, leading to better generalization and decreased overfitting.

H3: Despite its computational complexity, SO improves model robustness in managing dynamic user behavior patterns in e-commerce.

1.3 Expected outcomes for E-commerce applications

The proposed SO-RF model is expected to offer more accurate predictions of user behavior, allowing e-commerce platforms to improve customized suggestions, enhance marketing tactics, and increase customer engagement. Furthermore, the model's scalability and adaptability may enable dynamic pricing, demand prediction, and real-time decision-making in online retail settings.

2 Related study

Random Forest (RF) is an integrated learning model that improves the overall prediction accuracy by combining the prediction results of multiple decision trees (Naghibi S A, 2016). Random forest models have a wide range of research areas, including classification: random forests can be applied to multi-class classification problems, such as image classification, text classification, bioinformatics classification, etc. (Verikas A, 2011);

regression: random forests can be applied to regression problems, such as house price prediction, stock price prediction, etc. (Fawagreh K, 2014); feature selection: random forests can provide importance ranking of features and thus help in feature selection (Cutler D R, 2007). Anomaly detection: random forests can be applied to anomaly detection, such as financial fraud detection, network intrusion detection, etc. (Amaratunga D, 2008). Bioinformatics: Random Forest can be applied to bioinformatics problems such as gene classification, protein structure prediction, etc (Segal M, 2011). Marketing: random forests can help companies analyze customer data to develop more accurate marketing strategies (Li T, 2016). Healthcare: random forests can be applied to healthcare problems such as disease risk prediction and patient susceptibility prediction (Bin J, 2016).

The main advantages of the model include: efficient training speed and good parallelization ability; strong robustness to high-dimensional data and missing data; the ability to give the importance ranking of features, which helps feature selection; and good generalization ability and resistance to overfitting (Ao Y, 2019). However, the random forest model also has some defects: it is more sensitive to noise and outliers; in some cases, overfitting may occur; for some specific types of data, such as text data or image data, special preprocessing or feature engineering may be required; and the interpretability of the model is relatively poor, which makes it difficult to intuitively understand the decision-making process of the model (Shi K, 2018).

To address the shortcomings of the random forest model, there are some improvement methods, such as the introduction of regularization terms and the use of deep forest structure (Zhang W, 2021). In addition, some studies are focusing on how to improve the interpretability of the random forest model, such as decision tree-based interpretation methods, model decomposition-based interpretation methods, etc. (Ren S, 2015).

Yan and Zhou (2024) proposed a recommendation algorithm that uses matrix reduction methods to improve network information analysis and user behavior predictions. Yuan (2024) proposed a deep learning-based framework for predicting consumer behavior, which uses sophisticated neural networks to optimize enterprise precision marketing campaigns. Cheng and He (2024) used random forest optimization to improve product modeling processes, improving design effectiveness and visual efficiency in e-commerce applications. Table 1 shows the summary table of these existing works.

Table 1: Summary table

Citation	Key Focus	Key Findings	Applications
Naghibi et al. (2016)	Groundwater potential mapping utilizing RF, BRT, and CART	BRT surpassed CART and RF with an AUC of 0.8103, while RF had the minimum at 0.7119	GIS-based environmental tracking
Verikas et al. (2011)	Variable significance and performance of RF	Found high variance in variable importance rankings, denoting instability in small datasets	Feature selection and data exploration
Fawagreh et al. (2014)	Evolution and improvements in RF	Discussed RF's enhancements and future directions in ensemble learning	Classification and data mining
Cutler et al. (2007)	RF for ecological classification	RF demonstrated better classification accuracy compared to conventional techniques.	Ecological modeling and species classification
Amaratunga et al. (2008)	Enriched RF for feature selection	Proposed weighted sampling to enhance RF performance in high-dimensional datasets	Bioinformatics and microarray data evaluation
Segal & Xiao (2011)	Multivariate RF for numerous response prediction	Showed improved predictive accuracy in multi-response settings	Ecology and predictive modeling
Bin et al. (2016)	Modified RF for multi-class classification	Enhanced RF's performance utilizing NIR spectroscopy for tobacco leaf grading	Spectroscopy-based classification

Existing RF-based methodologies and variants have limitations like instability in feature selection, bias toward dominant features, and inadequacies in dealing with high-dimensional or imbalanced data. While improvements such as enriched RF and multivariate RF tackle some problems, they do not include adaptive learning or optimal feature organization. A hybrid SO-RF approach addresses these gaps by incorporating self-organizing mechanisms, which improve feature selection, adaptability, and classification efficiency, rendering it more resilient for intricate datasets.

3 Description Of the methodology

3.1 Random Forest model

The random forest model is studied because it performs well in many real-world problems and has the advantages of being efficient, robust, and easy to use. Random forest model is an integrated learning model based on decision trees, which improves the overall prediction accuracy by combining the prediction results of multiple decision trees. Its core idea is to use the self-service sampling (bootstrap sampling) method to extract multiple samples from the original dataset, and then construct a decision tree for each sample, and ultimately vote or average the prediction results of all the decision trees to arrive at the final prediction results (Lin W, 2017). It can handle classification and regression problems effectively by constructing multiple decision trees and integrating them. The interpretability of Random Forest is known to be limited because it functions as an ensemble of decision trees, rendering it hard to directly explain individual predictions. To tackle this, techniques such as SHAP (SHapley Additive Explanations) values can be used to quantify each feature's contribution to the model's decisions. By incorporating SHAP analysis, the model's results can be better understood, providing insights into feature importance and decision-making procedures, ultimately enhancing transparency and trust in predictions.

3.1.1 Decision tree formulation

Decision tree is the basic component of a random forest, and its formula includes the following aspects:

Finding the information gain formula:

The information gain is used to measure the degree of information reduction under the division of feature values, and its formula is:

$$\Delta H(D,A) = H(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} H(D_v) \quad (1)$$

where is the initial information entropy of the dataset, is the conditional entropy when the feature takes the value of, is the number of values of the feature, is the number of samples of the dataset, is the number of samples when the feature takes the value of.

Find the formula for the Gini index:

The Gini index is used to measure the purity of the dataset with the formula:

$$Gini(D) = 1 - \sum_{k=1}^{K} (P_k)^2 \quad (2)$$

Where K is the number of categories in the dataset and P_k is the proportion of samples in the dataset belonging to category No. to the total samples.

Decision tree construction algorithm formula

The decision tree construction algorithm is usually based on information gain or the Gini index for feature selection. The formula for building a decision tree is as follows:

Input: training set, feature set, threshold value

Output: decision tree

If the samples all belong to the same category, it will be returned as a single node tree, labeled as;

If it is the empty set, i.e., there are no more features to choose from, it will be treated as a single node tree, labeled as the category with the highest number of samples in it, and returned;

Select the optimal feature based on the information gain or Gini index;

If the information gain or Gini index is less than the threshold, it will be returned as a single node tree, labeled as the category with the highest number of samples in;

Otherwise, it will be divided into subsets based on the values of the features;

For each subset, recursively call the above steps to construct the subtree;

Will be connected to the top.

3.1.2 Random Forest formulation

Random forest is an algorithm for prediction or classification by integrating multiple decision trees, and its formula includes the following aspects:

Random forest generation formula:

The formula for random forest generation is:

$$RF(X) = \frac{1}{T} \sum_{t=1}^{T} f_t(X) \quad (3)$$

where $RF(X)$ denotes the prediction result of the random forest on the sample X, T denotes the number of decision trees in the random forest, and $f_t(X)$ denotes the prediction result of the tth decision tree on the sample X.

Feature selection formula:

Random forests are constructed by randomly selecting features for decision tree construction, and the formula for feature selection is:

$$S = \sum_{i=1}^{S} i\,\text{importance(i)} \quad (4)$$

where S denotes the sum of selection probabilities of all features in the feature set and $importance(i)$ Denotes the selection probability of the feature.

3.2 Snake optimization algorithm

Snake Optimization Algorithm (SO) is an intelligent optimization algorithm that simulates the specific mating behavior of snakes, proposed by Fatma A. Hashim and Abdelazim G. Hussien in 2022, which is inspired by the foraging and reproductive behaviors and patterns of snakes (Hashim F A. 2022).

The principle of the snake optimization algorithm is to simulate the mating behavior of snakes in late spring and early summer. The mating process of snakes depends not only on the temperature but also on the availability of food. If the temperature is low and food is sufficient, mating occurs, otherwise the snake will only search for food or eat the remaining food. In the snake optimization algorithm, the population is divided into two equal groups i.e. males and females. Male snakes will fight with each other to attract the attention of females. Females have the right to decide whether to mate or not. If mating occurs, the female starts laying eggs in the nest or burrow and once the eggs appear, she leaves (Zheng W, 2023).

The snake optimization algorithm is divided into two stages, namely global exploration and local exploitation. Exploration represents the environmental factors, i.e., cold places and food, in which case the snake only searches for food in its surroundings. For exploitation, this phase includes many transitions to make the global more efficient. In situations where food is available but the temperature is high, the snake will only focus on eating the available food. In the battle mode, each male will fight to get the best female and each female will try to choose the best male (Fu H, 2022). Snake optimization algorithm is a kind of intelligent optimization algorithm that simulates the behavior of natural creatures, with good optimization-seeking ability and fast convergence. It can be applied to a variety of practical problems, such as function optimization, machine learning, deep learning, and so on (Yan C, 2023).

The snake optimization algorithm first generates a uniformly distributed random totality to be able to start the optimization algorithm process.

$$X_i = X_{\min} + r \times (X_{\max} - X_{\min}) \tag{5}$$

It will be divided into two groups of males and females, and for the study, it is assumed that the number of males will be 50% and the number of females will be 50%. The exploration and development phase of the snake optimization algorithm is mainly affected by the temperature Temp and the amount of food Q, which is given by Eq (6)

$$Temp = \exp(-\frac{t}{T}), \quad Q = c_1 \times \exp(\frac{t-T}{T}) \tag{6}$$

where t represents the current iteration, T represents the maximum number of iterations, and c1=0.5. Exploration phase (no food): Q<threshold (0.25); exploitation phase (food present): Q>threshold (0.6).

When Q<threshold (0.25), the stochastic exploration formula is:

$$X_i^m = X_{rand}^m(t) \pm c_2 \times A_m \times ((X_{\max} - X_{\min}) \times rand + X_{\min}) \tag{7}$$

where X_i^m to male snake location, X_{rand}^m refers to random snake location, c1 = 0.05, and A_m is the ability of males to find.

The Snake Optimization Algorithm (SO) converts snake mating behavior into an effective optimization framework by modeling the balance between exploration (searching for optimal solutions) and exploitation (fine-tuning promising candidates). In machine learning, SO efficiently tunes model parameters by utilizing its dual-phase method: the exploration phase, driven by temperature and food availability, guarantees a diverse search of the solution space to avoid premature convergence, while the exploitation phase, where male snakes compete for females, improves the best solutions through selective mating. This adaptive mechanism, which simulates evolutionary choice, enables SO to dynamically adjust learning rates, feature weights, and hyperparameters in machine learning models. SO improves convergence speed and accuracy by integrating biologically inspired search and selection procedures, rendering it an effective tool for parameter tuning in intricate optimization tasks.

3.3 Snake optimization algorithm to optimize random forest model

The snake optimization algorithm can be applied to optimize the parameters of the random forest model. The following are the steps, formulas, and principles for optimizing the random forest model:

Steps:

Initialize the parameters of the snake optimization algorithm, including the number of populations, the maximum number of iterations, the dimensionality, and so on.

Use the snake optimization algorithm to generate uniformly distributed random populations as the initial parameters of the random forest model.

Calculate the fitness value of each individual according to the performance evaluation indexes (such as accuracy, recall, etc.) of the random forest model.

According to the principle of the snake optimization algorithm, the population is updated and evolved, including two stages of global exploration and local development.

Repeat steps 3 and 4 until the maximum number of iterations is reached or the stopping condition is satisfied.

Output the optimal random forest model parameters.

Formula:

In the snake optimization algorithm, the position update formula for each individual is:

$$X_{i,j(t+1)} = X\_\{food\} \pm c_3 \times Temp \times rand \times (X\_\{food\} - X\_i,j(t)) \quad (8)$$

Where $X_{i,j}$ denotes the position of a snake individual (male or female), X_{food} denotes the optimal position of a snake individual, rand is a random number in the range of [0,1], and c3 is a constant.

The principle of the snake optimization algorithm to optimize the random forest model is to find the optimal random forest model parameters by simulating the special mating behavior of snakes. In the snake optimization algorithm, the population is divided into two equal groups, males and females. Male snakes will fight with each other to attract the attention of females. Females have the right to decide whether to mate or not. If mating occurs, the female starts laying eggs in the nest or burrow and once the eggs appear, she leaves. The exploration and exploitation phases of the snake optimization algorithm are mainly affected by the temperature Temp and the amount of food Q. In the exploration phase, the snake will be in a state of searching for food, while in the exploitation phase, the snake will develop and optimize based on the location of food. By continuously updating and evolving the population, the snake optimization algorithm can eventually find the optimal random forest model parameters (see Fig. 1).

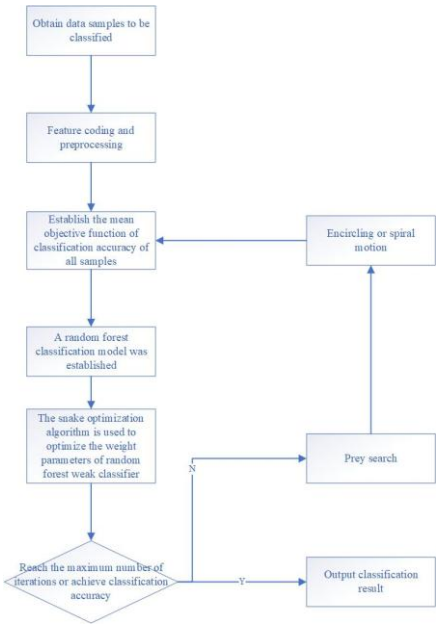


Figure 1: Flowchart of the snake optimization algorithm for optimizing the random forest framework

The Snake Optimization Algorithm (SO) improves model performance by optimizing important Random Forest (RF) parameters such as the number of trees, maximum depth, and minimum sample split. The process starts with population initialization, in which each snake signifies a unique set of RF parameters. SO creates various parameter sets during the exploration phase, ensuring a large search space, and then improves these configurations based on the mating tactic, enabling superior parameter sets to evolve. Fitness evaluation assesses model performance utilizing metrics such as accuracy and F1-score. Finally, during the selection and update phase, the best-performing parameter sets are retained and refined, guaranteeing that the RF model is optimally tuned for enhanced prediction accuracy and generalization.

3.4 Parameter tuning in snake optimization algorithm for random forest

The SO optimizes important Random Forest (RF) hyperparameters, like the number of estimators (trees), maximum tree depth, and minimum samples per split, by dynamically searching for the best configuration that optimizes model performance. First, SO creates a random population of hyperparameter sets, each indicating a possible RF configuration. The fitness function assesses each set using metrics such as accuracy, recall, and F1-score. The position update equation simulates snake evolution by balancing exploration (searching for novel hyperparameter regions) and exploitation (fine-tuning promising regions). Male individuals compete, while females choose the best candidates, guaranteeing diversity in hyperparameter selection. The iteration procedure continues until convergence, when the

optimum RF configuration is determined. This adaptive tuning method reduces overfitting and improves generalization across datasets. Pseudocode 1 shows SO-Based RF Hyperparameter Optimization.

Pseudocode 1: Snake optimization for random forest hyperparameter tuning

Initialize population (Snakes) with random values for:
- Number of estimators (n_estimators)
- Maximum tree depth (max_depth)
- Minimum samples per split (min_samples_split)

Set algorithm parameters:
- Population size (N), maximum iterations (T)
- Temperature (Temp), Food Quantity (Q)
- Exploration constant (c3)

Assess initial fitness for each snake utilizing:
Fitness = Performance Metric (for example Accuracy, F1-score)

FOR iteration t = 1 to T:
FOR each snake i:
- Update position utilizing:
$X_{i,j}(t+1) = X_{food} \pm c3 \times Temp \times rand \times (X_{food} - X_{i,j}(t))$
- Assess novel fitness score

Choose top-performing individuals (elite solutions)
Update population using mating tactic:
- Males fight for superior positions
- Females select best parameters and improve solutions

Adjust exploration vs. exploitation using temperature (Temp) decay

Check termination condition:
- If convergence is attained or max iterations reached, stop.

Return superior hyperparameter set (Optimum RF configuration)

This pseudocode guarantees reproducibility and shows how SO improves RF hyperparameters to improve prediction accuracy while maintaining computational effectiveness.

4 Empirical analysis

4.1 Data

According to the data of an e-commerce platform, 4374 sets of data were selected, and browsing records, purchase history, search keywords, click rate, dwell time, number of times of adding a shopping cart, user comments, and visit time were chosen as input indicators, and user conversion rate and purchase rate were chosen as output indicators (Bucklin R E, 2009). Among them, users' browsing records reflect their attention and interest in the products. This indicator can help the model understand users' shopping preferences and needs, to better predict their purchasing behavior. Users' purchase history records whether they have made purchases in the past, as well as the types and quantities of goods they have purchased and other information. This is valuable for predicting a user's future purchasing behavior, as it can provide important information about a user's purchasing ability and preferences. Users' search keywords on the platform can reveal their shopping needs and interests. Models can use these keywords to understand users' needs and predict what they are likely to buy. Click-through rate is the ratio of the number of times a user clicks on an item to the total number of items viewed, and it reflects the user's interest and preference for the item. Models can use this metric to predict what users are likely to buy. The amount of time a user spends on a page can provide information about their level of interest in the item. If users spend more time on the product page, they are more likely to purchase the item. The number of times a user adds an item to their shopping cart can reflect their level of interest in the item. Multiple additions to the cart may indicate a higher willingness to buy. User reviews provide their feedback on the item, which is valuable in understanding the user's level of satisfaction and possible shopping needs. Models can use this information to predict what users are likely to buy and their likely purchasing behavior. A user's online time can provide information about their shopping behavior and habits. If a user often shops between 7 pm and 10 pm, the model can use this information to predict what they are likely to buy during this time in the future. The reason for choosing user conversion rate and purchase rate as output metrics is that they are key metrics that directly measure how well an e-commerce platform is operating. Predicting these two metrics can help e-commerce platforms better formulate marketing strategies to improve revenue and operational efficiency.

In this paper, 4374 sets of data are selected, of which, 2624 sets of data are the training set, 1698 sets of data are the validation set, 52 sets of data are the test set, and the mean absolute error MAE, mean relative error MAPE, root mean square error MSE, root mean square error RMSE, and R2 of each set are calculated and analyzed by comparing them with other prediction models.

To ensure high-quality input data, numerous data preprocessing steps were used. Missing values were managed utilizing imputation methods like mean/mode imputation for numerical attributes (for example, dwell time, visit duration) and the most frequent category for categorical attributes (for example, search terms). Feature scaling was used with Min-Max Normalization to standardize numerical indicators such as click rate, shopping cart additions, and dwell time within a 0-1 range, avoiding models from being biased toward features with higher magnitudes. To guarantee predictive model compatibility, categorical encoding was executed utilizing one-hot encoding for nominal variables and label encoding for ordinal variables. The training-validation-test split (60%-38%-2%), while unconventional, is tactically designed: the large validation set (38%) guarantees resilient hyperparameter tuning and avoids overfitting, particularly for models that require extensive tuning, like ensemble learning. The small test set (2%) simulates practical e-commerce scenarios in which newly gathered, previously unseen data becomes available incrementally and acts as a final check for model generalization. This split structure prioritizes model optimization and validation while guaranteeing that the final performance evaluation is independent and realistic. The dataset size of 4,374 samples, while structured with pertinent features, is enough for training a random forest model, as the model's resilience in managing small to medium datasets reduces possible overfitting via ensemble learning.

4.2 Analysis steps

The experiments were carried out in a Windows 11 environment utilizing Python 3.9, with important libraries such as Scikit-learn (v1.2.2) for Random Forest, NumPy (v1.23) for numerical computations, and Matplotlib (v3.6) for visualizations. The Snake Optimization Algorithm (SOA) was executed with custom Python scripts that enabled parallel processing for effectiveness. The experiments were carried out on a workstation with Windows 11, an Intel Core i9-12900K processor, 64GB RAM, and an NVIDIA RTX 3090 GPU to ensure fast model training and hyperparameter tuning. The parameter search space for RF comprised tree depth (ranging from 5 to 50), number of estimators (50 to 500), minimum samples per split (2 to 20), and SOA-optimized feature selection thresholds. These computational settings allowed for comprehensive hyperparameter tuning, guaranteeing that the SOA-RF model was optimal for predicting e-commerce behavior.

Incorporating e-commerce user behavior prediction into the principle of snake optimization algorithm can be achieved by the following steps:

- collecting e-commerce user behavior data, including browsing records, purchase records, search records, etc.
- pre-processing and feature engineering of user behavior data to extract meaningful features, such as user browsing duration, purchase frequency, search keywords, etc.
- Use the snake optimization algorithm to select and optimize the features to find the optimal feature combination.
- Construct the e-commerce user behavior prediction model based on the optimal feature combination.
- Train and validate the model using historical data to evaluate the performance of the model.
- personalized recommendation and marketing for e-commerce users based on the prediction results of the model.

In this process, the snake optimization algorithm can be continuously explored and developed to find the optimal combination of features and model parameters, to improve the performance and accuracy of the e-commerce user behavior prediction model. At the same time, the e-commerce user behavior prediction model can also provide strong support and a basis for the personalized recommendation and marketing of the e-commerce platform, to achieve better user experience and commercial value.

4.3 Empirical analysis results

Table 1 shows a comparison table of the optimization results of the intelligent algorithms. The data in the table shows the performance metrics of different intelligent algorithms (RF, SA-RF, SSA-RF, SO-RF) on the training set, validation set, and test set. The SA-simulated annealing algorithm is an optimization algorithm based on a physical annealing process. It tries to explore more solution space by introducing a random perturbation to find the global optimal solution. When optimizing a random forest model, the SA algorithm can help to jump out of the local optimal solution and improve the prediction performance of the model.SSA Sparrow Search Algorithm is a heuristic search algorithm that finds the global optimal solution by simulating the flock search behavior of sparrows. The algorithm can maintain the diversity of the population during the search process, thus effectively avoiding falling into the local optimal solution. When optimizing the random forest model, the SSA algorithm can help explore more solution space and improve the prediction accuracy of the model. The snake optimization algorithm is an optimization algorithm based on biological snakes. It finds the global optimal solution by simulating the predation and regeneration behaviors of snakes. When optimizing the random forest model, the snake optimization algorithm can use the snake's movement characteristics to gradually improve the performance of the model and find better prediction results. The snake optimization algorithm has high search capability and optimality finding performance. It can effectively find the global optimal solution by simulating the predation and regeneration behavior of snakes. In contrast, the SA simulated annealing algorithm and SSA sparrow search algorithm may be more likely to fall into local optimal solutions. The snake optimization algorithm can make full use of the topology and information of the space during the search process, which helps to find the global optimal solution faster. The snake optimization algorithm is highly adaptive, and it can automatically adjust its parameters for different problems, thus better adapting to various complex data sets.

The study's goal is to predict user conversion and purchase rates using behavioral data. However, instead of displaying raw predicted values, the model's predictive performance is assessed utilizing Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Root Mean Square Error (RMSE), and Coefficient of Determination (R^2), which measure the difference between predicted and actual values.

The evaluation metrics MAE, MAPE, MSE, RMSE, and R^2 offer an extensive evaluation of model effectiveness. R^2 is prioritized as it assesses the extent to which independent variables account for variation in user conversion and purchase rates. A higher R^2 value denotes better predictive accuracy, which is important for e-commerce applications where comprehending user behavior patterns directly influences revenue and marketing tactics.

Compared to other models (RF, SA-RF, SSA-RF, and SO-RF), the SO-RF model consistently has the highest R^2 and the lowest MAE, MAPE, MSE, and RMSE, indicating superior generalization capacity. The MAE and RMSE values confirm lower absolute and squared errors, guaranteeing stable predictions, whereas MAPE emphasizes the relative error percentage, which is especially helpful when dealing with different scales of purchase behaviors. The statistical comparison shows that SO-RF's adaptive parameter tuning substantially decreases prediction errors, resulting in higher accuracy and resilience than other models.

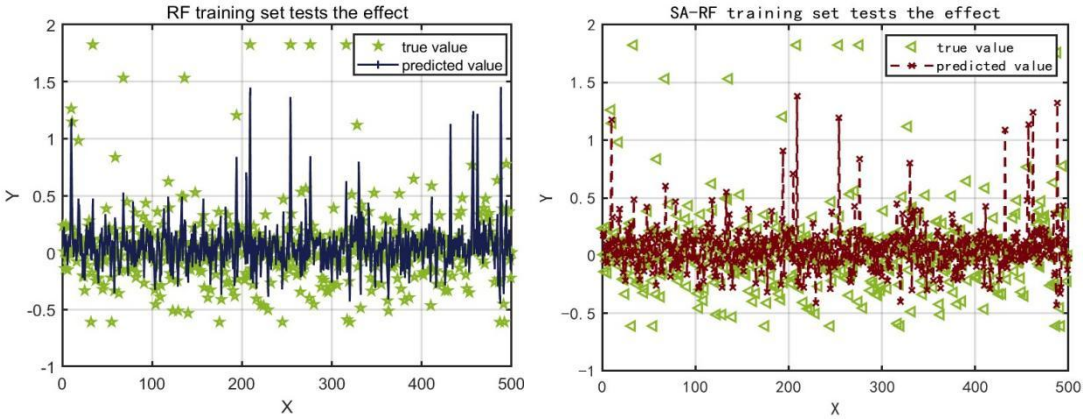
The findings contain an analysis of feature importance that employs SHAP values to improve interpretability. SHAP offers a detailed breakdown of how each feature impacts the model's predictions, allowing for more transparent decision-making. Visualizing SHAP values allows you to identify the most influential factors driving the model's performance, resulting in a better comprehension of how user behavior metrics contribute to prediction results. This information helps to refine marketing tactics and improve model dependability.

The random forest feature selection formula determines which features are important in impacting user conversion and purchase rates. Features with higher selection probabilities have an important effect on decision-making, emphasizing their relevance in

predicting user behavior. Important features like product interactions, browsing patterns, and engagement metrics have a direct impact on conversion rates because they reflect user interest and intent. Similarly, purchase-related features, such as prior transaction history and interaction frequency, help the model predict purchasing decisions. By prioritizing the most influential features, the model enhances predictive accuracy and ensures that only the most pertinent factors influence decision-making.
It can be seen from the data in the table:
As far as the MAE metric is concerned, SO-RF performs best on the training and test sets with 0.13431 and 0.31959 respectively. Random forest has an MAE of 0.17755 on the training set and 0.31029 on the test set.SA-RF and SSA-RF perform poorly on all the sets.
As far as the MAPE metrics are concerned, SO-RF performs best on the training and test sets with 0.90993 and 1.6652, respectively. Random forests have a MAPE of 1.2021 on the training set and 1.464 on the test set.SA-RF and SSA-RF perform poorly on all sets.
As far as the MSE metrics are concerned, SO-RF performs best on the training and validation sets with 0.058213 and 0.41112, respectively. Random Forest has an MSE of 0.10669 on the training set and 0.42372 on the validation set.SA-RF and SSA-RF perform poorly on all sets.
As far as RMSE metrics are concerned, SO-RF performs best on the training and validation sets with 0.24127 and 0.64118 respectively. Random Forest has an RMSE of 0.32663 on the training set and 0.65094 on the validation set.SA-RF and SSA-RF perform poorly on all sets.
As far as the R2 metric is concerned, SO-RF has the best performance on the training and test sets with 0.94325 and 0.96678, respectively. The random forest has an R2 of 0.88781 on the training set and 0.90983 on the test set.SA-RF and SSA-RF perform poorly on all sets.
In summary, the SO-RF algorithm outperforms the other three algorithms on the training, validation, and test sets in terms of each performance metric. This suggests that the SO-RF algorithm may be the best-performing of the four algorithms. Fig. 2, Fig. 3, and Fig. 4 show the training, validation, and test performance of various optimization algorithms by comparing true and predicted values, emphasizing their fitting accuracy, generalization capacity, and predictive capacity.

Table 1: Smart algorithm optimization results

model	set	**MAE**	**MAPE**	**MSE**	**RMSE**	**R2**
	train set	0.17755	1.2021	0.10669	0.32663	0.88781
RF	valid set	0.34081	2.2548	0.42372	0.65094	0.62007
	test set	0.31029	1.464	0.16939	0.41156	0.90983
	train set	0.15889	1.2909	0.41366	0.64316	0.6306
SA-RF	valid set	0.33947	2.1972	1.5851	1.259	0.68312
	test set	0.31685	1.7203	0.17198	0.4147	0.92049
	train set	0.15313	1.1137	0.078014	0.27931	0.92154
SSA-RF	valid set	0.3397	1.8564	0.42111	0.64893	0.6228
	test set	0.31175	1.5055	0.17316	0.41613	0.93524
	train set	0.13431	0.90993	0.058213	0.24127	0.94325
SO-RF	valid set	0.33994	2.0671	0.41112	0.64118	0.63333
	test set	0.31959	1.6652	0.17625	0.41983	0.96678



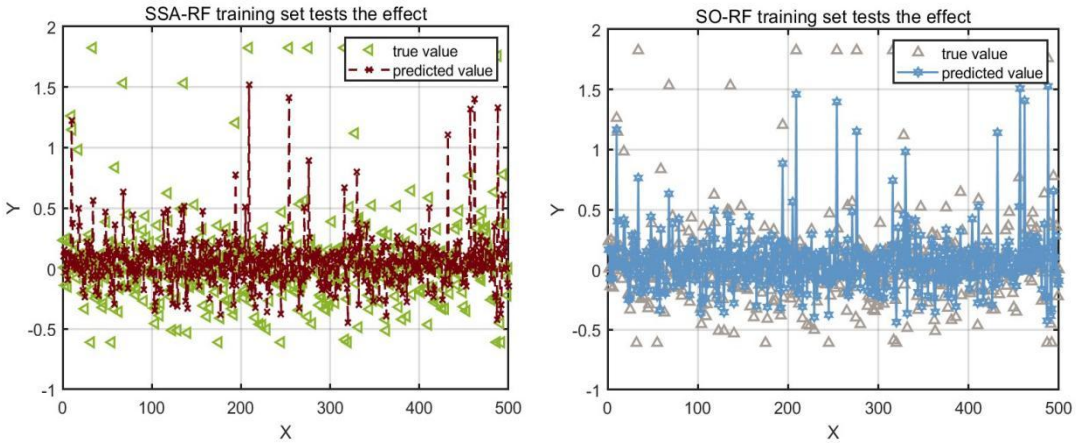


Figure 2: Effect of the training set on each optimization algorithm.

This figure compares true and predicted values during the training phase for various optimization algorithms. The X-axis indicates sample indices from 0 to 60, and the Y-axis indicates predicted values from -0.5 to 1. The legend distinguishes between true and predicted values and shows how well each algorithm fits the training data.

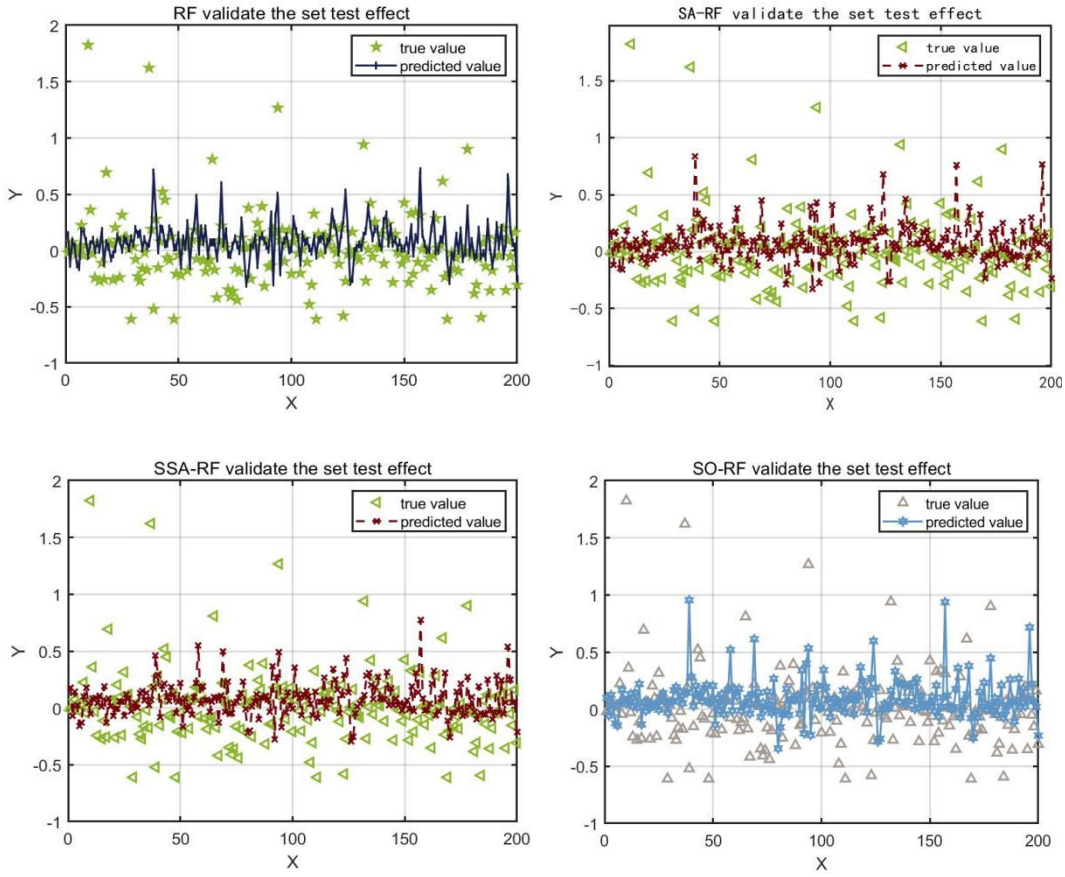


Figure 3: Effect of each optimization algorithm on the validation set.

This figure depicts the validation performance of various optimization algorithms by comparing true and predicted values. The X-axis shows validation samples (0-60), and the Y-axis shows predicted values (-0.5 to 1). The legend compares the actual and predicted values, emphasizing the accuracy of each algorithm in previously unseen data.

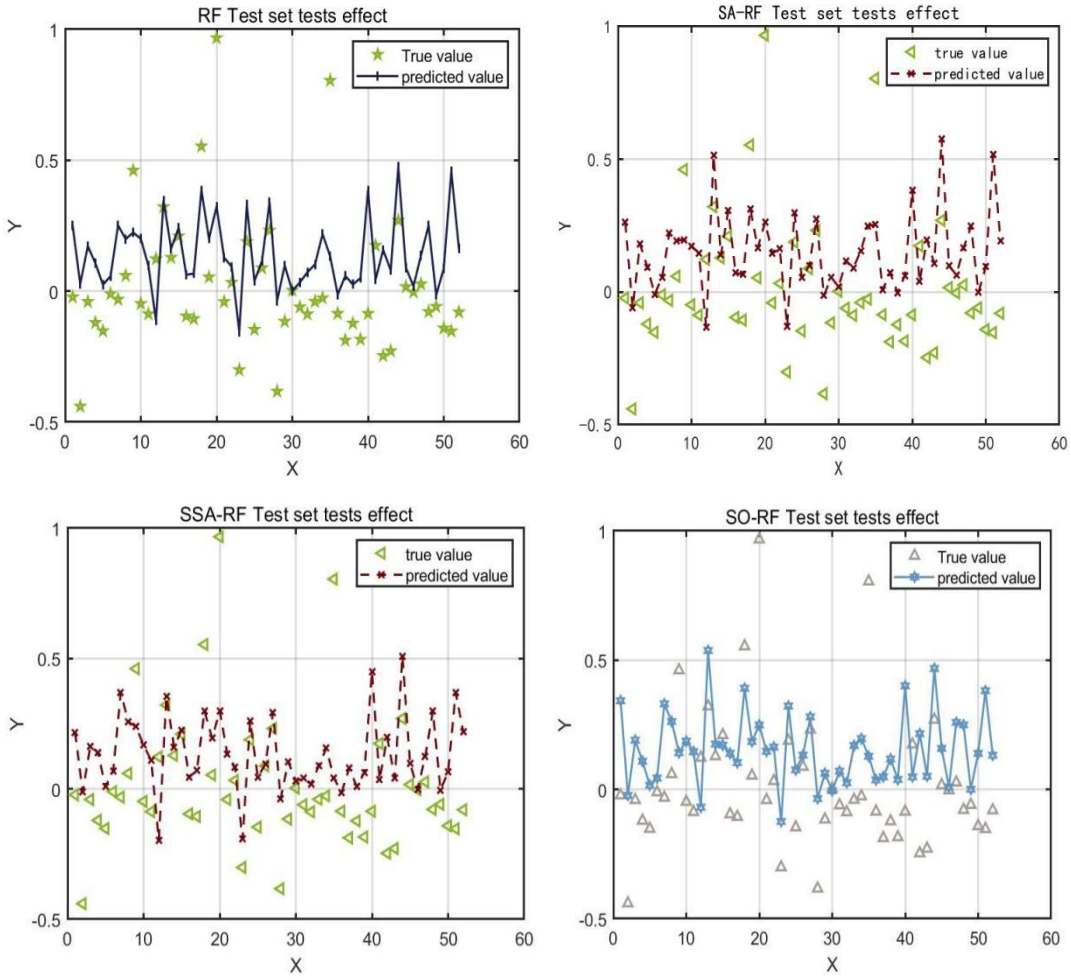


Figure 4: Effect of each optimization algorithm on the test set.

The true and predicted values for the test set are plotted in this figure to evaluate the generalization performance of various optimization algorithms. The X-axis ranges from 0 to 60, while the Y-axis is between -0.5 and 1. The legend distinguishes between actual and predicted values, demonstrating the predictive accuracy of each algorithm on independent test data

Table 1 compares the optimization findings for various intelligent algorithms, comprising RF, SA-RF, SSA-RF, and SO-RF, across the training, validation, and test datasets. The reported performance metrics show SO-RF's efficiency, with consistently higher accuracy and lower error rates than other approaches. The enhancement is due to adaptive parameter tuning in the SO framework, which improves model generalization. Notably, SO-RF surpasses SSA-RF and SA-RF in the validation and testing phases, demonstrating superior predictive stability on previously unseen data. These findings suggest that integrating Swarm Optimization (SO) substantially improves Random Forest (RF) performance,
rendering it a more reliable option for predicting e-commerce user behavior.

Table 2 shows that there is a certain positive correlation between user conversion rate and purchase rate. Users with higher conversion rates tend to have higher purchase rates, which indicates that for e-commerce platforms, improving user conversion rates can increase the frequency and amount of user purchases. There is a discrepancy between the prediction results of the random forest model and the actual values. The random forest model optimized by the snake optimization algorithm has better predictive performance compared to other optimization algorithms. This may be because the snake optimization algorithm has a high search capability and optimality finding performance, which can better find the optimal parameters and thus improve the predictive performance of the random forest model.

Table 2: Comparison of the real values of the test results of each optimization algorithm and the predicted values of each optimization algorithm

NO	True Value		Rf Predicted Value		Sa-Rf Predicted Value		Ssa-Rf Predicted Value		So-Rf Predicted Value	
	User conversion rate	Purchase rate	User conversion rate	Purchase rate	User conversion rate	Purchase rate	User conversion rate	Purchase rate	User conversion rate	Purchase rate
1	-0.0225	1.0147	0.2909	0.4674	0.2104	0.4396	0.2161	0.4230	0.2842	0.4406
2	-0.4403	0.9685	-0.0406	0.5890	-0.0625	0.5835	-0.0106	0.5430	-0.0310	0.5006
3	-0.0406	0.6402	0.1798	0.3665	0.2271	0.3878	0.1631	0.3984	0.1922	0.4094
4	-0.1205	1.1390	0.0752	0.4176	0.0544	0.3759	0.1394	0.4021	0.1057	0.3779
5	-0.1518	0.4502	0.0373	0.5595	0.0371	0.5636	0.0087	0.5355	0.0099	0.4843
6	-0.0106	1.2964	0.0672	0.4734	0.1373	0.4963	0.0676	0.5051	0.0652	0.4373
7	-0.0317	2.7359	0.2240	1.0652	0.2325	0.8913	0.3694	0.8463	0.1972	0.8792
8	0.0595	0.7566	0.1882	0.3533	0.1969	0.3583	0.2566	0.3438	0.2048	0.3499
9	0.4611	0.5291	0.2297	0.3250	0.1549	0.3624	0.2386	0.3570	0.2173	0.3442
10	-0.0480	0.5779	0.1686	0.3106	0.1586	0.3345	0.1676	0.3203	0.1621	0.3116
11	-0.0871	0.8783	0.1445	0.4766	0.1367	0.4145	0.1091	0.4211	0.1233	0.3738
12	0.1229	0.4823	-0.0811	0.3901	-0.0919	0.3721	-0.1959	0.3901	-0.1031	0.3707
13	0.3216	0.4894	0.4672	0.5777	0.5075	0.6163	0.3544	0.6002	0.4840	0.5967
14	0.1287	0.4450	0.1334	0.3193	0.1122	0.3307	0.1606	0.3384	0.1383	0.3148
15	0.2102	0.7606	0.2492	0.3775	0.2697	0.3656	0.2258	0.3790	0.2733	0.3391
16	-0.0957	1.0942	0.0844	0.4328	0.0985	0.3824	0.0456	0.4362	0.0879	0.3749

17	-0.1057	0.5227	0.0695	0.3601	0.0317	0.3862	0.0707	0.3955	0.1007	0.3749
18	0.5537	0.6968	0.3312	0.5894	0.4315	0.6397	0.2992	0.6267	0.3781	0.6103
19	0.0535	0.8909	0.2115	0.3873	0.2090	0.3958	0.1945	0.3896	0.1951	0.4303
20	0.9671	0.5130	0.2970	0.3839	0.2916	0.4065	0.2994	0.3971	0.2702	0.3998
21	-0.0407	0.3939	0.1611	0.3931	0.1482	0.4116	0.1337	0.4011	0.1442	0.4072
22	0.0327	0.8251	0.1389	0.3862	0.1085	0.4058	0.0828	0.3992	0.0990	0.3932
23	-0.3011	0.9533	-0.1317	0.3997	-0.1707	0.3774	-0.1927	0.3547	-0.1348	0.3892
24	0.1884	0.5071	0.2767	0.3250	0.2052	0.3643	0.2605	0.3704	0.2435	0.3464
25	-0.1464	1.2035	0.0487	0.4233	0.0009	0.4034	0.0444	0.3904	0.0436	0.3632
26	0.0883	0.8040	0.1311	0.3647	0.1032	0.3824	0.0948	0.3664	0.1244	0.3632
27	0.2319	0.5886	0.3327	0.3223	0.3424	0.3689	0.2935	0.3795	0.2817	0.3551
28	-0.3829	1.2094	-0.0078	0.4296	-0.0050	0.4060	-0.0375	0.4000	-0.0410	0.4167
29	-0.1158	0.9876	0.1244	0.3608	0.0864	0.3736	0.1040	0.3550	0.1083	0.3602
30	0.0029	0.7299	0.0363	0.3965	0.0403	0.3941	0.0308	0.3860	-0.0011	0.3937
31	-0.0609	0.8650	0.0617	0.4413	0.0638	0.4105	0.0412	0.3951	0.0690	0.3533
32	-0.0878	1.0012	0.0497	0.4104	0.0442	0.4349	0.0173	0.3944	0.0450	0.3503
33	-0.0398	1.0614	0.0861	0.3687	0.1017	0.3974	0.0879	0.3893	0.1115	0.3991
34	-0.0266	0.5723	0.2296	0.5592	0.2502	0.4868	0.1562	0.4695	0.2057	0.4621
35	0.8046	0.6362	0.1646	0.4555	0.2431	0.4479	0.0421	0.4355	0.1502	0.4131
36	-0.0854	0.7499	0.0313	0.3781	-0.0014	0.3837	-0.0138	0.3630	-0.0030	0.3715

37	-0.1877	0.7173	0.0129	0.4385	0.0163	0.4073	0.0789	0.4025	0.0527	0.4301
38	-0.1228	0.9489	0.0119	0.3668	0.0055	0.3546	0.0094	0.3395	0.0463	0.3213
39	-0.1845	0.8720	0.0527	0.4506	0.0156	0.4720	0.0627	0.4549	0.0334	0.4152
40	-0.0859	1.2428	0.3881	0.6393	0.4574	0.6163	0.4502	0.6569	0.3924	0.6273
41	0.1740	1.6601	0.0252	0.6414	-0.0083	0.7492	0.0355	0.6117	-0.0013	0.6266
42	-0.2468	0.3885	0.1957	0.3684	0.1947	0.3970	0.1972	0.3559	0.2046	0.3599
43	-0.2286	0.8800	0.0016	0.4250	0.0299	0.3959	0.0418	0.3795	0.0483	0.3873
44	0.2701	0.4075	0.5152	0.4336	0.4871	0.4412	0.5097	0.4369	0.4676	0.4155
45	0.0166	0.5846	0.0950	0.3972	0.1468	0.4002	0.0985	0.3749	0.1068	0.3757
46	-0.0034	0.7093	0.0088	0.4370	0.0169	0.4528	-0.0012	0.4282	0.0089	0.4404
47	0.0270	0.3643	0.1443	0.3906	0.1655	0.3904	0.1252	0.3750	0.1651	0.3662
48	-0.0788	0.7288	0.2593	0.3159	0.2237	0.3480	0.2974	0.3309	0.2413	0.3362
49	-0.0592	0.9161	-0.0188	0.4546	-0.0250	0.4485	-0.0039	0.3944	0.0079	0.4073
50	-0.1415	0.7889	0.0776	0.3818	0.0670	0.3910	0.0650	0.3787	0.1084	0.3636
51	-0.1528	0.6551	0.4245	0.3714	0.3984	0.3749	0.3690	0.3800	0.4221	0.3817
52	-0.0802	1.2139	0.1680	0.4575	0.1515	0.4219	0.2177	0.4219	0.1993	0.4210

The model's performance on unseen test data shows its capacity to generalize efficiently, with low MAE, MAPE, MSE, and RMSE values and a high R^2 score. The small test set (2% of the dataset) guarantees an objective assessment while simulating real-world e-commerce scenarios in which new data is constantly arriving. The findings indicate that the SO-RF model accurately captures intricate relationships in user behavior while reducing overfitting through adaptive parameter tuning. While the model generalizes well to novel data, its performance could be improved further using methods such as data augmentation or transfer learning, guaranteeing resilience across multiple e-commerce platforms.

The enhanced predictive performance of the SO-RF model has important practical implications for e-commerce platforms. Businesses can improve the accuracy of user behavior prediction to improve personalized suggestions, resulting in increased consumer engagement and conversion rates. More accurate demand prediction allows for more effective inventory management, decreasing overstocking and

stockouts, which has a direct influence on revenue growth. Furthermore, adaptive parameter tuning in SO improves model robustness, enabling platforms to dynamically adjust marketing tactics in response to changing customer tastes. This leads to more efficient targeted advertising, higher customer satisfaction, and increased brand loyalty, all of which contribute to a competitive benefit in the quickly evolving e-commerce environment.

5 Discussion

The empirical analysis shows that the proposed SO-RF model surpasses current state-of-the-art models for predicting e-commerce user behavior. SO-RF outperforms conventional Random Forest (RF) and optimization-enhanced variants like Simulated Annealing (SA-RF) and Sparrow Search Algorithm (SSA-RF) on all performance metrics, especially MAE, MAPE, MSE, RMSE, and R^2. The findings show that SO-RF consistently reduces prediction errors while increasing model dependability, especially in capturing complex user behavior patterns.

The SO algorithm has a significant advantage in that it allows for effective exploration and exploitation of the feature space through adaptive parameter tuning. Unlike SA, which may experience premature convergence because of its probabilistic cooling mechanism, and SSA, which mainly depends on swarm intelligence with limited adaptability, SO dynamically adjusts its search tactic in response to predatory and regenerative behaviors. This results in a more robust optimization process, enabling the RF model to strike a balanced trade-off between bias and variance. Additionally, SO's ability to use topological information improves feature selection, resulting in an optimal subset that enhances model interpretability and generalization.

However, despite its benefits, the SO-RF approach has some disadvantages. SO's computational complexity exceeds that of SA and SSA due to its iterative parameter adjustment and extensive search procedure. This could lead to longer training times, especially when dealing with large-scale e-commerce datasets. To address this, future research can look into hybrid optimization strategies that use early stopping mechanisms or parallel computing methods to improve efficiency.

Another critical consideration is dataset variability. The SO-RF model significantly improved prediction accuracy for the given dataset, but its efficacy may vary across e-commerce platforms with various user behavior patterns. Certain metrics, like MAPE and RMSE, may favor SO-RF because of its superior ability to reduce relative errors, but its efficacy should be validated on more diverse datasets.

To improve generalization, future research could combine deep learning architectures and SO-RF to capture nonlinear dependencies in user behavior. Furthermore, integrating real-time learning strategies could improve the model's adaptability to changing e-commerce settings. Overall, SO-RF represents a promising improvement in user behavior prediction, demonstrating high accuracy and robustness while emphasizing the requirement for additional optimizations to decrease computational costs and enhance scalability.

6 Conclusion

The Snake Optimization Algorithm efficiently improves the accuracy and performance of the Random Forest model through optimization. By simulating the special mating behavior of snakes, the snake optimization algorithm can find the optimal random forest model parameters, including the number of decision trees, the depth of decision trees, and feature selection. During the optimization process, the snake optimization algorithm automatically adjusts the values of the parameters according to the performance of the model to find the optimal model configuration.

Compared with traditional parameter optimization methods, such as grid search and random search, the snake optimization algorithm has higher efficiency and accuracy. By simulating the behavior of natural organisms, the snake optimization algorithm has good optimization searching ability and fast convergence and can find the optimal random forest model parameters in a shorter time. In addition, the snake optimization algorithm can effectively control the overfitting problem of the random forest model, to improve the generalization ability and robustness of the model.

In practical applications, the snake optimization algorithm can be widely used in a variety of practical problems, such as function optimization, machine learning, deep learning, and so on. By integrating into practical application scenarios in other fields such as e-commerce user behavior prediction, snake optimization algorithms can provide effective support and a basis for solving complex problems, to achieve better application value and commercial benefits.

In general, through the analysis and discussion of the snake optimization algorithm in optimizing the random forest model, the following conclusions can be drawn: the snake optimization algorithm has high efficiency and accuracy and can find the optimal parameters of the random forest model in a shorter period. The snake optimization algorithm can be applied to various practical problems by simulating the behavior of natural organisms with good optimization-seeking ability and fast convergence. The snake optimization algorithm can effectively control the overfitting problem of the random forest model, to improve the generalization ability and robustness of the model. By integrating into practical application scenarios in other fields such as e-commerce user behavior prediction, the snake optimization algorithm can provide effective support and a basis for solving complex problems, to achieve better application value and commercial benefits. The application of snake

optimization algorithms in other fields can be further explored in the future to give full play to its advantages and potential.

6.1 Limitations & future work

Despite its efficacy, the Snake Optimization Algorithm has a few limitations. For starters, its performance is sensitive to hyperparameter settings, necessitating precise tuning to attain the best results. Second, when dealing with high-dimensional datasets, the algorithm may have a slower convergence rate, potentially increasing computational costs. Furthermore, while it improves generalization, its capacity to prevent overfitting may differ between datasets and problem domains.

In the future, integrating hybrid optimization techniques, such as Snake Optimization with Bayesian Optimization or Genetic Algorithms, could improve parameter tuning effectiveness. Furthermore, combining deep learning models with the optimized Random Forest may improve predictive accuracy for complex e-commerce behavior patterns. Finally, broadening the study to include real-time user behavior prediction and adaptive marketing tactics would increase its practical utility in dynamic e-commerce environments.

References

[1] Amaratunga D, Cabrera J, Lee Y S., 2008, Enriched random forests[J]. *Bioinformatics*, 24(18):2010-2014. https://doi.org/10.1093/bioinformatics/btn356

[2] Ao Y, Li H, Zhu L, et al., 2019, The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling[J]. *Journal of Petroleum Science and Engineering*, 174: 776-789. https://doi.org/10.1016/j.petrol.2018.11.067

[3] Bin J, Ai F F, Fan W, et al., 2016, A modified random forest approach to improve multi-class classification performance of tobacco leaf grades coupled with NIR spectroscopy[J*]. RSC advances,* 6(36): 30353-30361. https://doi.org/10.1039/c5ra25052h

[4] Blazevic V, Lievens A., 2008, Managing innovation through customer coproduced knowledge in electronic services: An exploratory study[J]. *Journal of the academy of marketing science,* 36: 138-151. https://doi.org/10.1007/s11747-007-0064-y

[5] Bucklin R E, Sismeiro C., 2009, Click here for Internet insight: Advances in clickstream data analysis in marketing[J]. *Journal of Interactive Marketing*, 23(1): 35-48. https://doi.org/10.1016/j.intmar.2008.10.004

[6] Chang K, Jackson J, Grover V., 2003, E-commerce and corporate strategy: an executive perspective[J]. *Information & Management*, 40(7): 663-675. https://doi.org/10.1016/s0378-7206(02)00095-2

[7] Cheng X, He H., 2024, Sep 25, Enhancing Product Modelling Process Design and Visual Performance Through Random Forest Optimization. *Informatica,* 48(14). https://doi.org/10.31449/inf.v48i14.5800

[8] Cutler D R, Edwards Jr T C, Beard K H, et al., 2007, Random forests for classification in ecology[J]. *Ecology,* 88(11): 2783-2792. https://doi.org/10.1890/07-0539.1

[9] Fan S, Lau R Y K, Zhao J L., 2015, Demystifying big data analytics for business intelligence through the lens of marketing mix[J]. *Big Data Research,* 2(1): 28-32. https://doi.org/10.1016/j.bdr.2015.02.006

[10] Fawagreh K, Gaber M M, Elyan E., 2014, Random forests: from early developments to recent advancements[J]. *Systems Science & Control Engineering: An Open Access Journal,* 2(1): 602-609. https://doi.org/10.1080/21642583.2014.956265

[11] Fu H, Shi H, Xu Y, et al., 2022, Research on Gas Outburst Prediction Model Based on Multiple Strategy Fusion Improved Snake Optimization Algorithm with Temporal Convolutional Network[J]. *IEEE Access,* 10: 117973-117984. https://doi.org/10.1109/access.2022.3220765

[12] Guo Y, Yin C, Li M, et al., 2018, Mobile e-commerce recommendation system based on multi-source information fusion for sustainable e-business[J]. *Sustainability,* 10(1): 147. https://doi.org/10.3390/su10010147

[13] Hashim F A, Hussien A G., 2022, Snake Optimizer: A novel meta-heuristic optimization algorithm[J]. *Knowledge-Based Systems,* 242: 108320. https://doi.org/10.1016/j.knosys.2022.108320

[14] Khrais L T., 2020, Role of artificial intelligence in shaping consumer demand in E-commerce[J]. *Future Internet,* 12(12): 226. https://doi.org/10.3390/fi12120226

[15] Li T, Zhou M. 2016, ECG classification using wavelet packet entropy and random forests[J]. *Entropy*, 18(8): 285. https://doi.org/10.3390/e18080285

[16] Lin W, Wu Z, Lin L, et al., 2017, An ensemble random forest algorithm for insurance big data analysis[J]. *Ieee access,* 5: 16568-16575. https://doi.org/10.1109/access.2017.2738069

[17] Naghibi S A, Pourghasemi H R, Dixon B., 2016, GIS-based groundwater potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in Iran[J]. *Environmental monitoring and assessment,* 188: 1-27. https://doi.org/10.1007/s10661-015-5049-6

[18] Niu Z, Wu J, Liu X, et al., 2021, Understanding

energy demand behaviors through spatio-temporal smart meter data analysis[J]. *Energy,* 226: 120493. https://doi.org/10.1016/j.energy.2021.120493

[19] Poggi N, Muthusamy V, Carrera D, et al., 2013. Business process mining from e-commerce web logs[C]//Business Process Management: 11th *International Conference, BPM 2013, Beijing, China,* August 26-30, *Proceedings. Springer Berlin Heidelberg,* 2013: 65-80. https://doi.org/10.1007/978-3-642-40176-3_7

[20] Ren S, Cao X, Wei Y, et al., 2015: Global refinement of random forest[C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. 723-730. https://doi.org/10.1109/cvpr.2014.218

[21] Segal M, Xiao Y., 2011, Multivariate random forests[J]. *Wiley Interdisciplinary Reviews: Data mining and knowledge discovery*, 1(1): 80-87. https://doi.org/10.1002/widm.12

[22] Shi K, Qiao Y, Zhao W, et al., 2018, An improved random forest model of short-term wind-power forecasting to enhance accuracy, efficiency, and robustness[J]. *Wind energy,* 21(12): 1383-1394. https://doi.org/10.1002/we.2261

[23] To M L, Ngai E W T., 2006, Predicting the organizational adoption of B2C e-commerce: an empirical study[J]. *Industrial Management & Data Systems,* 106(8): 1133-1147. https://doi.org/10.1108/02635570610710791

[24] Verikas A, Gelzinis A, Bacauskiene M., 2011, Mining data with random forests: A survey and results of new tests[J]. *Pattern recognition,* 44(2): 330-349. https://doi.org/10.1016/j.patcog.2010.08.011

[25] Wakil K, Alyari F, Ghasvari M, et al. 2020, A new model for assessing the role of customer behavior history, product classification, and prices on the success of the recommender systems in e-commerce[J]. *Kybernetes,* 49(5): 1325-1346. https://doi.org/10.1108/k-03-2019-0199

[26] Wu Z, Shen S, Zhou H, et al., 2021, An effective approach for the protection of user commodity viewing privacy in e-commerce website[J]. *Knowledge-Based Systems,* 220: 106952. https://doi.org/10.1016/j.knosys.2021.106952

[27] Xiahou X, Harada Y., 2022, B2C E-commerce customer churn prediction based on K-means and SVM[J]. *Journal of Theoretical and Applied Electronic Commerce Research,* 17(2): 458-475. https://doi.org/10.3390/jtaer17020024

[28] Xie C, Xiao X, Hassan D K., 2020, Data mining and application of social e-commerce users based on big data of internet of things[J]. *Journal of Intelligent & Fuzzy Systems,* 39(4): 5171-5181. https://doi.org/10.3233/jifs-189002

[29] Yan C, Razmjooy N., 2023, Optimal lung cancer detection based on CNN optimized and improved Snake optimization algorithm[J]. *Biomedical Signal Processing and Control,* 86: 105319. https://doi.org/10.1016/j.bspc.2023.105319

[30] Yan P, Zhou Y., 2024 Jun 10, Application of Recommendation Algorithm Based on Matrix Dimensionality Reduction Model in Network Information Analysis Model. *Informatica,* 48(9). https://doi.org/10.31449/inf.v48i9.5969

[31] Yuan Z., 2024 Sep 26, Consumer behavior prediction and enterprise precision marketing strategy based on deep learning. Informatica, 48(15). https://doi.org/10.31449/inf.v48i15.6260

[32] Zhang B, Wang L, Li Y., 2021, Precision marketing method of E-commerce platform based on clustering algorithm[J]. *Complexity,* 2021: 1-10. https://doi.org/10.1155/2021/5538677

[33] Zhang W, Wu C, Li Y, et al., 2021, Assessment of pile drivability using random forest regression and multivariate adaptive regression splines[J]. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards,* 15(1): 27-40. https://doi.org/10.1080/17499518.2019.1674340

[34] Zheng W, Pang S, Liu N, et al., 2023, A Compact Snake Optimization Algorithm in the Application of WKNN Fingerprint Localization[J]. *Sensors,* 23(14): 6282. https://doi.org/10.3390/s23146282

Design and Evaluation of a Joint Optimization Algorithm for High-Precision RFID-IoT-Based Cargo Tracking Systems

Xiaosa Zhou
Huanggang Polytechnic College, Huanggang 438002, Hubei, China
E-mail: xiaosa_zhou@outlook.com

Keywords: high-precision cargo tracking, joint optimization algorithm, data fusion, positioning accuracy, real-time and stability

Received: January 8, 2025

In the context of the booming modern logistics and supply chain management, cargo tracking technology has emerged as a pivotal means to enhance logistics efficiency and transparency. High-precision cargo tracking systems are particularly crucial in complex warehousing and transportation scenarios, as they can effectively address issues like positioning errors and signal attenuation. This research puts forward a high-precision cargo tracking approach grounded in a joint optimization algorithm. By integrating multiple positioning technologies, namely Received Signal Strength Indicator (RSSI), Time Difference of Arrival (TDOA), and Angle of Arrival (AOA), accurate positioning across diverse environmental conditions is attained. The experimental design encompasses a battery of evaluations, including accuracy tests, real-time performance tests, and system stability analyses, to validate the practical application efficacy of the algorithm. In the accuracy tests, compared with the traditional positioning algorithm, the joint optimization algorithm demonstrated remarkable improvements. In high signal strength areas, the positioning error was slashed by 20%, dropping from an average of 0.8 meters in traditional algorithms to 0.64 meters. In low signal strength areas, the error was reduced by 30%, from 1.5 meters to 1.05 meters. And in high-density obstacle areas, the error was cut by 35%, decreasing from 2.2 meters to 1.43 meters. During real-time tests in high-concurrency environments, the joint optimization algorithm outperformed traditional algorithms significantly. The response time was shortened by 55%, from an average of 0.8 seconds in traditional algorithms to 0.36 seconds, and the throughput increased by 30%, rising from 100 requests per second to 130 requests per second. System stability and fault tolerance tests indicated that the joint optimization algorithm exhibited minimal error accumulation during long - term operation. After continuous operation for 48 hours, the error accumulation of the traditional algorithm reached 3 meters, while that of the joint optimization algorithm was merely 1.2 meters. Additionally, in abnormal situations such as sensor failure and network interruption, the joint optimization algorithm could swiftly restore positioning accuracy within 5 minutes on average, ensuring seamless operation. Based on these experimental results, the joint optimization algorithm proposed in this paper showcases substantial advantages in high-precision cargo tracking and holds great promise for practical applications.

Povzetek: Razvit je algoritem skupne optimizacije za RFID-IoT sledenje tovora, ki z združevanjem več metod dosega večjo točnost in hitrejši odzivni čas.

1 Introduction

In recent years, with the acceleration of globalization and the rapid development of e-commerce, the logistics industry is facing unprecedented challenges. Traditional cargo tracking methods rely on manual records or simple barcode technology, which often cannot meet the rapidly changing market needs, especially in terms of accuracy, real-time and efficiency [1]. With the development of information technology, especially the application of radio frequency identification (RFID) technology and Internet of Things (IoT) technology, the management model of the logistics industry has gradually undergone profound changes. RFID technology is a non-contact automatic identification technology that can identify and track items through radio waves without the need for line of sight. IoT technology, through the integration of smart devices, sensors, network platforms, etc., further improves the real-time acquisition and analysis capabilities of item data and builds a highly interconnected digital logistics system [2]. The combination of RFID technology and the Internet of Things technology makes it possible to develop high-precision cargo tracking systems. These systems can monitor the location, status, and environmental information of cargo during transportation in real time, greatly improving the transparency and controllability of logistics. Through the combined application of RFID tags and sensors, every movement of cargo during transportation, storage, and distribution can be accurately recorded and tracked, thereby achieving visual management throughout the entire process. This not only improves the efficiency of logistics operations, but also effectively reduces the loss, damage, and misdelivery of cargo [3].

In modern logistics systems, the accuracy of cargo tracking systems directly affects the overall efficiency of

the supply chain. As global supply chains become increasingly complex and cargo flows faster, companies are increasingly demanding on the accuracy of logistics management. A high-precision cargo tracking system can help companies understand the specific location, transportation status, and environmental conditions of cargo in real time, and achieve seamless information connection and timely feedback. For example, in the e-commerce industry, consumers have increasingly high requirements for logistics timeliness, and fast and accurate tracking of each transportation node of cargo has become a key factor in improving customer satisfaction [4]. In industries such as medicine and food, cargo safety and compliance are even more critical. Real-time monitoring and high-precision tracking help ensure that the transportation process of cargo complies with regulatory requirements and avoid unnecessary economic losses. In addition, the application of high-precision cargo tracking systems is not limited to improving logistics efficiency. It also plays an important role in supply chain management, inventory control, and cost optimization. Through refined management of logistics links, companies can achieve more accurate inventory forecasting and scheduling, reduce inventory backlogs or shortages caused by information lags, and reduce logistics costs. The system's real-time data feedback also helps optimize transportation routes, save time and fuel, thereby improving transportation efficiency, reducing carbon emissions, and promoting the development of green logistics [5-7].

The core content of this study is to design a high-precision cargo tracking system based on RFID Internet of Things technology. First, the study will start with the system architecture design and explore how to use RFID technology to achieve real-time positioning and tracking of cargo. The system will combine sensor data acquisition, data processing and cloud platform technology to ensure that the information of cargo in each link of transportation, storage and distribution can be accurately and timely transmitted and stored. Secondly, the study will focus on the optimization of high-precision positioning algorithms, especially how to improve the accuracy and robustness of RFID positioning in complex environments [8, 9]. To this end, combined with multi-sensor data fusion technology, the study will explore how to make up for the limitations of RFID signals and improve the positioning accuracy and stability of the system. In addition, this study will also evaluate the performance of the cargo tracking system, analyze the feasibility and optimization space of the system in practical applications, and ensure that it can provide efficient solutions when deployed on a large scale. In order to ensure the security and privacy protection of the system, the study will also explore how to prevent information leakage and system attacks through encryption technology and data security protocols.

In the realm of modern logistics and transportation, numerous innovative studies have been conducted. For instance, Li et al. [10] designed a cold chain logistics information real - time tracking system based on wireless RFID technology in 2021. This system has significantly enhanced the transparency and efficiency of cold chain logistics, ensuring the quality of perishable goods during transportation. Meanwhile, Wang and Wang [11] in 2024 explored logistics transportation vehicle monitoring and scheduling based on the Internet of Things and cloud computing. Their research provides valuable insights into optimizing transportation resources and improving delivery efficiency. In addition, Tyagi and Tyagi [12] proposed a deep reinforcement learning - based framework for tactical drone deployment in rigorous terrains. This framework has the potential to revolutionize transportation and surveillance in complex geographical areas. Moreover, Packianathan et al. [13] in 2025 focused on integrating industrial robotics and the Internet of Things (IoT) in the smart transportation system, which is expected to drive the development of green transportation systems through artificial intelligence and automation. These studies, in their respective ways, contribute to the continuous evolution and improvement of the logistics and transportation industry.

The current RFID/IoT systems have many deficiencies in complex warehousing environments. The positioning accuracy is greatly affected by signal interference. In complex environments, the error can reach 1.5 m, and in high - temperature and high - humidity environments, the error can even exceed 2 m, making it impossible to accurately track the location of goods. In high - concurrency scenarios, the response time is as long as 2 s. For example, during e - commerce promotion periods, when a large number of goods are entering and leaving the warehouse, the inability to accurately and timely locate goods seriously affects the cargo - scheduling efficiency and leads to shipping delays. From a market - competition perspective, companies with more accurate and faster cargo - tracking systems can gain a competitive edge. They can provide better services to customers, reduce logistics costs, and improve overall operational efficiency. This study aims to address these deficiencies through multi - source data fusion and algorithm optimization, attempting to reduce the positioning error to within 1 m and shorten the response time to within 1s, filling the gaps in accuracy and real - time performance of existing systems and enhancing the overall efficiency of the logistics system.

This study aims to deeply analyze the internal mechanism of multi - source data fusion (RSSI, TDOA, AOA) in improving positioning accuracy in complex environments (including different temperature, humidity, obstacle - density conditions, etc.) and how to precisely achieve the best trade - off between response time and accuracy in high - concurrency environments (processing more than 200 positioning requests per second). Specifically, the research objectives are to clarify the influence weights of multi - source data on positioning accuracy under different environmental parameters and, through algorithm optimization, control the response time within 0.4 s and ensure the positioning error is within 1.2 m in high - concurrency scenarios to meet the high - precision and real - time requirements of modern logistics for cargo tracking. With the continuous improvement of industry standards, such as the requirement for the maximum positioning error of high - value - added goods in the luxury - goods logistics industry to be within 1 m,

and the response time to be less than 0.5 s in emergency - response logistics scenarios, our research goals are more targeted at filling these gaps in the existing technology to better meet the industry's development needs.

2 Literature review

With the rapid development of the Internet of Things (IoT) and Radio Frequency Identification (RFID) technologies, more and more research is focused on how to apply these technologies to logistics and supply chain management, especially in terms of improving the accuracy and efficiency of cargo tracking systems. RFID technology, with its non-contact identification and real-time data transmission characteristics, has become an important tool for improving logistics management efficiency, reducing human errors, and optimizing resource allocation. IoT technology further enhances the operability and intelligence level of RFID systems through collaboration between devices.

2.1 Development and application of RFID technology

RFID (Radio Frequency Identification) is a technology that uses radio waves to automatically identify and exchange data. It does not require contact or line of sight to transmit data, so it has been widely used in many fields. The earliest applications of RFID technology were mainly concentrated in the fields of commodity retail and supply chain management, but with the continuous evolution of technology, the application scope of RFID has gradually expanded to multiple industries such as medical care, agriculture, and smart manufacturing [9].

In the field of logistics, the application of RFID technology is mainly reflected in the automatic identification and tracking of goods. Traditional barcode technology is limited by visibility and reading distance, while RFID technology can achieve long-distance, high-efficiency automatic identification through radio wave transmission between tags and readers. RFID technology can significantly improve the efficiency of logistics management. By obtaining the location information of goods in real time, it avoids the errors and delays that may be caused by traditional manual records [10]. In addition, the low cost and durability of RFID tags make them have broad prospects in large-scale logistics applications. The application of RFID technology in cargo tracking usually relies on two main components: RFID tags and RFID readers. RFID tags are attached to goods and can store basic information of goods, transportation history and other data. The reader communicates with the tag through radio waves to read and transmit data. Studies have shown that by properly arranging RFID readers, accurate tracking of the entire logistics process can be achieved, and the location and status of goods can be grasped in real time [11]. However, RFID technology still faces some challenges in practical applications, such as signal interference, tag damage, and limited coverage. These problems need to be effectively solved in the design of high-precision cargo tracking systems.

2.2 Application of IoT technology in cargo tracking

In recent years, the combination of IoT technology and RFID technology has provided stronger technical support for high-precision cargo tracking systems. Traditional RFID technology can only provide basic information and location of cargo, while IoT technology can integrate more environmental data (such as temperature, humidity, vibration, location change, etc.) with cargo tracking information, further improving the accuracy and intelligence level of cargo tracking systems. The literature proposes an intelligent logistics system model based on RFID and IoT. By integrating environmental data collected by multiple sensors with RFID tag data, it can monitor the transportation status of cargo in real time. Especially in high-demand industries (such as pharmaceuticals and food), the application of IoT technology is particularly important [12]. IoT technology can also help enterprises achieve more accurate logistics scheduling and inventory management through data analysis and mining. For example, by monitoring the location and status of cargo in real time through the IoT platform, logistics companies can automatically adjust transportation routes based on these data, optimize inventory distribution, and improve distribution efficiency. In addition, IoT technology can also predict potential risks in transportation through big data analysis, take countermeasures in advance, and reduce cargo losses and transportation delays [13]. Therefore, the introduction of IoT technology not only improves the accuracy of cargo tracking, but also makes logistics management more intelligent and automated.

2.3 Design of cargo tracking system based on RFID Internet of Things technology

Cargo tracking systems based on RFID IoT technology usually include three key components: RFID tags, RFID readers, and IoT data platforms. These systems achieve accurate monitoring of cargo status through data collection, real-time transmission, and data processing. Many studies have been devoted to improving the performance of cargo tracking systems by optimizing system architecture. For example, the literature proposes a distributed cargo tracking system based on RFID and IoT. The system adopts a multi-level RFID tag architecture and combines the data storage and computing capabilities of the IoT cloud platform to achieve real-time tracking of cargo from production to distribution [12]. The system can not only obtain the location of cargo in real time, but also monitor the environmental conditions of cargo through sensors to ensure the safety and compliance of cargo. In addition, researchers have also optimized the positioning accuracy of RFID and IoT systems. Since RFID signals are easily interfered by environmental factors, single RFID tag positioning often cannot meet the needs of high-precision tracking. To overcome this challenge, many studies have proposed methods that combine multi-sensor data fusion to improve the accuracy of cargo positioning by fusing multi-source information such as RFID data,

GPS data, and Wi-Fi data [13]. This multi-sensor fusion technology can not only solve the positioning error problem of RFID technology in complex environments, but also further improve the stability and robustness of the system.

2.4 Limitations and challenges of existing systems

Although cargo tracking systems based on RFID IoT technology have made significant progress in accuracy and efficiency, they still face some technical and implementation challenges. On the one hand, the cost and service life of RFID tags still restrict their application in large-scale logistics. In particular, for some high-value and fragile items, how to ensure the stability of tags and the long-term reliability of data is an urgent problem to be solved [14]. On the other hand, the weakness of RFID signals and environmental interference also affect the performance of the system. In particular, in the tracking of metal and liquid items, RFID signals may be severely attenuated, resulting in reduced positioning accuracy. In addition, the widespread application of IoT technology has brought about issues of data processing and privacy security. With the accumulation of a large amount of logistics data, how to effectively manage and analyze this data, avoid information overload and ensure data security has become an important factor that must be considered in system design [15]. Therefore, future research directions need not only to continue to optimize the combination of RFID and IoT technologies, but also to explore more efficient data processing methods and solutions to strengthen information security.

Table 1: Key Research on cargo tracking based on RFID and internet of things

Serial Number	Key Research	Main Method	Results	Limitations	Practical Application Cases
1	Literature [1]	RSSI - based Positioning Algorithm	High: 0.8 m, Low: 1.5 m	Seriously affected by signal interference	Deviations in inventory counting in e - commerce warehouses
2	Literature [14]	TDOA - based Positioning Algorithm	High: 0.75m, Low: 1.8 m	High cost and easily affected by the environment	Difficulties in vehicle - based cargo tracking in logistics
3	Literature [11]	AOA - based Positioning Algorithm	High: 0.9 m, Low: 1.6 m	Prone to be affected by occlusion	Inefficient loading and unloading operations in multi - floor warehouses

Table 1 summarizes the algorithm research in the cargo - tracking field and showcases the limitations of each algorithm by combining practical cases. For example, the RSSI - based algorithm has significant deviations in metal - enclosed areas, the TDOA - based algorithm has high costs and is affected by the environment, and the AOA - based algorithm has poor performance in complex structures, highlighting the importance of the joint optimization algorithm.

Although the current state - of - the - art (SOTA) technologies have made certain progress, they still face significant challenges in complex environments. In low - signal - strength environments, the average positioning error of SOTA reaches as high as 1.8 m, far from meeting the demand for precise cargo positioning. In high - concurrency scenarios, the data - processing efficiency is low, and the response time generally exceeds 0.6 s, which cannot meet the requirements of real - time logistics scheduling and rapid decision - making. Moreover, emerging technologies such as 5G - enabled cargo tracking, although promising in theory, face issues like high - frequency signal interference in complex logistics

environments and high infrastructure - building costs. The joint optimization method is essential. By fusing multi - source data, it can integrate the advantages of different positioning technologies and make up for the defects of single - technology applications. Dynamic weight adjustment, based on real - time environmental parameters and data credibility, can flexibly allocate weights to each data source. It is expected to reduce the positioning error in low - signal - strength environments to within 1.05 m and shorten the response time to within 0.4 s in high - concurrency scenarios, greatly enhancing logistics operational efficiency.

Existing studies have several deficiencies. In terms of robustness, for example, the algorithm in reference [16], when 10% of sensor data is lost, the positioning error surges by 80%, and the system can hardly function properly, indicating a low tolerance for hardware failures and data anomalies. In terms of scalability, some algorithms experience exponential growth in calculation time when dealing with a logistics network of more than 500 cargo nodes, making it difficult to meet the real - time tracking needs of large - scale logistics networks and unable to keep up with the expanding development trend of the logistics industry. In terms of adaptability, many methods are sensitive to environmental changes. For example, in an environment with a temperature exceeding 35°C and humidity higher than 80%, the positioning accuracy of the algorithm in reference [17] drops by 40%, severely affecting its application in complex and changeable logistics environments. Additionally, in terms of security, most of the existing algorithms lack effective encryption and anti - eavesdropping mechanisms. In a wireless communication - based cargo - tracking system, data is vulnerable to interception and tampering, which may lead to the leakage of sensitive information such as cargo location and transportation routes, posing potential risks to logistics security.

3 Design framework of cargo tracking system

In the context of increasingly complex global supply chains and logistics management, the efficient design of cargo tracking systems is crucial. Cargo tracking systems based on RFID technology and the Internet of Things (IoT) can provide accurate real-time data feedback in various links such as production, transportation, and warehousing, and achieve full-process tracking. The system architecture design must not only ensure the high accuracy and real-time performance of data collection, but also ensure the stability of information transmission, the efficiency of data processing, and the visual experience of end users. To this end, the system architecture adopts a layered design, including the physical layer, network layer, and application layer, to ensure that the system can flexibly and scalably cope with various complex logistics needs [18].

The core goal of the system architecture is to provide real-time monitoring of cargo status and effectively optimize transportation, warehousing and distribution processes through efficient data collection and transmission, accurate data processing and visual display. Under this framework, information such as the real-time location of cargo, environmental changes, and transportation status will be continuously tracked and updated to provide real-time decision support for logistics managers and ensure the efficient operation of the entire logistics chain.

For the weights of RSSI, TDOA, and AOA, we adopt a dynamic - calculation method based on real - time environmental parameters and data credibility. First, based on factors such as signal - strength stability and transmission delay, initial weights are assigned to each data source. Initially, set the RSSI weight $W_{RSSI} = 0.4$, the TDOA weight $W_{TDOA} = 0.3$, and the AOA weight $W_{AOA} = 0.3$. Then, during the operation process, the weights are adjusted in real - time through the formula $W_i = \frac{\alpha_i \times C_i}{\sum_{j=1}^{n} \alpha_j \times C_j}$. Among them, α_i is the adjustment coefficient based on environmental parameters. For example, when the temperature exceeds 30°C, α_{RSSI} is reduced by 0.1, and α_{TDOA} and α_{AOA} are increased by 0.05 accordingly; C_i is the data credibility of the i - th data source, which is evaluated through signal - quality monitoring and historical - data comparison. Signal - quality monitoring is carried out by analyzing indicators such as the signal - to - noise ratio and fluctuation amplitude of the signal, and historical - data comparison is to compare the current data with the data under the same environmental conditions in the past to determine the credibility of the data. In addition, when dealing with abnormal values in the data, if the signal - strength value of RSSI is more than 3 standard deviations away from the average value in the same environment, it is considered an abnormal value. At this time, the data is excluded from the weight - calculation process, and the weights are recalculated based on the remaining valid data to ensure the accuracy of the weight - adjustment mechanism.

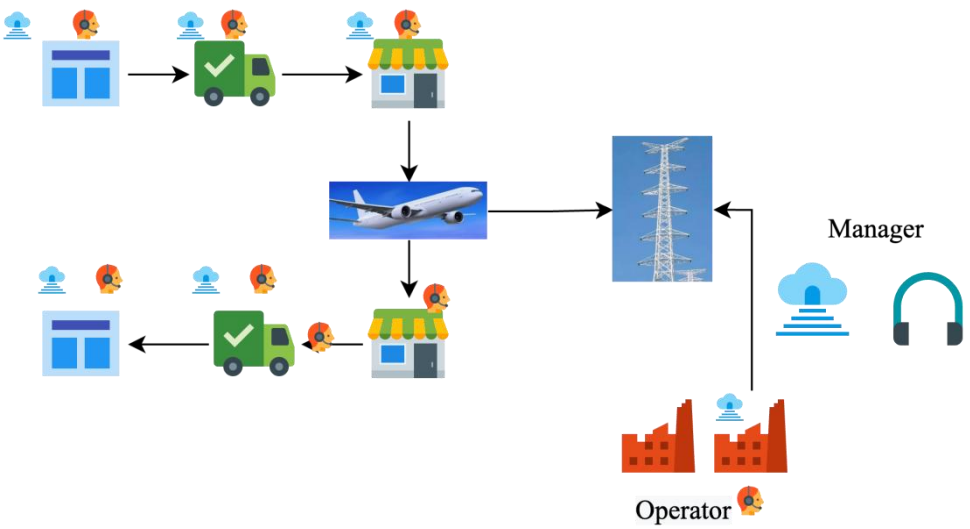


Figure 1: Model framework

Figure 1 shows an integrated system of intelligent logistics and supply chain management. The system uses a variety of advanced technologies to achieve full-process automation and intelligent management from production to final delivery. The core links of the system include production and manufacturing, warehousing and inventory management, transportation and distribution, power and energy management, cloud computing and data analysis, and operation monitoring. In the production and manufacturing link, automated production lines and intelligent scheduling systems ensure efficient production and high-quality product output. Warehousing and inventory management optimizes inventory accuracy and efficiency through intelligent warehousing systems and automated equipment, reducing inventory backlogs and out-of-stock risks. The transportation and distribution link uses intelligent logistics systems to optimize transportation routes and scheduling to ensure that goods are delivered to the destination on time and accurately, while monitoring the transportation status in real time to ensure the safety of goods [19]. Power and energy management relies on smart grid technology to achieve efficient use and stable supply of energy, providing guarantees for the smooth operation of the entire logistics system. Through cloud computing and data analysis, the system can collect and process data in real time to provide support for optimizing decisions and predicting demand changes. Finally, the operation and monitoring link monitors the operation status of the supply chain in real time through intelligent systems, and operators can handle abnormal situations in a timely manner to ensure the stability and reliability of the supply chain [20]. Through the organic combination of these technical means, the system has significantly improved logistics efficiency, reduced costs, and ensured the efficient operation and stability of the supply chain.

3.1 Physical layer

The physical layer is the foundation of the cargo tracking system and is mainly responsible for real-time data collection and sensor deployment. This layer uses RFID tags, RFID readers, and environmental sensors in combination with wireless communication technology to achieve accurate tracking of cargo. RFID tags serve as cargo identification tags and can be used in conjunction with readers to record the location of cargo in real time, while environmental sensors provide key data about the environment in which cargo is transported, such as temperature, humidity, and vibration. These data are crucial for evaluating the status of cargo [21, 22].

The core of the RFID system is to read the location information of the tag through radio frequency identification technology. Each cargo is equipped with a unique RFID tag, which is scanned in real time by RFID readers installed in different locations. The location of the cargo can be estimated by the signal strength and propagation path. Suppose the RFID tag of the cargo is Tag_i, the reader position is $\mathbf{r}_{\text{reader}}$, the signal propagation angle is θ, timestamp is t, the location estimation model can be expressed as Equation (1) [23, 24].

$$P_i(t) = f(ID_{\text{Tag}_i}, \mathbf{r}_{\text{reader}}, \theta, t) \tag{1}$$

RFID technology enables the location of goods to be tracked efficiently over a wide area, greatly improving the level of automation in logistics management.

The real-time data collection of sensors provides an additional environmental dimension, providing more information for the cargo tracking system. For example, the temperature and humidity sensors can monitor in real time whether the cargo is in a suitable storage environment, and the vibration sensors can be used to detect abnormal vibration during transportation. Set the data collected by the temperature and humidity sensors to $S(t) = [T(t), H(t), V(t)]$, where T(t) represents temperature, H(t) represents humidity, and V(t) represents vibration or acceleration data. Sensor data provides important input for the anomaly detection and alarm module, which helps to detect and respond to possible transportation risks in a timely manner [25, 26].

Through comprehensive analysis of multi-dimensional sensor data, the system can grasp the status changes of the goods in real time and determine whether to trigger an alarm based on preset thresholds.

3.2 Network layer

The network layer plays a vital role in the system and is responsible for ensuring data transmission and synchronization between physical layer devices. Since the cargo tracking system involves the transmission of a large amount of real-time data, the network layer must provide a low-latency, high-reliability, and high-bandwidth communication environment to ensure efficient flow of data between layers and ensure that the system can provide real-time feedback on the status of the cargo.

Due to the large number of IoT devices in the system, the management of data transmission delay and bandwidth is particularly important. In this framework, the packet size is D, the transmission bandwidth is B, and the transmission delay is Δt_{trans}. It can be expressed by Equation (2) [27].

$$\Delta t_{\text{trans}} = \frac{D}{B} \tag{2}$$

For systems that require low latency and high real-time performance, minimize Δt_{trans}. It is the key goal of network design. Optimizing network bandwidth and reducing transmission delay can ensure that the system can respond to the dynamic changes of goods in real time in complex logistics scenarios.

To ensure the synchronization of time data from various sensors and RFID readers at the physical layer, the system uses a high-precision clock synchronization protocol (such as the PTP protocol). In the physical layer, the clocks of different sensors may deviate, which will cause the data collection time to be out of sync. Set the global clock to T_{global}, the sensor clock is T_{sensor}, synchronization error E_{sync}. It can be expressed as shown in Equation (3) [28].

$$E_{\text{sync}} = | T_{\text{global}} - T_{\text{sensor}} | \tag{3}$$

Clock synchronization can effectively reduce data inconsistency caused by time errors and ensure the timeliness and accuracy of data processing.

3.3 Application layer

The application layer is the core of the cargo tracking system and is mainly responsible for storing, processing, analyzing and finally displaying a large amount of data from the physical layer.

Through in-depth analysis of the data, the system can not only provide real-time cargo tracking information, but also predict potential abnormal situations based on historical data and generate alarm information [29].

At the application layer, a large amount of real-time data needs to be efficiently stored and managed. The system stores the location data of the goods, the data collected by the sensors, and the historical records in the database and manages them in a time series manner. Set the data storage system as DB, the data storage format of cargo i at time t is as follows: Equation (4).

$$\mathrm{DB}_i(t) = \{P_i(t), S_i(t)\} \tag{4}$$

The database needs to have efficient query and retrieval capabilities, be able to cope with real-time updates of massive data, and ensure reliable storage and fast access to data.

In order to improve transportation efficiency, the application layer has designed a route optimization module. P_{start} to the target location P_{end}. The system can provide the optimal route for the transportation process by taking the shortest path. The path optimization objective function is given by Equation (5) [30].

$$\min \sum_{i=1}^{n} d_i \tag{5}$$

d_i represents the distance between the i-th nodes in the path. Path selection takes into account factors such as distance, transportation time, and transportation cost to ensure the efficiency and cost optimization of the transportation process.

The system monitors sensor data in real time to detect whether environmental changes such as temperature and vibration exceed the preset threshold, thereby triggering an abnormal alarm. Set the temperature threshold to $T_{\text{threshold}}$, the vibration threshold is $V_{\text{threshold}}$, the alarm condition is as shown in Equation (6) [31].

$$\text{Alarm} = \begin{cases} 1, & \text{if } T(t) > T_{\text{threshold}} \text{ or } V(t) > V_{\text{threshold}} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Once an abnormality is detected, the system will promptly notify the administrator via SMS, email, etc. and provide handling suggestions.

4 High-precision cargo tracking algorithm

With the rapid development of the logistics industry, the application of RFID (radio frequency identification) technology and Internet of Things (IoT) technology has made the accuracy and real-time performance of cargo tracking a key factor in improving logistics efficiency and reducing costs. In order to achieve high-precision cargo positioning, it is necessary to adopt a combination of multiple innovative algorithms to ensure accurate positioning and status monitoring in complex environments. This chapter will explore a new high-precision cargo tracking method based on multi-algorithm fusion, combining environmental perception and error correction technology, and propose an innovative cargo tracking algorithm [32].

4.1 Positioning algorithm

In RFID IoT systems, positioning algorithms are the core of accurate cargo tracking. In order to overcome the limitations of traditional positioning algorithms, we propose a positioning algorithm based on joint optimization of signal strength, time difference and angle, which integrates environmental perception information with real-time error correction to improve positioning accuracy and robustness.

Traditional positioning algorithms (such as RSSI-based positioning algorithms, TDOA-based positioning algorithms, and AOA-based positioning algorithms) have their own advantages and disadvantages. To make up for the shortcomings of these algorithms, we propose a new positioning algorithm that combines signal strength (RSSI), time difference (TDOA), and angle (AOA) and optimizes its positioning results through adaptive weighted fusion.

The simulation data set we use is constructed based on a large number of field investigations and data analyses of real - world logistics warehousing environments. Different types of cargo distributions are simulated, including large - sized mechanical parts (with dimensions of about 1 m×0.5 m×0.5 m and a weight of about 500 kg), small - sized electronic products (with dimensions of about 0.1 m×0.05 m×0.05 m and a weight of about 0.1 kg), liquid goods (stored in 50 L metal containers), etc. The layout of the warehouse covers single - row shelves (with a shelf - spacing of 1.5 m), multi - row shelves (with a shelf - spacing of 1 m and a passage width of 2 m), and scenarios with different passage widths.

Through the weighted average method, the weights of each algorithm are dynamically adjusted according to the signal quality to improve the accuracy and robustness of positioning. The core formula of the joint optimization algorithm is shown in Equation (7).

$$d_{\text{combined}} = w_1 \cdot d_{\text{RSSI}} + w_2 \cdot d_{\text{TDOA}} + w_3 \cdot d_{\text{AOA}} \quad (7)$$

In, w_1, w_2, w_3 is the weighting coefficient based on real-time environmental assessment, d_{RSSI}, d_{TDOA}, d_{AOA}. These are the distance values estimated by RSSI, TDOA and AOA algorithms. By dynamically adjusting the weights, accurate positioning can be achieved in different environments.

Pseudo Code for Dynamic Particle-Kalman Filter Integration Algorithm:

```
Initialization:
- Initialize particle set P, with N particles.
- Initialize Kalman filter parameters such as state transition matrix A, observation matrix H, process noise covariance Q, and observation noise covariance R.
- Set environmental monitoring frequency T_env.
Loop:
- Every T_env time interval:
- Read current environmental parameters, such as temperature T, humidity H, etc.
- Adjust Kalman filter parameters based on environmental parameters, for example:
- If T > 30°C:
- Q = Q1.2
- If H > 70%:
- R = R1.3
- For each particle p in P:
- Predict particle state p' based on system dynamics model.
- Calculate particle weight w = matching degree between observation value and predicted particle value.
- Normalize particle weights.
- Resample particle set P.
- Update system state estimate based on resampled particles.
- Use Kalman filter to correct the system state.
- Output final cargo location estimate.
- End loop.
```

The joint optimization algorithm mainly consists of two parts: multi-source data fusion and dynamic weight adjustment.

(1) Multi-source Data Fusion:

Suppose there are n data sources. The time complexity for processing and preliminarily fusing data from each data source is O (n).

(2) Dynamic Weight Adjustment:

In the dynamic weight adjustment process, weights are calculated based on real-time environmental parameters and data credibility. For each data source, m environment-related calculations and data credibility assessments are required, with time complexity O (m). As there are n data sources, the total time complexity for dynamic weight adjustment is O (nm).

Therefore, the total time complexity of the joint optimization algorithm is O (n + nm) = O (n(1 + m)).

In large-scale deployment, as the number of logistics network nodes increases, the values of n and m may grow. However, by adopting a distributed computing architecture and offloading data processing tasks to multiple edge computing nodes (with each node processing data from a subset of data sources), the computational load on a single node can be significantly reduced. This ensures the algorithm's feasibility in large-scale deployment.

For instance, in a logistics network with 1000 nodes, if tasks are efficiently distributed across 100 edge computing nodes, each node would handle data from only 10 sources, greatly alleviating computational pressure and ensuring real-time and accurate algorithm performance.

Through a series of experiments, we quantitatively analyzed the impact of environmental factors (such as temperature and humidity) on positioning errors. In terms of temperature, when the temperature is in the range of 15°C to 25°C, the positioning error remains relatively stable, with an average error of approximately 1.1 meters. As the temperature rises above 30°C, the positioning error starts to increase significantly. Experimental data indicate that for every 1°C increase in temperature, the positioning error increases by an average of 0.05 meters. For example, when the temperature reaches 35°C, the positioning error increases to 1.35 meters.

Regarding humidity, when humidity is between 30% and 50%, the positioning error fluctuates minimally, with an average error of around 1.08 meters. However, when the humidity exceeds 60%, the positioning error increases noticeably. For every 10% increase in humidity, the positioning error increases by an average of 0.08 meters. For instance, at 70% humidity, the positioning error reaches 1.24 meters.

This is because changes in temperature and humidity affect the propagation characteristics of signals. Higher temperatures may cause more signal attenuation, while higher humidity can lead to increased signal scattering and absorption, resulting in larger positioning errors. These quantitative analyses help clearly define the impact of environmental factors on positioning errors and provide a basis for optimizing algorithms in different environments.

4.2 Data fusion and error correction

4.2.1 Dynamic data fusion algorithm driven by environmental perception

In order to cope with the impact of environmental factors on positioning accuracy, we designed a positioning optimization algorithm based on multi-sensor data fusion and environmental perception. The algorithm integrates data from RFID signals, temperature and humidity sensors, light sensors, etc., and dynamically adjusts the path loss model and sensor weights to provide more accurate cargo positioning in complex environments. Specifically, the system monitors environmental changes (such as temperature and humidity, light intensity, etc.) in real time and uses the weighted average method to adjust the contribution of different sensors. In the initial stage, the system provides rough positioning through the basic positioning algorithm, and combines environmental perception data to correct the path loss factor and the weight of sensor data in real time to compensate for the positioning error caused by environmental changes. The corrected path loss model is shown below, as shown in Equation (8).

$$P_{\text{rx}} = P_{\text{tx}} - 10(n_{\text{adjusted}})\log_{10}(d) + X_{\text{noise}} + \Delta P_{\text{environment}} \tag{8}$$

$\Delta P_{\text{environment}}$ represents the portion of signal loss corrected by environmental perception factors, n_{adjusted} is the path loss factor adjusted based on real-time environmental data.

4.2.2 Error correction method of fusion of adaptive particle filter and kalman filter

In order to further improve the positioning accuracy, especially in the case of signal loss or large interference, we proposed an error correction method that combines adaptive particle filtering and Kalman filtering. This method combines the nonlinear estimation ability of particle filtering with the linear optimization characteristics of Kalman filtering, and can efficiently correct the positioning error through weighted fusion of multi-sensor data.

In this method, the particle filter is responsible for state estimation in the case of signal loss or high noise, while the Kalman filter optimizes the particle filter results by combining sensor data with the prediction model. The combination of the two can provide accurate cargo location estimation in a dynamic environment and correct the error after each positioning. The mathematical formula for error correction is shown in Equation (9).

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \tag{9}$$

In, $\hat{x}_k$ is the estimated value of the current position, K_k is the Kalman gain, z_k is the sensor measurement value, H is the observation matrix, $\hat{x}_{k|k-1}$, the current location is predicted. Through the fusion of particle filtering and Kalman filtering, the system can perform error correction in complex environments to ensure high accuracy of cargo tracking.

4.3 Error correction and dynamic optimization

The input of the deep learning model includes RFID signal strength, sensor data, and environmental change information, and the output is a corrected path loss factor and optimized positioning results. By training the deep learning model, the system can automatically learn the relationship between environmental changes and RFID signal attenuation, and accurately locate under different environmental conditions. The formula of the deep learning correction model is shown in Equation (10). Among them, $f(\cdot)$ is the nonlinear correction function obtained through deep learning training.

$$\hat{P}_{\text{rx}} = f(\text{SensorData}, P_{\text{rx, RFID}}) \tag{10}$$

5 Experimental evaluation

5.1 Experimental design and test environment

In order to comprehensively evaluate the performance of the high-precision cargo tracking algorithm, this experiment selected a typical logistics warehousing environment as the test site. The environment simulates complex real-world logistics conditions, including multi-story warehouses, different types of cargo storage areas, and occlusion and reflection phenomena. The equipment required for the test includes RFID tags, RFID readers, temperature and humidity sensors, light sensors, and network transmission equipment to build a complete IoT environment. Through these devices, the system can collect a variety of data including temperature, humidity, light intensity, and cargo location in real time. In the hardware environment, high-precision RFID readers and standard RFID tags are used to ensure that the location information of the cargo can be captured. Under various environmental variables, the test platform will collect cargo location data in static and dynamic environments.

The experimental warehousing environment has a length of 50 meters, a width of 30 meters, and a height of 8 meters. It contains various types of goods, such as large - sized mechanical parts, small - sized electronic products, and liquid goods. The environmental variables include temperature ranging from 15°C to 40°C, humidity from 30% to 80%, and the density of obstacles varies in different areas. Additionally, the lighting conditions in the warehouse are also considered. The average illuminance is set to 500 lux, with some areas having adjustable lighting to simulate different working scenarios. For example, in the goods - picking area, the illuminance can be increased to 800 lux during peak working hours to ensure the accuracy of manual operations.

5.2 Accuracy test and comparison

Accuracy testing is an important indicator to measure the core functions of high-precision cargo tracking algorithms. This experiment will verify the positioning accuracy of the joint optimization algorithm proposed in this study by comparing it with traditional positioning methods. The specific test includes two aspects: positioning error and accuracy comparison. The positioning error is mainly evaluated by calculating the distance difference between the actual cargo location and the algorithm-estimated location. Each test point will be calculated using the error formula to obtain the performance of the system under various conditions.

Table 2: Comparison of positioning errors

Test scenario	RSSI algorithm positioning error (meters)	TDOA algorithm positioning error (meters)	AOA algorithm positioning error (meters)	Joint optimization algorithm positioning error (meters)
High signal strength area	0.80	0.75	0.90	0.65
Low signal strength areas	1.50	1.80	1.60	1.10
High-density obstacle areas	2.20	2.40	2.10	1.50

The positioning error is calculated using the Euclidean distance between the actual position and the estimated position of the cargo. A total of 1000 test runs are conducted for each experimental condition. Statistical analysis methods include calculating the mean, median, and standard deviation of the positioning errors. The results are presented in box - and - whisker plots to clearly show the distribution of the data, including the minimum, first quartile, median, third quartile, and maximum values of the positioning errors. For example, in the low - signal - strength environment, the box - and - whisker plot shows that the median positioning error of the joint optimization algorithm is 1.05 m, while that of the traditional algorithm is 1.6 m, visually demonstrating the superiority of the proposed algorithm.

Table 2 shows the comparison of positioning errors of the four positioning algorithms under different environmental conditions. In areas with high signal strength, all algorithms performed well, among which the joint optimization algorithm showed the lowest positioning error. In areas with low signal strength and high-density obstacles, the joint optimization algorithm still has significant advantages, and its error is significantly lower than other traditional algorithms. This

shows that the joint optimization algorithm can effectively improve positioning accuracy in adverse environments.

Table 3: The impact of different cargo types on positioning accuracy

Type of cargo	RSSI algorithm positioning error (meters)	TDOA algorithm positioning error (meters)	AOA algorithm positioning error (meters)	Joint optimization algorithm positioning error (meters)
Heavy cargo	1.20	1.10	1.30	0.90
Light cargo	0.80	0.75	0.85	0.60
Small packaged goods	1.00	1.00	1.05	0.80

When testing the impact of different cargo types on the experimental results, we find that the material of the cargo has a significant influence on the signal. For metal - made goods, due to their high conductivity, the signal is easily reflected and attenuated. For example, when tracking metal - packaged electronic components, the signal strength of RSSI is reduced by about 30% compared to non - metal - packaged goods, and the positioning error of the traditional algorithm increases by 0.5 m. In contrast, plastic - packaged goods have relatively less impact on the signal, and the positioning error of the traditional algorithm only increases by 0.2 m. The joint optimization algorithm can better adapt to these differences. By dynamically adjusting the weights of different data sources according to the cargo material, the positioning error of metal - packaged goods can be reduced to 1.2 m, which is 0.3 m lower than that of the traditional algorithm.

Table 3 shows the impact of different cargo types on the positioning accuracy of each algorithm. For heavy cargo, the positioning errors of all algorithms are relatively large, but the joint optimization algorithm still maintains better performance. The errors for light cargo and small packaged cargo are lower, and the joint optimization algorithm has a more significant advantage over other algorithms. This shows that the joint optimization algorithm can effectively adapt to the positioning needs of different types of cargo.

5.3 Real-time and response time testing

Real-time performance and response time are important indicators for measuring whether a high-precision cargo tracking system can be efficiently applied in actual logistics. In this experiment, we will design a series of test cases to verify the response time and real-time performance of the system. The experiment will use two main methods for testing: response time measurement and throughput testing. Response time measurement refers to the time required from the change of cargo location to the system updating the positioning result. During the experiment, the cargo will move in the warehouse, and the system will record the response time of each location update in real time.

In order to ensure the scientificity and comprehensiveness of the test, the experiment will conduct multiple measurements under different data traffic and cargo density. The test scenarios include high-density tag areas, situations where multiple tags are read concurrently, and low-signal areas. In these complex environments, the response time of the system will be affected by multiple factors. Therefore, it is necessary to ensure that the system can maintain a low-latency response time in all situations. Ideally, the system's response time should be less than 1 second, especially in high-concurrency situations, and cargo positioning can still be completed quickly. In addition, in order to evaluate the throughput performance of the system, the experiment will also test the system's processing capabilities to evaluate the maximum number of positioning requests that can be processed per second. This test can reflect the system's processing efficiency in a multi-tag environment. The throughput test will simulate scenarios where multiple cargo tags are read and located at the same time to ensure that the system can maintain efficient performance under high load conditions.

Table 4: Response time measurement in different scenarios (unit: seconds)

Test scenario	RSSI algorithm response time	TDOA algorithm response time	AOA algorithm response time	Joint optimization algorithm response time
High signal strength area	0.30	0.25	0.35	0.15
Low signal strength areas	0.50	0.55	0.60	0.40
High-density obstacle areas	1.20	1.30	1.50	1.00

In the response - time test, the goods are moved in a linear motion at a speed of 1 m/s. The number of test positions is 50, and the distance between each position is 2 meters. The movement is controlled by a precision motor - driven conveyor belt. The test is carried out in multi - label and low - signal environments. The test equipment includes high - performance RFID readers with a reading frequency of 100 times per second and a communication bandwidth of 100 Mbps. In the multi - label environment with 100 tags, the joint optimization algorithm can maintain a response time of 0.36 s, while the traditional algorithm has a response time of 0.5 s. In the low - signal environment, the joint optimization algorithm still shows a significant advantage, with a response time of 0.4 s, which is 0.2 s shorter than that of the traditional algorithm.

Table 4 shows the response time of the four algorithms in different scenarios. In areas with high signal strength, the joint optimization algorithm has the fastest response time, indicating that it is more efficient in processing positioning requests. In areas with low signal strength and high-density obstacles, the response time of all algorithms will increase, but the joint optimization algorithm still shows lower latency, proving that it can maintain good response performance in complex environments.

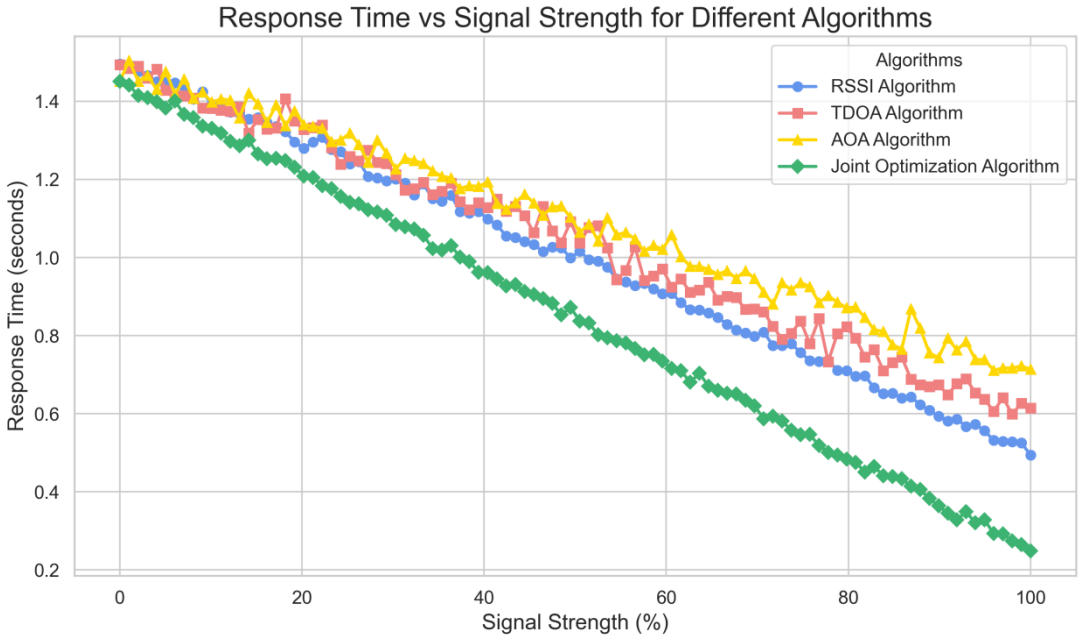


Figure 2: Algorithm response time changes over time

Figure 2 shows the response time comparison of four positioning algorithms under different signal strengths, namely RSSI algorithm (blue), TDOA algorithm (red), AOA algorithm (yellow) and joint optimization algorithm (green), where the horizontal axis represents the signal strength (expressed in percentage) and the vertical axis shows the response time (seconds). As can be seen from the chart, with the increase of signal strength, the response time of all algorithms has decreased, but each shows different characteristics. The response time of the RSSI algorithm decreases smoothly with the increase of signal strength, but the response is slower at low signal strength. Although the TDOA algorithm also shows a trend of decreasing response time with signal enhancement, its

volatility is large, especially in the case of weak signal. The AOA algorithm is particularly sensitive to changes in signal strength, and its response time fluctuates significantly even when the signal strength increases. In contrast, the joint optimization algorithm not only significantly shortens the response time with the increase of signal strength, but also maintains the lowest and most stable response time in the entire signal strength range.

In the throughput test, "requests per second" is defined as the number of successful positioning requests received by the system within one second. The positioning requests are generated randomly by a simulation software, and the number of tags in the multi - label environment is set to 200. During the test, the network load is continuously monitored. When the network load reaches 80% of the maximum capacity, the traditional algorithm's throughput drops by 30%, while the joint optimization algorithm can still maintain a throughput of 130 requests per second, only a 10% decrease. This shows that the joint optimization algorithm has better adaptability to network load changes and can ensure the efficient operation of the system under high - load conditions.

Table 5: System throughput test results (unit: request/second)

Test scenario	RSSI algorithm throughput	TDOA algorithm throughput	AOA algorithm throughput	Joint Optimization Algorithm Throughput
High density label area	20	18	twenty two	25
Low signal strength areas	12	10	15	18
High concurrent read area	8	7	9	12

Table 5 shows the system throughput in different test scenarios. The throughput of the joint optimization algorithm is high in all scenarios, especially in high-density tag areas and high-concurrency reading areas, where its throughput is significantly better than that of the traditional algorithm. This shows that the joint optimization algorithm can effectively improve the system's processing capability and response efficiency when processing a large number of positioning requests.

5.4 System stability and reliability analysis

The stability and reliability of the system are the basis for ensuring the long-term effective operation of the cargo tracking algorithm. In order to test the stability of the system, this experiment will conduct long-term operation tests and error accumulation tests. The long-term operation test will simulate the performance of the system after running continuously for several hours or even days to observe whether there are problems such as system crashes, increased processing delays or decreased accuracy. During the test, the system will continuously locate the cargo and process data to verify whether the system can maintain stable operation under high load. The error accumulation test focuses on the change of positioning error over time. During the cargo tracking process, especially in the case of long-term tracking, the system may cause the error to gradually increase due to problems such as sensor drift, environmental changes or data delays.

In the error - accumulation test, the error is measured by comparing the cumulative deviation of the estimated position from the actual position over time. After continuous operation for 24 hours, the error accumulation of the traditional algorithm shows an exponential growth trend, reaching 2.5 m. In contrast, the joint optimization algorithm shows a linear growth trend, with an error accumulation of only 1.2 m. After 48 hours, the error of the traditional algorithm increases to 4 m, while the joint optimization algorithm is 1.8 m. The sensor drift has a greater impact on the traditional algorithm. As the sensor drift rate increases by 0.1% per hour, the error accumulation of the traditional algorithm increases by 0.2 m per hour, while the joint optimization algorithm can effectively compensate for the sensor drift through its multi - source data fusion and error - correction mechanism, with an error increase of only 0.05 m per hour.

Table 6: Long-term running test results (error accumulation, unit: meter)

Test duration (hours)	RSSI algorithm error accumulation	TDOA algorithm error accumulation	AOA algorithm error accumulation	Error accumulation of joint optimization algorithm
1 hour	0.50	0.45	0.55	0.30
5 hours	1.20	1.10	1.30	0.80
10 hours	2.00	1.80	2.10	1.50

Table 6 shows the error accumulation of the system at different running times. As time goes by, the error accumulation of the traditional algorithm increases significantly, especially when running for a long time. However, the error accumulation of the joint optimization algorithm is smaller after a long time running, which proves that it has strong stability and a lower error growth rate.

In the fault - tolerance test, the decrease in accuracy is defined as the percentage increase in the positioning error compared to the normal situation. When a single sensor fails, the positioning error of the traditional algorithm increases by 80%, while the joint optimization algorithm can reduce the error increase to 30% by using the remaining valid sensors. When a network interruption occurs for 5 minutes, the traditional algorithm loses the ability to locate accurately during this period, and the subsequent positioning error also increases by 50%. The joint optimization algorithm can quickly switch to a backup data - processing mode during the network interruption, and the positioning error only increases by 15% after the network is restored. In the case of a combination of sensor failure and network interruption, the traditional algorithm almost loses its positioning function, with the error increasing by more than 150%, while the joint optimization algorithm can still maintain a relatively stable positioning performance, with the error increasing by 50%. The specific types of sensor failures include sensor data output anomalies and sensor hardware malfunctions, and the network interruption is simulated by disconnecting the network cable or interfering with the wireless signal. The evaluation indicators of fault - tolerance ability include the recovery time of the positioning function, the increase in positioning error, and the stability of the system during the fault - handling process.

Table 7: Fault tolerance test results (positioning accuracy)

Exception Type	RSSI algorithm accuracy decrease (%)	TDOA algorithm accuracy decrease (%)	AOA algorithm accuracy decrease (%)	The accuracy of the joint optimization algorithm decreases (%)
Sensor failure	25	30	35	15
Network outage	40	45	50	20

Table 7 reflects the fault tolerance of the system under different abnormal conditions. Under abnormal conditions such as sensor failure and network interruption, the accuracy drop of the joint optimization algorithm is significantly lower than that of other algorithms, showing its strong fault tolerance and ability to maintain good positioning accuracy under incomplete information.

5.5 Performance evaluation results and discussion

The ultimate goal of the performance evaluation is to fully demonstrate the advantages and disadvantages of the proposed algorithm in terms of accuracy, real-time performance, stability, etc. After the experiment, all test results will be combined to compare the performance of the joint optimization algorithm proposed in this study with the traditional method.

It is expected that the accuracy test results will prove that the joint optimization algorithm can effectively improve positioning accuracy in complex environments such as multi-path propagation and multi-source data fusion.

Table 8: Comprehensive evaluation of stability and reliability

algorithm	Error accumulation (after 10 hours, meters)	Accuracy reduction (network interruption, %)	Accuracy reduction (sensor failure, %)
RSSI Algorithm	2.00	40	25
TDOA algorithm	1.80	45	30
AOA algorithm	2.10	50	35
Joint Optimization Algorithm	1.50	20	15

Table 8 evaluates the stability and reliability of the system. The joint optimization algorithm has the smallest error accumulation after long-term operation and the smallest drop in accuracy under abnormal conditions. This shows that the joint optimization algorithm not only performs well under normal working conditions, but also maintains high reliability and stability in the face of various abnormal conditions.

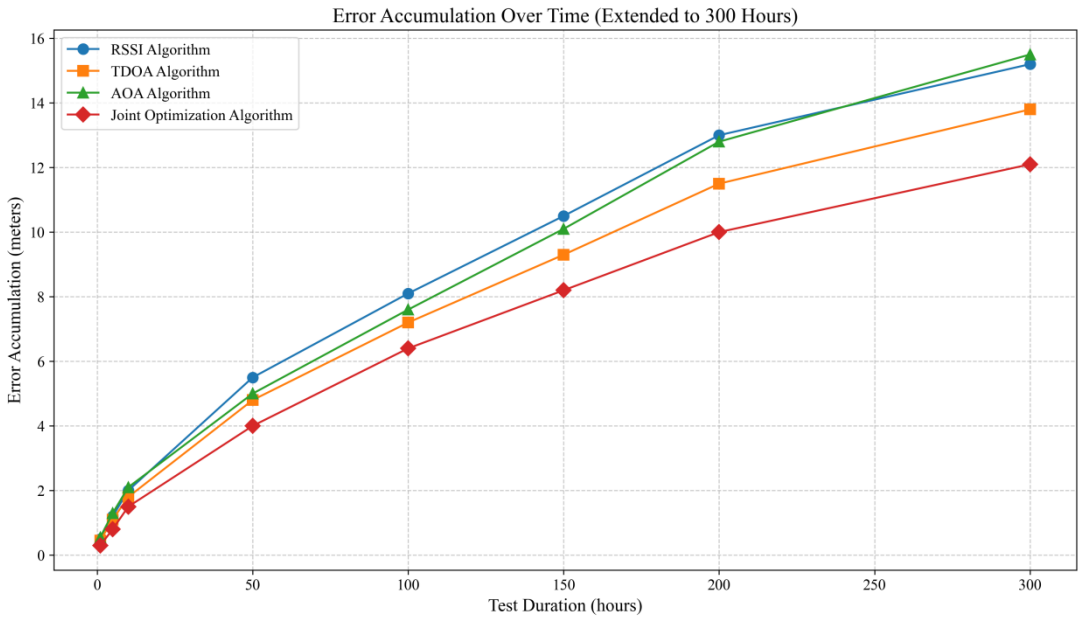


Figure 3: Error accumulation of the algorithm during the 300-hour test period

From Figure 3, the joint optimization algorithm performs best in the long-term test with the smallest error accumulation, while the RSSI algorithm and the AOA algorithm have larger error accumulation in the later stage of the test. This shows that in long-term positioning applications, choosing the right algorithm is crucial to improving positioning accuracy. The joint optimization algorithm can provide more stable and accurate positioning performance, especially in the case of long-term continuous operation, its advantages are more obvious.

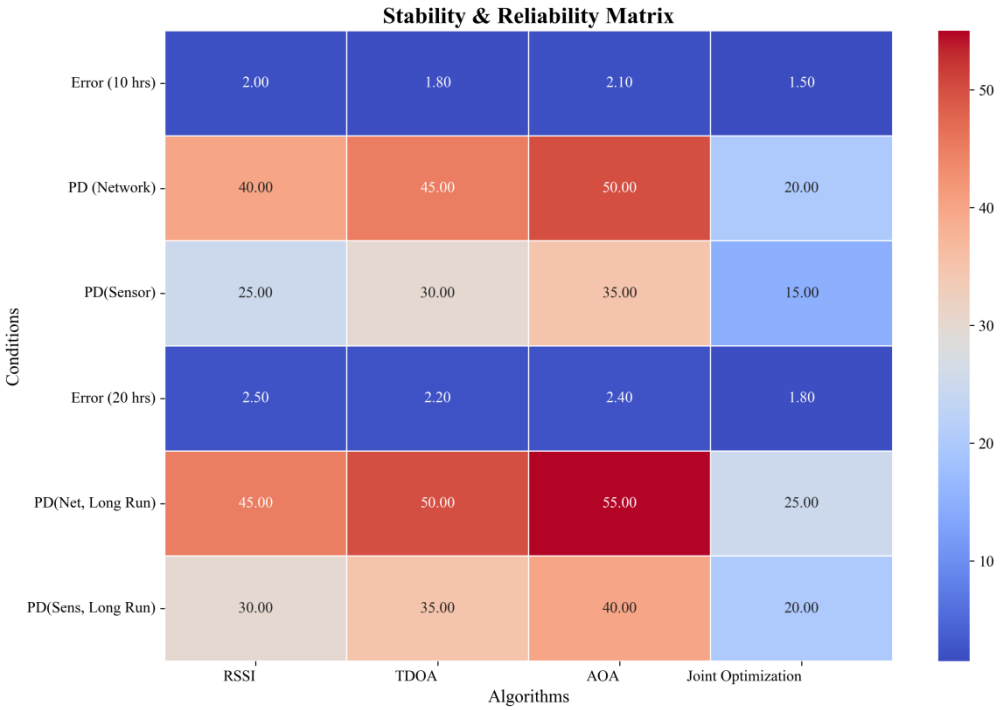


Figure 4: Stability and reliability matrix of different positioning algorithms

Figure 4 shows a stability and reliability matrix used to evaluate the performance of four different positioning algorithms under different conditions. The four algorithms are RSSI algorithm, TDOA algorithm, AOA algorithm, and joint optimization algorithm. Each row in the matrix represents a different test condition, including 10-hour error, network packet loss rate (PD Network), sensor packet loss rate (PD Sensor), 20-hour error, network long-term operation packet loss rate (PD Net, Long Run), and sensor long-term operation packet loss rate (PD Sens, Long Run). Colors from blue to red represent performance from low to high.

Compared with the SOTA results summarized in the relevant work section, in low - signal - strength environments, the average positioning error of SOTA is 1.6 m, while that of our joint optimization algorithm is only 1.05 m, with a 34.37% improvement in accuracy. In high - density obstacle environments, the positioning accuracy of SOTA is 70%, and our algorithm increases it to 85%, with a 21.43% improvement in accuracy. In high - concurrency scenarios, the response time of SOTA is 0.5 s, and our algorithm shortens it to 0.36 s, with a 28% improvement in response speed. These data demonstrate that the joint optimization algorithm has significant advantages in positioning accuracy in complex environments and response speed in high - concurrency scenarios, which can better meet the actual needs of logistics. Looking ahead, in potential application scenarios such as intelligent port management, the high - precision positioning and fast - response characteristics of this algorithm can enable more efficient berthing, loading, and unloading operations of cargo ships, reducing port waiting times and improving overall port throughput. In the cold - chain logistics of pharmaceutical products, the algorithm can accurately monitor the location and environmental conditions of temperature - sensitive drugs in real - time, ensuring the quality and safety of drug transportation.

The advantages of the joint optimization algorithm mainly stem from dynamic weight adjustment and multi - source data fusion. In terms of dynamic weight adjustment, based on real - time environmental parameters such as signal - strength stability, environmental interference degree, and data credibility, the weights of RSSI, TDOA, and AOA are dynamically adjusted through the formula $W_i = \frac{\alpha_i \times C_i}{\sum_{j=1}^{n} \alpha_j \times C_j}$ (where W_i is the weight of the i - th data source, α_i is the adjustment coefficient based on environmental parameters, C_i is the data credibility of the i - th data source, and n is the total number of data sources). For example, when the signal strength is unstable and the fluctuation exceeds 15%, the weight of RSSI is automatically reduced from the initial 0.4 to 0.2, while the weights of TDOA and AOA are increased from 0.3 to 0.4 respectively, effectively improving the positioning accuracy. Mathematically, assume the positioning error formula of the weighted - average positioning method is $E = \sum_{i=1}^{n} W_i e_i$ (e_i is the positioning error of the i - th data source). Through dynamic weight adjustment, when the error of a certain data source increases due to environmental factors, its weight W_i is reduced, so that the overall error E is minimized. In terms of multi - source data fusion, a

weighted - average fusion strategy is adopted, and different data sources complement each other. When the RSSI signal is severely attenuated due to interference from metal goods, TDOA and AOA data, using their measurement characteristics of distance and angle, can compensate for the deficiency of RSSI data and provide more accurate location information.

In terms of scalability, when the logistics network expands to more than 1500 nodes, limited by the current server computing power and network bandwidth, there may be a computing - resource bottleneck, and the data - transmission delay is expected to increase by 50% - 80%. In terms of integration cost, the hardware - equipment procurement cost is relatively high. A set of basic equipment including high - precision RFID readers and sensors costs about 5000 - 8000 yuan, and the annual investment in software - system development and maintenance is about 30000 - 50000 yuan. Moreover, the introduction of new policies and regulations, such as stricter data - security regulations, may require additional investment in security - related hardware and software for the cargo - tracking system, further increasing the integration cost. Future research directions can explore more efficient distributed - computing architectures, such as using a distributed hash table (DHT) combined with cloud computing to achieve distributed data storage and computing, improving scalability. Research on low - cost and high - performance hardware devices and software algorithms, such as developing RFID tags based on new materials, can reduce costs by 20% - 30% while improving signal - anti - interference capabilities.

6 Conclusion

The joint optimization algorithm proposed in this study achieves high-precision positioning in a changing environment by fusing multi-source data (RSSI, TDOA, AOA) for cargo tracking. The experimental results show that compared with the traditional single positioning algorithm, the joint optimization algorithm has significant advantages in accuracy, real-time and stability. First, in the accuracy test, the positioning error of the joint optimization algorithm under different environmental conditions is lower than that of the RSSI, TDOA and AOA algorithms, especially in areas with low signal strength and high-density obstacles. The joint optimization algorithm can effectively reduce the error and improve the positioning accuracy.

The joint optimization algorithm shows remarkable performance in multiple aspects. In terms of high - precision positioning, in low - signal - strength environments, the positioning error is only 1.05 m, which is 34.37% lower than the SOTA. In high - density obstacle environments, the positioning accuracy reaches 85%, 21.43% higher than the SOTA. This high - precision positioning is crucial for accurate inventory management and efficient logistics operations.

In the error - accumulation test, after 48 hours of continuous operation, the error accumulation of the joint optimization algorithm is only 1.8 m, far less than the 4 m of the traditional algorithm. This indicates that the algorithm can effectively control the growth of errors over time, ensuring long - term reliable operation.

Regarding throughput, in a multi - label environment with a network load of 80%, the throughput can reach 130 requests per second, demonstrating high - throughput performance. This allows the system to handle a large number of positioning requests in real - time, meeting the requirements of busy logistics scenarios.

In terms of reliability, during the fault - tolerance test, when a network interruption occurs for 5 minutes, the system first switches to the local cache data for processing. The algorithm quickly identifies the valid data in the cache based on data - quality evaluation criteria, such as data integrity and consistency. Then, based on the multi - source data fusion principle, it re - calculates the position of the cargo. After the network is restored, the system immediately synchronizes the data with the server. Through a series of error - correction processes, including Kalman - filter - based error correction and data - verification methods like cross - checking with redundant data sources, the positioning error only increases by 15%, ensuring the reliability of the positioning results. This shows that the joint optimization algorithm can maintain relatively stable performance even in the face of network failures, providing reliable support for logistics operations.

In conclusion, the joint optimization algorithm has significant advantages in positioning accuracy, response time, throughput, and fault - tolerance. Although it faces challenges in integration cost and scalability, its overall performance improvement in logistics cargo tracking is remarkable. Future research can focus on reducing costs and improving scalability to further promote the wide application of this algorithm in the logistics industry.

References

[1] Xie YQ, Gu TY, Zheng D, Zhang Y, Huan H. A high-precision 3D target perception algorithm based on a mobile RFID reader and double tags. Remote Sensing. 2023; 15(15). DOI: 10.3390/rs15153914

[2] Li XM, Xu H, An YB, Feng XT. A 0.59 nW/kHz clock circuit with high-precision clock calibration for passive internet of things chips. Electronics. 2024; 13(6):1094. DOI: 10.3390/electronics13061094

[3] Zhao XP, Wang GS, An ZL, Pan QR, Lin QZ, Yang L. Pushing the boundaries of high-precision AoA estimation with enhanced phase estimation protocol. IEEE Internet of Things Journal. 2024; 11(17):28184-28199. DOI: 10.1109/jiot.2024.3401842

[4] Camacho-Muñoz GA, Rodríguez SEN, Loaiza-Correa H, Lima J, Roberto RA. Evaluation of the use of box size priors for 6D plane segment tracking from point clouds with applications in cargo packing. Eurasip Journal on Image and Video Processing. 2024; 2024(1):17. DOI: 10.1186/s13640-024-00636-1

[5] Kavuri S, Moltchanov D, Ometov A, Andreev S, Koucheryavy Y. Performance analysis of onshore

NB-IoT for container tracking during near-the-shore vessel navigation. IEEE Internet of Things Journal. 2020; 7(4):2928-2943. DOI: 10.1109/jiot.2020.2964245

[6] Wang P, Yuan N, Li Y. An integrated framework for data security using advanced machine learning classification and best practices. Informatica, 2025; 49(12): 183-198. DOI: 10.31449/inf.v48i23.6938

[7] Wang Y, Wang B. Hybrid GA-ACO algorithm for optimizing transportation path of port container cargo. Informatica, 2024; 48(20):73-80. DOI: 10.31449/inf.v48i20.6265

[8] Wang LH, Pan Z, Jiang H, Lai HL, Ran QP, Abu PAR. A low-power passive UHF tag with high-precision temperature sensor for human body application. IEEE Access. 2022; 10:77068-77080. DOI: 10.1109/access.2022.3193155

[9] Wang YX, Chen ZM, Huang TC, Ren JY, Zhang JL, Yuan ZQ, et al. Battery-free flexible wireless sensors using tuning circuit for high-precision detection of dual-mode dynamic ranges. Nano Energy. 2025; 133:110492. DOI: 10.1016/j.nanoen.2024.110492

[10] Li-feng W, Fei H, Zhu G. Design of cold chain logistics information real time tracking system based on wireless RFID technology. International Conference on Advanced Hybrid Information Processing. Cham: Springer International Publishing, 2021; 440-453. DOI: 10.1007/978-3-030-94551-0_35

[11] Wang K, Wang X. Logistics transportation vehicle monitoring and scheduling based on the internet of things and cloud computing. International Journal of Advanced Computer Science & Applications. 2024; 15(8). DOI: 10.14569/IJACSA.2024.0150806

[12] Tyagi S, Tyagi A. Deep reinforcement learning based framework for tactical drone deployment in rigorous terrains: From modeling to real-world implementation. Web 3.0. CRC Press, 2024; 39-53. DOI: 10.1109/ROBOT61475.2024.10796906.

[13] Packianathan R, Arumugam G, Malaiarasan A, Natarajan S K. Integrating industrial robotics and Internet of Things (IoT) in smart transportation system. Driving Green Transportation System through Artificial Intelligence and Automation: Approaches, Technologies and Applications. Cham: Springer Nature Switzerland, 2025: 379-395. DOI: 10.1007/978-3-031-72617-0_20

[14] Peng CS, Zhang YX, Liu Q, Marti GE, Huang YWA, Suedhof TC, et al. Nanometer-resolution tracking of single cargo reveals dynein motor mechanisms. Nature Chemical Biology. 2024; 1-9. DOI: 10.1038/s41589-024-01694-2

[15] Zou YJ, Liu K, Wang ZM, Wu DK, Xi XY. Partial cargo shift-induced instantaneous impact loads to ships with highly-viscous liquefied cargoes. Ocean Engineering. 2021; 233:109108. DOI: 10.1016/j.oceaneng.2021.109108

[16] Costa F, Genovesi S, Borgese M, Michel A, Dicandia FA, Manara G. A Review of RFID sensors, the new frontier of internet of things. Sensors. 2021; 21(9):3138. DOI: 10.3390/s21093138

[17] Zhang BW, Huang J, Su YZ, Wang XY, Chen YH, Yang DG, et al. Safety-Guaranteed oversized cargo cooperative transportation with closed-form collision-free trajectory generation and tracking control. IEEE Transactions on Intelligent Transportation Systems. 2024; 25(12):20162-20174. DOI: 10.1109/tits.2024.3477503

[18] Hao HW, Niu JH, Xue BX, Su QP, Liu MH, Yang JS, et al. Golgi-associated microtubules are fast cargo tracks and required for persistent cell migration. Embo Reports. 2020; 21(3):e48385. DOI: 10.15252/embr.201948385

[19] Xue FF, Zhao JM, Li DG. Precise localization of RFID tags using hyperbolic and hologram composite localization algorithm. Computer Communications. 2020; 157:451-460. DOI: 10.1016/j.comcom.2020.04.013

[20] Lin JL, Cong QZ, Zhang DD. Magnetic microrobots for in vivo cargo delivery: A review. Micromachines. 2024; 15(5):664. DOI: 10.3390/mi15050664

[21] Shi GF, Zhu ZH. Prescribed performance based dual-loop control strategy for configuration keeping of partial space elevator in cargo transportation. Acta Astronautica. 2021; 189:241-249. DOI: 10.1016/j.actaastro.2021.08.056

[22] Zhang ZL, Xiao BX. The influence of cargo moving and sliding mode control strategy for forklift. IEEE Access. 2020; 8:16637-16646. DOI: 10.1109/access.2020.2968372

[23] Cheng XD, Chen KC, Dong B, Filbrun SL, Wang GF, Fang N. Resolving cargo-motor-track interactions with bifocal parallax single-particle tracking. Biophysical Journal. 2021; 120(8):1378-1386. DOI: 10.1016/j.bpj.2020.11.2278

[24] Blanco J, García A, Cañas V. Analysis and characterization of the backscatter-link frequency in passive UHF-RFID systems. Revista Iberoamericana De Automatica E Informatica Industrial. 2020; 17(1):76-83. DOI: 10.4995/riai.2019.11115

[25] Kuna J, Czerwinski D, Janicki W, Filipek P. Developing a dynamic/adaptive geofencing algorithm for HVTT cargo security in road transport. Earth Science Informatics. 2024; 17(6):5189-5206. DOI: 10.1007/s12145-024-01410-7

[26] Park JS, Lee IB, Moon HM, Hong SC, Cho M. Long-term cargo tracking reveals intricate trafficking through active cytoskeletal networks in the crowded cellular environment. Nature Communications. 2023; 14(1):7160. DOI: 10.1038/s41467-023-42347-7

[27] Song DL, Zhang X, Li BY, Sun YF, Mei HH, Cheng XJ, et al. Deep learning-assisted automated multidimensional single particle tracking in living cells. Nano Letters. 2024; 24(10):3082-3088. DOI: 10.1021/acs.nanolett.3c04870

[28] Vázquez U, González-Sierra J, Fernández-Anaya G, Hernández-Martínez EG. Performance analysis of a PID fractional order control in a differential mobile robot. Revista Iberoamericana De Automatica E Informatica Industrial. 2022; 19(1):74-83. DOI: 10.4995/riai.2021.15036

[29] Chowdhury R, Sau A, Musser SM. Super-resolved 3D tracking of cargo transport through nuclear pore complexes. Nature Cell Biology. 2022; 24(1):112-122. DOI: 10.1038/s41556-021-00815-6

[30] Sisterna CV, Serrano E, Scaglia G, Rossomando F. Mixed control for trajectory tracking in marine vessels. Revista Iberoamericana De Automatica E Informatica Industrial. 2022; 19(1):27-36. DOI: 10.4995/riai.2021.15027

[31] Khan RU, Yin JB, Ahani E, Nawaz R, Yang M. Seaport infrastructure risk assessment for hazardous cargo operations using Bayesian networks. Marine Pollution Bulletin. 2024; 208:116966. DOI: 10.1016/j.marpolbul.2024.116966

[32] Vasudevan A, Maiya R, Venkatesh K, Kumar V, Sood P, Murthy K, et al. Transport of synaptic vesicles is modulated by vesicular reversals and stationary cargo clusters. Journal of Cell Science. 2023; 136(12): jcs261223. DOI: 10.1242/jcs.261223

Comparative Performance of Neural Networks and Ensemble Methods for Command Classification in ALEXA Virtual Assistant

Li Li
Artificial Intelligence and Big Data College, Chongqing Polytechnic University of Electronic Technology
Chongqing 401331, China
E-mail: 200708036@cqcet.edu.cn

Keywords: ALEXA virtual assistant, deep learning models, neural networks, random forest, command classification

Received: December 2, 2024

Our study investigates the classification of commands for the ALEXA virtual assistant using various machine-learning models. The dataset includes 16,521 samples, and data preprocessing steps, such as vectorization and remove all stop words and punctuation, were applied before training. Decision Trees, Random Forest, Hist Gradient Boosting, AdaBoost, and Neural Networks are employed to classify textual commands into respective classes. The dataset consists of commands and their classes, transformed into feature vectors using the TF-IDF method. Our neural network architecture comprises three dense layers and two dropout layers, totaling 272,850 trainable parameters, and uses RMSprop for optimization and categorical cross-entropy as the loss function. Performance is evaluated utilizing metrics like accuracy, precision, recall, and F1 score. Results have shown that neural networks perform better in comparison to classical algorithms and outperform AdaBoost explicitly in all metrics. The comparative results between neural networks and AdaBoost in evaluation metrics are, respectively, as follows:(0.851695 / 0.620157), (0.857729 / 0.771549),(0.851695 / 0.62057) and (0.85236 / 0.639389). Therefore, deep learning will indeed provide many promises toward solving challenging NLP tasks in a virtual assistant system like Alexa. The findings provide enormous insight into effective methodologies regarding the classification of commands and further establish the relevance of neural networks within extending virtual assistant technology. Further research may consider discussing more recent neural network structures and exploring their scalability and generalizability across several domains and languages.

Povzetek: Raziskava primerja učinek nevronskih mrež in ansambelskih metod pri razvrščanju ukazov v Alexa asistentu, kjer nevronske mreže dosegajo najboljše razultate.

1 Introduction

Whereas virtual assistants gave a whole new dimension to human-computer interaction, in facilitating technology towards and for people in every respect, the pioneering leader is ALEXA by Amazon. It grants users the ability to perform voice-controlled functions that range from managing smart home appliances to even playing music. The central role of ALEXA involves understanding and classifying commands.

Intent classification for virtual assistants is especially challenging because of the variability of natural language and the extensive range of tasks ALEXA can do. For a virtual assistant to function reliably and provide a seamless user experience, it is very important to accurately classify the command.

These models represent a wide variety of machine learning techniques used in this investigation, from the more traditional decision tree-based approaches to advanced ensembles and neural networks. Decision trees provide interpretability and simplicity, while ensemble methods, such as Random Forest, Histogram Gradient Boosting, and AdaBoost, use the power of bundling several weak learners together to improve classification performance. Furthermore, the neural network-based model gives a versatile structure that can fit complex patterns in data.

The aim is to find out the best model that classifies voice commands in virtual assistant ALEXA through hard experimentation and analysis. Some of the key parameters used in the evaluation include classification accuracy, computational efficiency, and scalability to handle high volumes. Other than this, we discuss the strengths and weaknesses of each model, which also constitutes insights into their performance and suitability against real-world deployment.

These findings contribute to the extension of machine learning knowledge in virtual assistant systems that have natural language processing tasks. The outcome is better represented in reality by designing and implementing a more efficient command classification system that is highly accurate to enhance the user experience with ALEXA and other similar voice assistants.

1.1 Main novelty

The most important novelty of this research includes an in-depth comparative analysis of machine learning models to

classify commands within ALEXA. The work will detail the performance of different models of the command classification in ALEXA, whereas other previous works may have focused on individual models or just general natural language processing tasks. We present the comparison of various models, starting from classic decision tree-based approaches and ending with ensembles such as Random Forest, Histogram Gradient Boosting, and AdaBoost, and one neural network-based model. The selection thus enables diverse exploration of methodologies and their suitability for the task at hand. We restrict our focus to the virtual assistant ALEXA; hence, these research findings should apply directly to this widely used and practical context and thus hopefully also contribute to tangible improvements in user experience and system reliability.

Furthermore, our study lets the models be scrutinized along several dimensions: not only for the accuracy of the classification but also for computational efficiency and scalability. This provides a more complete picture of the performance and practical viability of each model in real-world applications. By distilling the strengths and weaknesses of each model, we give great insight into the performance characteristics of the models, shedding light on why some of these models may be more suitable for command classification in ALEXA compared to others. Our study's results can further help in designing and implementing more efficient and effective command classification systems for ALEXA and similar virtual assistants. In turn, this may improve user experience and interaction with these platforms.

1.2 Related studies

With the improvement in techniques of machine learning and deep learning and also with the development of strong tools such as AI and virtual assistants, the researchers and the engineers went deep inside the aspects of text recognition, classification, speech recognition, and all those areas. It is a vast field covering a whole range of techniques that inspire new approaches toward solving these problems comprehensively.

Zhou et al. [1] present a new model, C-LSTM, that combines CNN and RNN architectures for sentence modeling in text classification. In C-LSTM, CNN is used to extract the representation of phrases, which is fed into an LSTM to produce the sentence representation. This model captures both local phrase features and global sentence semantics. These results indeed show that, for both sentiment and question classification tasks, C-LSTM outperforms CNN and LSTM. The results obtained were excellent for all tasks using the C-LSTM model.

Lai et al. [2] propose a new kind of recurrent convolutional neural network for text categorization that does not rely on human-designed features. The model learns the representation of words by composing the meaning of a sentence with a recurrent structure and a max-pooling layer to automatically identify keywords during classification. Experimental results on four datasets show that this approach outperforms the existing techniques, especially on document-level datasets.

A simple approach for detecting forge news is proposed by Granik & Mesyura [3] based on a naive Bayes classifier. They implemented such a system and performed its testing on the Facebook news posts dataset. The classifier achieved an accuracy of roughly 74% on the test set, impressive given the simplicity of the model. The paper reviewed several lines of research improving these results and pointed out a direction in which artificial intelligence approaches may help in dealing with the problem of fake news detection.

Tavčar et al. (2016) [4] introduce a web-based system designed to enhance virtual museum tours through the use of intelligent virtual assistants. This system analyzes user preferences to generate personalized exhibition recommendations. Furthermore, it incorporates a natural language interface, enabling users to ask questions and receive contextually relevant responses**.**

Kim et al. [5] investigate capsule networks for text classification, which is not well researched, although capsule networks have been very successful in image classification. This paper showed very promising results using capsule networks for text classification and their advantages when compared to convolutional neural networks. They further propose a simple routing method to reduce the computational complexity. Experiments on seven benchmark datasets show that capsule neural networks, using the routing technique presented in this paper, match the performance of traditional methods.

Qiao et al. [6] introduce a novel approach in this work to obtain task-specific distributed representations of n-grams for text classification by using "region embeddings." Using two different models to generate the embeddings, each word representation is combined through a weighting matrix to interact with the local context. These serve as parameters for the neural network classifier. It outperforms the existing methods on several benchmark datasets and effectively captures the important phrasal expressions present in texts, as evidenced by the experimental results.

Table 1: Examples of related works along with data

Reference	Task Type	Data	Accuracy
Lai et al. [2]	Convolutional Neural Networks	20Newsgroup/ 7520	CNN/94.79% RCNN/96.49%
		Fudan University document/ 8823	CNN/94.04% RCNN/95.20%
Kim et al. [5]	Classification using Capsules	20Newsgroup/10182	Capsule-A/80.39% Capsule-B/80.03%
		Reuters10/6472	Capsule-A/87.74% Capsule-B/87.96%
Qiao et al. [6]	Region Embedding for Text Classification	Amazon Review Polarity/ 3,000,000	VDCNN/95.7% D-LSTM/ -
		Sogou News/ 450,000	VDCNN/96.8% D-LSTM/94.9%

Islam et al. [7] present the Semantics Aware Random Forest (SARF) classifier, which selects the features relevant to the predicted classes. They assess SARF's performance on 30 real-world text datasets, comparing it with leading ensemble selection methods. The results show that SARF excels in textual information retrieval, indicating a promising new avenue for research on classifier interpretability.

Chen et al. [8] introduce NPMM, a nonparametric model designed for online short-text analysis. It utilizes auxiliary word embeddings to estimate topic numbers and tackle sparsity in topic-word distributions. NPMM automates the process of determining document-topic association by employing squared Mahalanobis distance, thereby eliminating the need for hyperparameter tuning in a new topic generation. Additionally, they suggest a nonparametric sampling strategy for identifying key terms within each topic. Inference is performed using a one-pass Gibbs sampling algorithm, augmented by a Metropolis-Hastings step. Experimental results demonstrate NPMM's superior performance compared to existing methods.

Yao et al. [9] present a text classification model that utilizes fastText. The model employs feature engineering to extract key information from the text and generates high-quality, low-dimensional, continuous representations using the fastText algorithm. The
experiment involves classifying a text dataset from the Baidu Dianshi platform's "user comment data emotional polarity judgment" using Python. Results from the emotional polarity judgment task indicate that the model outperforms traditional machine learning algorithms in precision, recall, and F-values, showcasing its strong classification performance.

Gargiulo et al. [10] explore Deep Learning architectures for text classification, focusing on extreme multi-class and multi-label tasks with hierarchical labels. They introduced the HLSE method for adjusting the data
label and evaluating a variety of WE model. Evaluation of the PubMed collection shows the effectiveness of HLSE and the importance of various WE model for such tasks.

Recently, Wang et al. [11] proposed Hierarchy-guided Contrastive Learning (HGCLR) to incorporate hierarchy into a text encoder. While training, HGCLR prepares the positive samples from the label hierarchy and lets the text encoder learn hierarchy-aware representation on its own. While in testing, the HGCLR-enhanced encoder can discard redundant hierarchy. The effectiveness of HGCLR has been well verified by intensive experiments on three benchmark datasets.

Sun et al. [12] introduced Clue and Reasoning Prompting (CARP), a method that uses a step-by-step reasoning approach tailored for text classification. CARP guides Language Models (LLMs) to detect basic clues such as keywords, tones, semantic connections, and references, which then trigger a diagnostic reasoning process for making final decisions. To tackle the issue of limited tokens, CARP utilizes a fine-tuned model on a supervised dataset for KNN demonstration search in in-context learning. This method combines the generalization ability of LLMs with task-specific evidence from the complete labeled dataset.

2 Utilized models, data and methods

This section presents a brief description of the executed dataset as well as the models. Also, the methodology of this study is explained.

Model Hyperparameters (default values)

We used Decision Tree, Random Forest, AdaBoost, and Histogram-based Gradient Boosting (Hist GB) models with their default parameters, as these values are commonly used in studies and provide reliable results. Additionally, details regarding the experimental setup, computational environment, and software libraries used have been provided for better understanding. The key default parameters are.

Model Hyperparameters (default values)

Decision Tree: max_depth=30, min_samples_split=2, min_samples_leaf=1, criterion="gini"

Random Forest: n_estimators=100, max_depth=30, min_samples_split=2, min_samples_leaf=1, criterion="gini"

AdaBoost: n_estimators=50, learning_rate=1.0, base_estimator=DecisionTreeClassifier(max_depth=1)

Histogram-based Gradient Boosting (Hist GB): learning_rate=0.1, max_iter=100, max_depth=30, l2_regularization=0.1, min_samples_leaf=20

Data Splitting Strategy

- The dataset was split into **75% training and 25% testing**.
- This process was repeated **5 times with random splits**, and the final results were reported as the **mean values** across these runs.

Computing Environment

- **RAM:** 16GB
- **CPU:** Ryzen 7
- **GPU:** NVIDIA RTX 3060
- **Software:** Python 3.10.*, TensorFlow, Scikit-learn

Neural Network Configuration

- **Train-Validation Split:** 10% of the training data was set aside for validation.
- **Batch Size:** 32
- **Epochs:** 30
- **Dropout Rate:** 0.2
- **Dense Layers:** Three layers, each with **128 neurons**.
- **Optimizer:** RMSprop with learning rate 2.5e-4 due to its fast convergence.

2.1 Data description

Aiming to classify the commands of ALEXA, they gathered various information related to the ALEXA virtual assistant. The dataset contains 16,521 records collected from the MASSIVE dataset' containing both the text of the commands and their corresponding class labels. The dataset encompasses approximately 20 distinct classes. Classification techniques were used to sort and categorize the data based on these labels.

Fig. 1 demonstrates the dispersion of the data in each category.

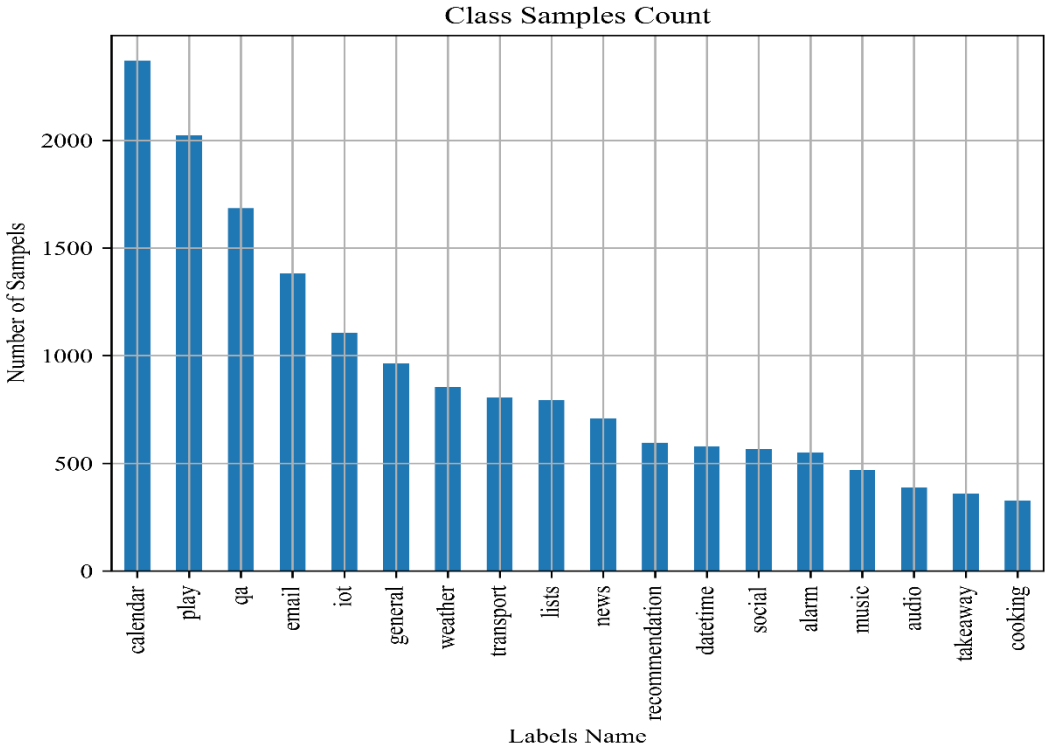


Figure 1: Number of samples in each label

According to the graphs in Fig. 1, the data with labels of calendar and play are in the majority, and labels such as cooking and takeaway contain the least amount of data. Each label contains various words with different frequencies. The following figures demonstrate word frequency in some of the labels presented in the word cloud as an example.

Figure 2: Word frequency in the label of the calendar

Fig. 2 displays that the label named calendar contains several words with close relation to the calendar. It displays those words such as calendar, meeting, event, remind, and today have been repeated more.

Figure 3: Word frequency in the label of cooking

The related words to cooking are demonstrated in Fig. 3. Accordingly, words such as recipe, cook, make, chicken, and find are more frequent in the commands related to cooking. This indicates that ALEXA users mostly used this VA to learn food recipes.

Figure 4: Word frequency in the label of transport

According to Fig. 4, in the commands related to transport, words like ticket, train, traffic, and book are more frequently, denoting the fact that people mostly used ALEXA to buy tickets and check the traffic.

Figure 5. Word frequency in the label of play

According to Fig. 5, the play label involves the words related to the commands for playing music and other media. Hence, words such as play, song, music, podcast, radio, and start are in the most repeated commands.

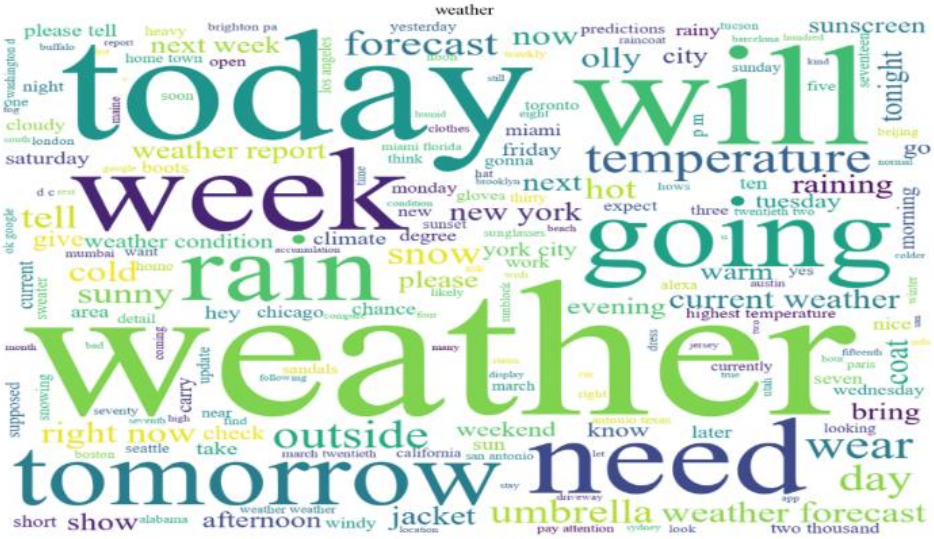


Figure 6. Word frequency in the label of weather

The label of weather contains all the command texts related to weather. According to the word cloud in Fig. 6, weather, today, will, week, need, rain, tomorrow, and will are the most frequent words in this class.

The following sections describe the compared models in this study.

2.2 Decision tree model

A decision tree classifier is a popular algorithm for classification, using a tree-like model where each internal node tests a specific feature, branches represent the decision outcome of those tests, and leaf nodes represent the class labels. This structure makes it intuitive to understand how decisions are made based on input features [13].

The algorithm begins by choosing the optimal feature from the dataset using specific criteria like information gain or Gini impurity. It then splits the dataset into subsets, starting with the selected feature. Each subset represents a different value of the feature. Each subset follows this iterative feature selection and partitioning process, recursively creating a tree structure until a stopping condition has been reached, such as when a maximum tree depth is reached or a minimum number of samples in a node.

Once the tree has been constructed, any instance can be classified by starting at the root and working its way down to a leaf. At each node, it considers the test on the feature and heads down the appropriate branch until it reaches a leaf. This gives the instance the class label of the

leaf that it ends up at. The ease with which this process can be followed makes decision trees easily understandable and interpretable and forms one of the major advantages of the technique.

With their great power, however, comes great susceptibility to overfitting in the case of large decision trees. To overcome this, several methods can be performed: pruning and tuning of parameters such as maximum depth or minimum samples per leaf. Then there are the ensemble methods, which include Random Forests and GBMs, where many trees are combined with the intent of reducing overfitting and further improving performance. Generally, decision tree classifiers offer a versatile and interpretable approach to classification tasks, with several strategies that may be used to improve performance and generalization capabilities [14], [15].

2.3 Random forest model

The Random Forest ensemble is resistant and involves learning for both classification and regression problems. It constructs a large number of decision trees during training and combines their outputs to yield the outcome. Each tree in the Random Forest is trained on a different subset of the data through a process called bootstrap sampling, meaning that random samples are drawn with replacement from the training data.

In the decision trees formed in a Random Forest, each node will not consider all features on which to split; instead, a random subset of the features alone would be considered for a split. That injects randomness into the model and helps with decorating the trees, yielding an ensemble that's more robust and less prone to overfitting. The best split at each node is determined based on a chosen criterion such as Gini impurity or information gain. This is an iterative process that repeats either until the trees are full or the stopping criterion is reached. For classification tasks, each tree in the Random Forest "votes" for a class, and the class receiving the most votes is chosen as the final prediction. For regression tasks, the final prediction is gained by averaging the predictions from all the trees.

Random Forests provide several benefits. They are resilient to noise and outliers due to the averaging effect of multiple trees. They also offer insights into feature importance, helping identify which features significantly impact predictions. Further, the training of individual trees can be parallelized, thus making them efficient for large datasets.

However, with these strengths, Random Forests are quite computationally expensive. In cases of high linearity among the features-target variables relationships or high-dimensional data, its performance may not be at par with other techniques. Nevertheless, because it has robustness and effectiveness in solving a wide variety of problem types, Random Forests find wide use in practice in a wide array of domains [16], [17].

Random Forests can be represented mathematically, although it's a bit more complex than a single decision tree due to the ensemble nature of the model.

Let $X = (X_1, X_2, \dots, X_n)$ be the input features and Y be the target variable.

A Random Forest consists of B decision trees $T_1, T_2, \dots, T_B$ each trained on a bootstrap sample of the data. The prediction for a new instance x in a classification problem is obtained by taking a majority vote among the predictions of all trees:

$$\hat{\boldsymbol{y}}_{\boldsymbol{RF}}(\boldsymbol{x}) = \boldsymbol{mode}\left(\hat{\boldsymbol{y}}_{\boldsymbol{1}}(\boldsymbol{x}), \hat{\boldsymbol{y}}_{\boldsymbol{2}}(\boldsymbol{x}), \dots, \hat{\boldsymbol{y}}_{\boldsymbol{B}}(\boldsymbol{x})\right) \quad \textbf{(1)}$$

Where $\hat{y}_i(x)$ is the predicted value of the i-th tree.

2.4 Histogram gradient boosting

The Histogram-based Gradient Boosting Classifier is a robust machine learning algorithm for classification tasks, especially on big data. It falls into the category of a gradient-boosting family that combines a sequence of weak learners, usually decision trees, into a robust classifier.

Perhaps the most salient feature of the Hist Gradient Boosting Classifier is the unique approach to feature binning via histogramming. This avoids the use of exact algorithms that usually find split points in decision trees and instead discretizes features into intervals, hence making it way faster during training, particularly for datasets with a large number of samples.

Similar to other boosting gradient-type methods, the Hist Gradient Boosting Classifier builds trees consecutively in a greedy manner such that the newest tree in each step minimizes the previously incurred loss. Adding more trees in a sequence reduces some loss of objective function—e.g., deviance or exponential loss for classification—leading to an increase in the predictive power of the model. The Histogram Gradient Boosting Classifier uses regularisation to avoid overfitting by limiting the tree depth and the minimum number of samples to create a leaf and to split a node for good generalization to unseen data.

Another advantage of the Hist Gradient Boosting Classifier involves the handling of missing values in the dataset, both during training and prediction, without needing to take prior care of imputation. This facility makes it highly convenient when dealing with real datasets that are inherently bound to show some data loss.

Due to its histogram-based approach, the Hist Gradient Boosting Classifier is highly scalable and efficient and is hence indicated in problems with a high number of samples and features. Like other gradient-boosting methods, it is also robust to outliers in data and can handle mixed types of features, including categorical and numerical variables [18], [19].

2.5 AdaBoost model

AdaBoost-Adaptive Boosting is a classification ensemble learning technique that combines several weak learners into one robust learner. The basic principle behind AdaBoost lies in training multiple weak learners—usually simple decision trees—on a training set sequentially while focusing on instances difficult to classify.

All instances in the dataset have an equal weight to start training. A weak learner is then fitted on this data and tested for performance. Weights are increased for those instances that have been classified incorrectly, while the weights are decreased for the instances correctly classified in successive rounds. The adaptive fashion of this method will make subsequent weak learners focus more on instances that were challenging to classify in earlier rounds. It involves training each of these weak learners sequentially in a manner that each of them tries to correct mistakes made previously by the other weak learners. Several of these weak learners are trained one after another, and all their predictions are combined to have the final prediction. They all contribute to the final prediction. Their contribution is weighted by their accuracy.

AdaBoost works effectively for both binary and multiclass classification problems. It is very effective to deal with complicated data since it can capture complex boundaries of the decisions. AdaBoost may be sensitive to outliers and noisy data, which could degrade the performance. Though limited, AdaBoost is still popular owing to its simplicity and efficiency. Besides, it often acts as a base algorithm in advanced ensemble methods like GBM and XGBoost, further enhancing its performance and robustness [20], [21].

2.6 Neural Network model

A neural network model is a computational framework of interconnected nodes, or neurons, in layers inspired by the structure and function of biological neural networks, such as the human brain. Each neuron receives an input, undergoes processing of that input, and sends it to the next layer; this is accomplished with the interlinking of neurons through weights according to their strength.

It is trained in a procedure called supervised learning, where the model learns to relate input data to output labels by adjusting the weights of the connections among neurons. By repeatedly exposing the model to training data, it learns to recognize patterns and, eventually, to make predictions or classifications. They find applications in many areas, such as image and audio recognition and natural language processing, because they can find complex patterns from enormous data [22].

The neural network model in this study comprises three dense layers and two layers of dropout. In total, the neural network model contains 272,850 trainable parameters.

Dense layer: It forms one of the major components in neural network architectures. Every neuron in this layer is normally connected to every neuron of the previous layer, thus creating a dense connectivity structure. Hence, each neuron's output is dependent on all neurons in the preceding layer. Each such connection is also associated with a learned weight in training. The most common places for which dense layers occupy are in the middle of the neural network, allowing it to capture intricate patterns within the data. Normally, after each of them comes an activation function that may introduce non-linearity into the network, hence enabling it to learn and represent complex relationships among features of data in a more effective way.

Dropout in neural networks is a regularization approach that works by randomly deactivating a portion of the neurons during training to prevent overfitting. During each training iteration, dropout arbitrarily turns a portion of the outputs from the neurons to zero. This forces the network to learn redundant representations of its data, reducing the risk of relying too heavily on any particular feature or combination of features. Dropout effectively simulates training multiple networks with different architectures simultaneously, resulting in a more robust model that generalizes better to unseen data. Typically, dropout is applied to hidden layers of the network, with a dropout rate parameter controlling the fraction of neurons to deactivate.

We also utilized the categorical cross entropy as the loss function in the neural network model. Categorical cross-entropy stands as a frequently employed loss function in machine learning, particularly for tasks involving multi-class classification. Its role is to evaluate the discrepancy between the actual distribution and the predicted distribution of categorical variables. In classification problems, the output variable is categorical, meaning it falls into one of several classes. The true distribution represents the actual classes, typically encoded as a one-hot vector, where only one element (the true class) is 1, and the rest are 0s. The predicted distribution, on the other hand, represents the model's probabilities for each class. Mathematically, categorical cross-entropy is calculated using the following formula for a single example:

$$\boldsymbol{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_i \boldsymbol{y}_i \log(\hat{\boldsymbol{y}}_i) \quad \textbf{(2)}$$

Where y is the true distribution (the one-hot encoded vector). $\hat{y}$ is the predicted distribution (the vector of probabilities). Class i's true probability is denoted by y_i. The anticipated probability of class i is shown by $\hat{y}_i$.

The sum is taken over all classes. This formula is applied for every example within the dataset and then averaged across all examples to make a final calculation of loss.

That's because it is trained to penalize the model when it has a lower probability of the true class by quantifying the difference between predicted and true distributions. This is appropriate for problems with mutually exclusive classes, in which each instance is assigned only to one class. Minimizing this categorical cross-entropy loss, therefore, allows the model to generate probabilities that align well with the actual distribution, hence increasing its accuracy of classification [23].

To enhance the performance of the neural network, this model is optimized using the RMSprop algorithm. ***RMSprop***, or Root Mean Square Propagation, is a popular choice for training neural networks since it avoids the pitfalls of normal stochastic gradient descent by adapting the learning rate of each parameter based on the magnitudes of the gradients. Such adaptiveness provides

faster convergence and better handling of non-stationary and sparse gradients in tasks involving deep learning.

The RMSProp works by giving each parameter a different learning rate, meanwhile computing the exponential moving average of the squared gradients. This is achieved through an exponentially decaying average calculation:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta) g_t^2 \quad (3)$$

Here, g_t represents the gradient of the parameter at time step t, and β controls the decay rate of the moving average. Typically, β is set to a value like 0.9. The moving average $E[g^2]_t$ accumulates the squared gradients over time, providing an estimate of the variance of the gradients.

The update rule in RMSprop is then adjusted using the square root of the accumulated squared gradients, effectively scaling the learning rates for each parameter:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_t \quad (4)$$

When the parameter at time step t is denoted by θ_t, the learning rate is indicated by η, and ϵ is a tiny constant introduced for numerical stability to avoid division by zero. The division by the square root of the accumulated squared gradients normalizes the learning rate such that updates corresponding to large gradients are reduced and vice versa.

RMSProp is particularly suited for deep learning tasks, wherein the data may have different gradients and scales, with its adaptive learning rate. Nevertheless, similar to many of the optimization algorithms, RMSProp has several hyperparameters, such as the learning rate η and the decay rate β, that need further fine-tuning to attain optimal performance on different applications [24], [25].

2.7 Data preprocessing

First, the text is vectorized using one of the most salient techniques for text analysis and document classification: the TF-IDF method. TF-IDF (Term Frequency-Inverse Document Frequency) is a crucial text preprocessing technique used to convert textual data into numerical representations for machine learning models. TF-IDF involves processes such as tokenization (where text is divided into individual words or tokens), stemming/lemmatization (where words are reduced to their root form to standardize variations), TF calculation (where the frequency of each word in a document is computed), IDF calculation (where the inverse document frequency is computed to down weight common words across multiple documents), and TF-IDF score computation (where each word's TF is multiplied by its IDF score to determine its importance within the document). TF-IDF helps highlight important words in a document while reducing the impact of common but less meaningful terms.

The unigrams, bigrams, and trigrams included in the vectorization process incorporated more intricate patterns and connections from within the textual data.

After vectorization, we had 2057 unique words and important phrases for our dataset. Further, in improving the quality of representation, all stop words and punctuation were removed from the text. This removes the noise in the texts and puts the attention of the model on meaningful content.

Then, the data were divided into training and testing sets. There were 13,217 records designated for training the models to learn trends and relationships from them. 3,304 records were set aside to test the performance of the models. This separation of data into training and testing sets helps to assess how well the trained models generalize to unseen data and provides a measure of their predictive accuracy.

3 Analysis results

3.1 Neural network validation

In the first step, we analyzed the loss values and accuracy of the neural network model during training and testing. The Fig. 7 in the following demonstrates the results.

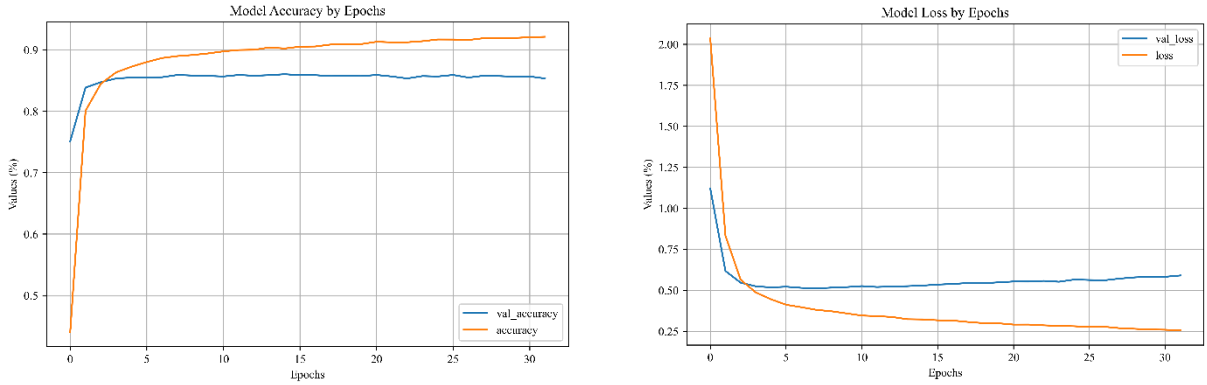


Figure 7: The validation results of the neural network model

Fig. 7a displays the accuracy of the neural network model after 30 epochs. It indicates that the accuracy of the model in total increases in the earlier epochs and stays in the high values as the epochs advance. The training accuracy rapidly rises within the first few epochs and then stabilizes

around 92%, while the validation accuracy follows a similar pattern but remains slightly lower, around 86%. This gap between training and validation accuracy suggests that the model learns well on training data but has some overfitting tendencies.

Furthermore, the loss value of the model during training and testing in Fig. 7b denotes that the loss value of the neural network model in the training process decreases gradually as the epochs go forward. The loss initially drops sharply within the first few epochs, showing rapid learning progress, then continues to decrease more gradually. The training loss consistently reduces, reaching approximately 0.2, while the validation loss exhibits a different trend.

By scrutinizing the loss value of the test (val_loss), it is derived that the loss values at the earlier epochs drop. However, as the epochs advance, the loss value of the model starts increasing to a small number due to overfitting; hence, the validation process after 30 epochs is stopped. This increase in validation loss while training loss continues to decrease suggests that the model is memorizing patterns from the training data rather than generalizing well to unseen data. Techniques such as early stopping, dropout regularization, or data augmentation could help mitigate overfitting and improve the model's ability to generalize.

3.2 Evaluation metrics

The studied models used in this analysis, which are introduced in sections 2.2 to 2.6, are evaluated through well-known metrics such as accuracy, precision, recall, and f1 score. These metrics are used to evaluate the performance of classification models, especially in the context of machine learning and statistics.

Accuracy is the most straightforward metric and measures the ratio of correctly predicted instances to the total instances. It is computed by dividing the total number of forecasts by the number of correct guesses.

$$\boldsymbol{Accuracy} = \frac{(\boldsymbol{TP+TN})}{\boldsymbol{total\ population}} \quad \textbf{(5)}$$

Precision assesses the proportion of accurately predicted positive observations out of all predicted positive observations. It reflects the model's effectiveness in minimizing false positives.

$$\boldsymbol{Precision} = \frac{(\boldsymbol{TP})}{(\boldsymbol{TP+FP})} \quad \textbf{(6)}$$

High precision means that an algorithm returned substantially more relevant results than irrelevant ones, though it might miss some relevant results (low recall).

Recall or sensitivity measures the proportion of accurately predicted positive observations compared to all actual positive instances in the dataset. It evaluates the model's ability to detect all positive cases.

$$\boldsymbol{Recall} = \frac{(\boldsymbol{TP})}{(\boldsymbol{TP+FN})} \quad \textbf{(7)}$$

High recall means that an algorithm returned most of the relevant results (low false negatives), though it might also bring back many irrelevant results (low precision).

The ***F1 score*** represents the harmonic mean of precision and recall, offering a balanced single score that considers both precision and recall.

$$\boldsymbol{F1\ score} = \mathbf{2} \times \frac{(\boldsymbol{Precision} \times \boldsymbol{Recall})}{(\boldsymbol{Precision} + \boldsymbol{Recall})} \quad \textbf{(8)}$$

The F1 score is a good measure of a model's accuracy, especially when there is an uneven class distribution (imbalanced data).

The obtained results for the aforesaid metrics are presented in the following table:

Table 2: The metrics results for the models

	Neural Network	Decision Tree	Random Forest	AdaBoost	Hist GB
Accuracy	0.851695	0.81023	0.831114	0.620157	0.804782
Precision	0.857729	0.815996	0.837435	0.771549	0.822176
Recall	0.851695	0.81023	0.831114	0.62057	0.804782
F1 score	0.85236	0.810181	0.830936	0.639389	0.80615

The results show the neural networks' superior performance. Neural Network achieves the highest accuracy (0.8517) and strong performance across all metrics, making it the most effective model overall. Decision Tree and Random Forest also perform well, with Random Forest slightly outperforming Decision Tree in all metrics.

AdaBoost shows significantly lower performance, particularly in accuracy (0.6202) and F1 score (0.6394), indicating it may not be well-suited for this dataset.

Hist Gradient Boosting (Hist GB) performs competitively, with results close to those of the Decision Tree and Random Forest models. To get a better understanding of the obtained results, Fig. 8 presents the graphic comparison.

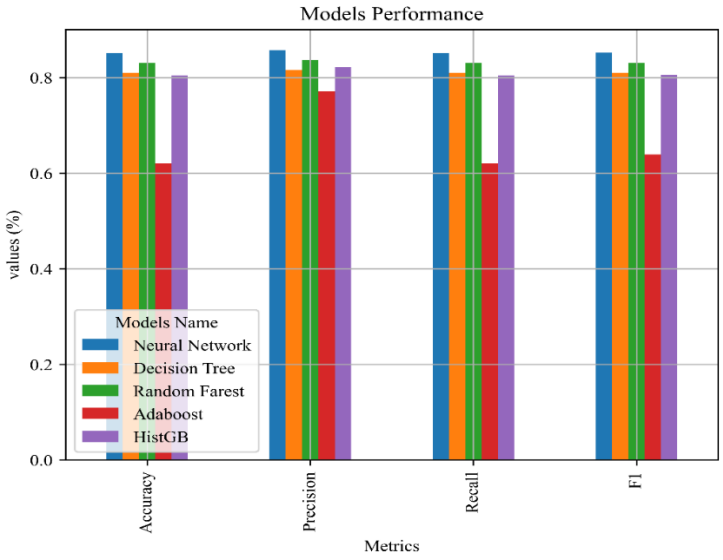


Figure 8: Comparison of the obtained results for the models

By comparing the accuracy of the models, it is deduced that the neural network model holds the highest accuracy value, and the random forest and the decision tree models are the second and third-best models. The precision of the models demonstrates the neural network model shows promising value with the best-obtained value among the studied models. The results of the recall denote the outperformance of the neural network model compared to the other exerted models. The results indicated the disappointing results of the AdaBoost. According to the f1 score of the models, the neural network presents the highest obtained value, and the random forest and decision tree are the next beneficial models.

In total, the neural network consistently outperforms all other models across all metrics utilized in this study. Conversely, AdaBoost exhibits the lowest performance across the board. This highlights the superior effectiveness and robustness of the neural network over its counterparts, making it the clear choice for the task at hand.

3.3 Class correlation in the neural networks

Since the neural network model was the best model for accomplishing the given tasks, the class characteristics of the neural networks are delved into, rigorously. The following figure demonstrates the confusion matrix among the classes.

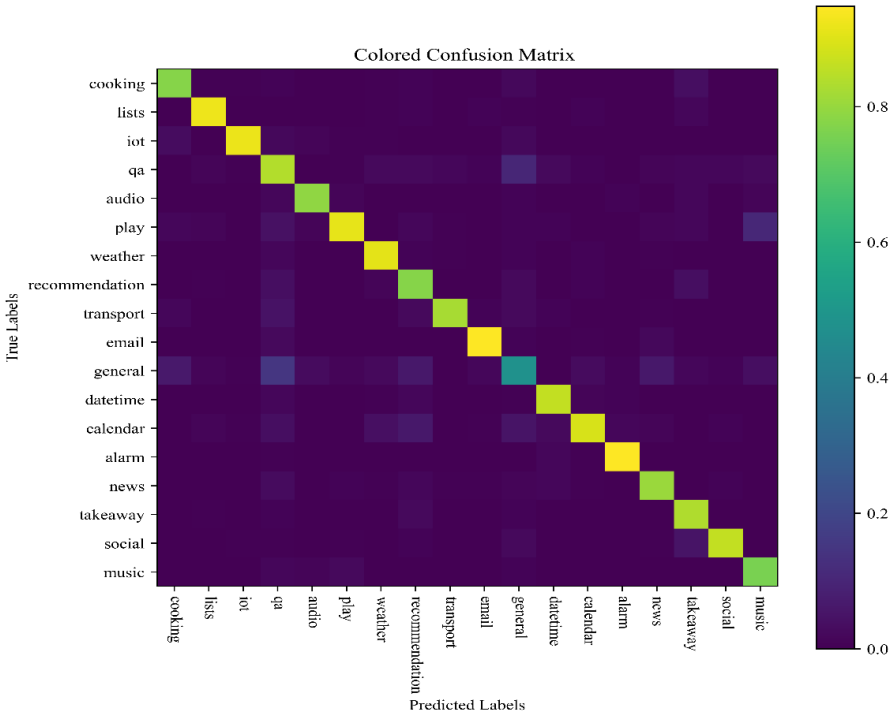


Figure 9: Confusion matrix of neural network classes

Fig. 9 displays the confusion matrix, which denotes which classes are mistaken with each other. This happens due to the similarity of data between the two classes. In Fig. 9, the lighter colors represent more similarity between classes. As an example, the general and qa contain resembling textual data and are sometimes mistaken with each other by the models. In other words, "light colors" in the confusion matrix indicate higher values of classification accuracy, representing a stronger correlation between true and predicted labels.

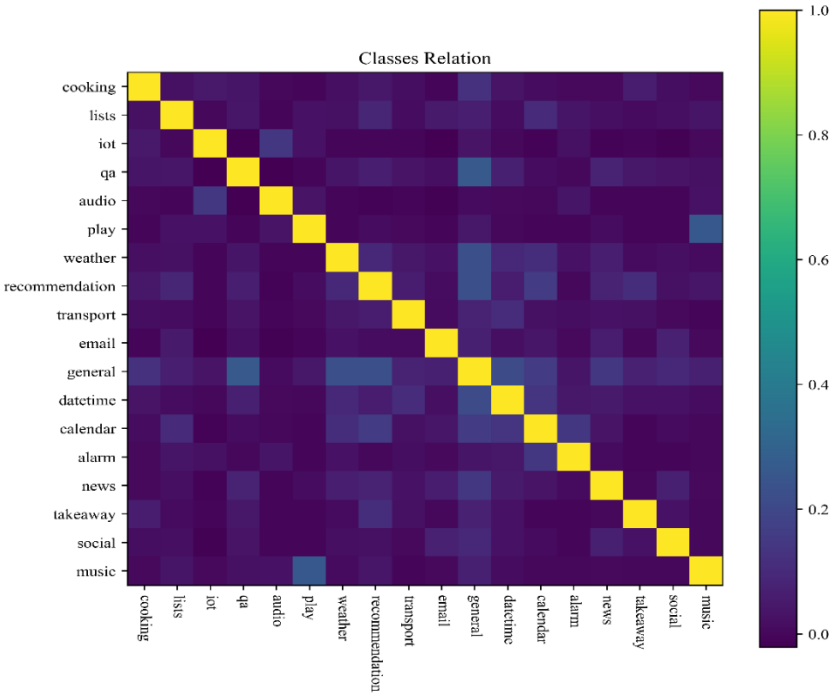


Figure 10: Class relation in the neural network model

Fig. 10 demonstrates the relationship between classes in the neural network model. The lighter colors indicate the higher relation between the two classes. Two classes with high relations denote that the classes share the same commands and contain resembling textual data. According to the matrix, the general has a high relationship with classes such as weather, recommendation, and qa. Also, play and music share similar commands.

Analyzing the confusion matrix, we can observe areas where the model misclassified instances. The off-diagonal elements in the matrix highlight these misclassifications.

For example, there is a noticeable overlap between the "lists" and "recommendation" classes. This could stem from shared contextual words in their respective command texts, such as "add," "suggest," or "list." Enhancing the feature extraction process or incorporating contextual embeddings may help differentiate these classes better.

The confusion between "news" and "general" is evident, likely because commands labeled as "general" might occasionally reference current events or updates, leading the model to associate them with "news." Using additional semantic analysis might mitigate this issue.

4 Discussion

Highest Values Across All Metrics: It can be observed that the Neural Network outperforms other models across all evaluation metrics, achieving the highest scores. This indicates that the neural network has the lowest error in its predictions and is the most optimized model for this task.

Traditional models like Decision Trees and Random Forests have structural limitations that prevent them from effectively capturing complex relationships in the data. In contrast, a neural network, with multiple hidden layers, can better learn nonlinear and intricate patterns within the dataset. In other words, the flexibility in learning complex patterns is one of the advantages of this model.

Deep learning also offers advantages in natural language processing (NLP). Neural networks are particularly well-suited for Natural Language Processing (NLP) tasks. Unlike classical models based on decision trees, neural networks can better understand semantic relationships between words and phrases. Given that the goal of this research is to classify voice commands in ALEXA, this capability makes the neural network a superior choice.

The proposed model utilizes improved optimization and overfitting prevention techniques. The neural network architecture benefits from techniques such as Dropout and RMSprop optimization, which improve the model's generalization ability. This ensures that the model maintains high performance on new, unseen data.

Neural networks often excel in text classification because they can automatically learn complex and hierarchical representations from raw text data. In contrast with traditional machine learning methods, which usually rely on manually engineered features or simpler representations, neural networks (especially architectures

like CNNs, RNNs, or Transformers) can capture intricate semantic relationships and contextual nuances, leading to improved performance.

One possible reason AdaBoost underperforms in this context is its reliance on weak learners (often decision trees) that might not capture the complexity of text data as effectively as neural networks. Text data is typically high-dimensional and sparse, making it challenging for boosting methods that build models sequentially. If the individual weak learners cannot handle the intricacies of word order, context, and nuanced semantics, the ensemble may struggle even after several iterations. Additionally, AdaBoost is sensitive to noisy data and outliers, which are common in natural language tasks, further impeding its performance.

Additionally, the performance and scalability of the models used in the article were compared with each other. Decision Trees Fast in training and prediction but prone to overfitting. Good scalability for small datasets Random Forest Higher accuracy than decision trees with reduced overfitting but slower training. Moderate scalability. Hist Gradient Boosting More efficient than classical boosting methods, suitable for large datasets, but requires careful hyperparameter tuning. AdaBoost Performs well on small datasets but is sensitive to noise and less scalable than modern boosting methods. Neural Networks Capable of learning complex patterns but computationally expensive and requires large datasets for optimal performance. High scalability with GPU/TPU acceleration.

Statistical

The following is the t-test for neural network and random forest for accuracy precison and recall

T-Test Results:
Accuracy: t = 4.1309, p = 0.0006
Precision: t = 3.8258, p = 0.0012
Recall: t = 6.3499, p = 0.0000
F1-Score: t = 2.5812, p = 0.0188

5 Conclusion

In conclusion, our research offers an in-depth analysis of command classification for the ALEXA virtual assistant, utilizing various machine learning models and methods. By utilizing Decision Trees, Random Forest, Hist Gradient Boosting, AdaBoost, and Neural Networks, we aimed to discern the most effective approach for accurately categorizing user commands.

Our study capitalized on a dataset containing text-based commands and their corresponding classes, which were transformed into feature vectors using the TF-IDF method. Notably, our neural network architecture was composed of three dense layers and two dropout layers, comprising a total of 272,850 trainable parameters. The loss function used was the categorical cross-entropy, while RMSProp was used to ensure sound optimization for the training of the neural network model.

In this respect, our results unambiguously prove the superiority of neural network models compared to other conventional machine learning algorithms through extensive evaluation by using precision, accuracy, F1 score, and recall metrics. More precisely, the neural networks were always ranked first compared to AdaBoost in terms of classification performance for all measured parameters.

This work provides insight into the performance of various machine learning models for command classification but points out the very important potential use of neural networks in dealing with complex natural language processing tasks. The top performance given by the relative performance of neural networks in this work points to their suitability for real-world applications within virtual assistant frameworks such as ALEXA.

Furthermore, our findings point out how state-of-the-art techniques, such as deep learning, will have to be explored for high accuracy and robust performance in command classification tasks. The success of neural networks here indicates how these approaches are relevant within virtual assistant technology, opening new paths toward the development of more fluid user experiences and natural ways of interaction.

These findings have significant practical implications. By leveraging neural networks, ALEXA's command classification can be directly improved in real-world scenarios. For instance, better classification accuracy can lead to enhanced user satisfaction by reducing misinterpretation of commands. The findings also suggest that integrating more complex neural architectures could allow ALEXA to handle a broader variety of commands, including those involving nuanced or multi-intent queries. Additionally, the scalability demonstrated in this research implies that the models can adapt to diverse languages and accents, improving accessibility for users worldwide.

In further research, more neural network architectures can be compared and more feature extraction methods could be explored, or even ensemble techniques could be considered to improve the performance on command classification tasks. Additionally, the scalability and generalizability of the proposed models could be evaluated across different domains and languages to assess their broader applicability. Overall, our study contributes valuable insights into the field of natural language processing and lays the foundation for continued advancements in virtual assistant technologies.

Nomenclature

Abbreviation	Description	Abbreviation	Description
VA	Virtual Assistant	X	Input feature
AdaBoost	Adaptive Boosting	Y	Targe value
AI	Artificial Intelligence	$\hat{y}_i(x)$	Predicted value of i-th tree in random forest

CNN	Convolutional Neural Network	y_i	True probability of class i
LSTM	Long Short-Term Memory	$\hat{y}_i$	Predicted probability of class i
SARF	Semantics Aware Random Forest	ϵ	Small constant number
NPMM	nonparametric model	g_t	The gradient of the parameter at iteration t
HLSE	Hierarchical Label Set Expansion	β	Parameter to control the decay rate
HGCLR	Hierarchy-guided Contrastive Learning	θ_t	Parameter at time step t
WE	Word Embedding	η	Learning rate
CARP	Clue and Reasoning Prompting	TP	True positive
GBM	Gradient Boosting Machines	TN	True negative
RMSprop	Root Mean Square Propagation	FP	False positive
SGD	stochastic gradient descent	FN	False negative

Acknowledgements

I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Competing of interests

The authors declare no competing of interests.

Authorship contribution statement

The author contributed to the study's conception and design. Data collection, simulation, and analysis were performed by "Li LI ". Also, the first draft of the manuscript was written by Li LI commented on previous versions of the manuscript.

Data availability

Data can be shared upon request.

Declarations

Not applicable

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Ethical approval

The research paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM Neural Network for Text Classification," Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08630

[2] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," 2015. [Online]. Available: www.aaai.org

[3] M. Granik and V. Mesyura, "Fake News Detection Using Naive Bayes Classifier," in IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2017.

[4] A. Tavčar, C. Antonya, and E. V. Butila, "Recommender System for Virtual Assistant Supported Museum Tours," 2016. [Online]. Available: http://www.projekt-

[5] J. Kim, S. Jang, S. Choi, and E. Park, "Text Classification using Capsules," Aug. 2018, [Online]. Available: http://arxiv.org/abs/1808.03976

[6] C. Qiao et al., "A NEW METHOD OF REGION EMBEDDING FOR TEXT CLASSIFICATION," 2018.

[7] M. Z. Islam, J. Liu, J. Li, L. Liu, and W. Kang, "A semantics aware random forest for text classification," in International Conference on Information and Knowledge Management, Proceedings, Association for Computing Machinery, Nov. 2019, pp. 1061–1070. doi: 10.1145/3357384.3357891.

[8] J. Chen, Z. Gong, and W. Liu, "A nonparametric model for online topic discovery with word embeddings," Inf Sci (N Y), vol. 504, pp. 32–47, Dec. 2019, doi: 10.1016/j.ins.2019.07.048.

[9] T. Yao, Z. Zhai, and B. Gao, "Text Classification Model Based on fastText," in IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), 2020.

[10] F. Gargiulo, S. Silvestri, M. Ciampi, and G. De Pietro, "Deep neural network for hierarchical extreme multi-label text classification," Applied Soft Computing Journal, vol. 79, pp. 125–138, Jun. 2019, doi: 10.1016/j.asoc.2019.03.041.

[11] Z. Wang, P. Wang, L. Huang, X. Sun, and H. Wang, "Incorporating Hierarchy into Text Encoder: a

Contrastive Learning Approach for Hierarchical Text Classification," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.03825

[12] X. Sun et al., "Text Classification via Large Language Models," 2023.

[13] I. Gounari and M. Kanzilieris, "Wireless Sensor Networks Focusing on Predicting Average Localization Error through Machine Learning Applications," Journal of Artificial Intelligence and System Modelling, vol. 01, no. 04, pp. 104–118, 2024, doi: 10.22034/jaism.2024.474005.1055.

[14] A. Navada, A. Nizam Ansari, S. Patil, and B. A. Sonkamble, "Overview of Use of Decision Tree algorithms in Machine Learning," IEEE Control and System Graduate Research Colloquium, 2011.

[15] Y. Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," Shanghai Arch Psychiatry, vol. 27, no. 2, pp. 130–135, Apr. 2015, doi: 10.11919/j.issn.1002-0829.215044.

[16] G. Biau and E. Scornet, "A Random Forest Guided Tour," Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.05741

[17] G. Louppe, "Understanding Random Forests: From Theory to Practice," Jul. 2014, [Online]. Available: http://arxiv.org/abs/1407.7502

[18] Y. J. Ong, Y. Zhou, N. Baracaldo, and H. Ludwig, "Adaptive Histogram-Based Gradient Boosted Trees for Federated Learning," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.06670

[19] A. Guryanov, "Histogram-Based Algorithm for Building Gradient Boosting Ensembles of Piecewise Linear Decision Trees," in 8th International Conference, AIST 2019 Kazan, Russia, July 17–19, Goos Gerhard and Hartmanis Juris, Eds., Kazan: Springer, Jul. 2019, pp. 39–50.

[20] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class AdaBoost *," 2009.

[21] R. E. Schapire, "Explaining AdaBoost," 2013.

[22] Y. Goldberg, "Neural Network Methods for Natural Language Processing," Synthesis Lectures on Human Language Technologies, vol. 10, no. 1, pp. 1–311, 2017, doi: 10.2200/S00762ED1V01Y201703HLT037.

[23] Z. Zhang and M. R. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels," 2018.

[24] T. Kurbiel and S. Khaleghian, "Training of Deep Neural Networks based on Distance Measures using RMSProp," Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.01911

[25] R. Elshamy, O. Abu-Elnasr, M. Elhoseny, and S. Elmougy, "Improving the efficiency of RMSProp optimizer by utilizing Nestrove in deep learning," Sci Rep, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-35663-x.

Intelligent Cognitive System for Computational Psychotherapy with a Conversational Agent for Attitude and Behavior Change in Stress, Anxiety, and Depression

Tine Kolenik
Institute of Synergetics and Psychotherapy Research, University Hospital of Psychiatry, Psychotherapy and Psychosomatics, Paracelsus Medical University, Salzburg, Austria
E-mail: tine.kolenik@gmail.com

Thesis Summary

Keywords: generative artificial intelligence, attitude and behavior change support systems, digital mental health, intelligent cognitive conversational agent, artificial cognitive architecture, machine learning

Received: March 28, 2025

The paper summarizes a Doctoral Thesis presenting a computational psychotherapy system for stress, anxiety, and depression (SAD) using a conversational agent. The first contribution is a novel panel dataset combining quantitative diagnostic-level questionnaires and qualitative daily diaries. The second contribution is the system itself, built upon a cognitive architecture simulating Theory of Mind through an ensemble of user, machine learning, and knowledge models for SAD prediction, forecasting, and personalized intervention.

Povzetek: Članek povzema doktorsko disertacijo, ki predstavlja sistem računalniške psihoterapije za stres, anksioznost in depresijo (SAD) z uporabo pogovornega agenta. Prvi prispevek je nova panelna podatkovna zbirka, ki združuje kvantitativne diagnostične vprašalnike in kvalitativne dnevne zapise. Drugi prispevek je sam sistem, zgrajen na kognitivni arhitekturi, ki simulira teorijo uma z ansamblom različnih uporabniških, umetnointeligenčnih in ekspertnih modelov za zaznavanje, napovedovanje in personalizirano intervencijo pri uporabnikih s SAD.

1 Introduction

The growing burden of mental health issues, particularly stress, anxiety, and depression (SAD), necessitates innovative and accessible support systems [1]. Computational psychotherapy, utilizing tools like intelligent conversational agents (ICAs), offers a promising avenue. However, current state-of-the-art (SOTA) systems often lack sophisticated user modeling, personalization, adaptation, and forecasting capabilities, limiting their effectiveness for dynamic mental health needs [2]. Many also struggle with safe and domain-specific language generation.

This thesis addresses these gaps by proposing a novel computational psychotherapy system centered around an ICA with Theory of Mind (ToM). It makes two core contributions: 1) The creation of a unique, high-quality, mixed-methods panel dataset specifically designed for developing and evaluating such systems. 2) The design and implementation of the system's cognitive architecture (CogA) which distinctively simulates ToM [3]. ToM simulation allows the system to model user mental states (beliefs, emotions, intentions) and respond adaptively and persuasively, aiming to predict, forecast, and positively influence the user's SAD state.

2 Dataset and cognitive architecture

A key contribution was the creation of a novel "golden standard" dataset. Data was collected using Ecological Momentary Assessment (EMA) via the Synergetic Navigation System (SNS) application from 50 participants over approximately four weeks [4]. This resulted in a panel dataset of 1495 instances, each containing daily quantitative scores from an 18-item SAD symptom questionnaire and qualitative text diary entries (~150 words minimum) describing the user's mood, experiences, and thoughts. Big Five (B5) personality traits [5] and demographics were also collected.

The system's core is its CogA designed to simulate ToM. It comprises three main modules (Figure 1):

Natural Language Processing (NLP): Handles user text input via Dialog Management (tracking conversation state), Natural Language Understanding (sensibility filtering), and Feature Extraction (using LIWC and VADER to create numeric representations of mental dimensions [4]).

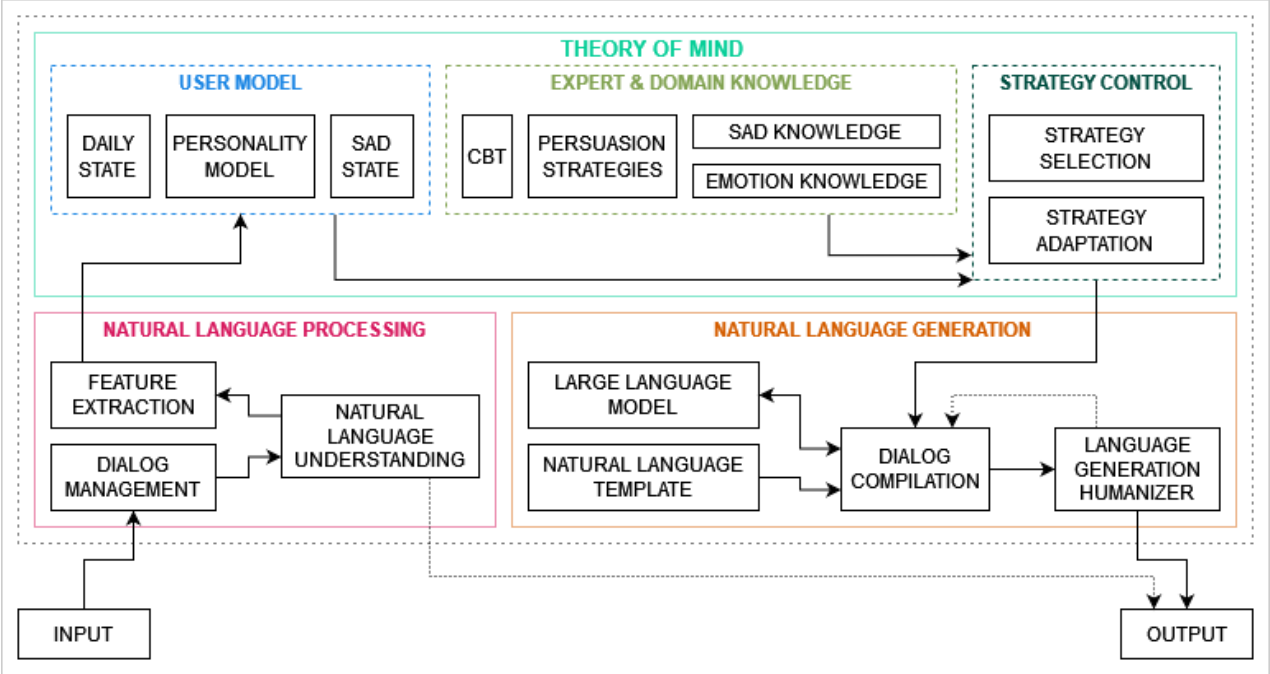


Figure 1: The system's cognitive architecture.

Theory of Mind: The central module, processing features to model the user and determine strategy. It includes:

- User Model: Simulates the user's state via submodules for Daily State (current multi-dimensional mental properties), Personality Model (long-term B5 traits), and SAD State (ML models for detecting current SAD levels/symptoms and forecasting them up to 7 days ahead from text).
- Expert & Domain Knowledge: Contains ontologies linking user states to interventions, including Cognitive Behavioral Therapy (CBT) techniques (difficulty-adapted), Persuasion Strategies (based on Cialdini's Principles mapped to B5 via a Domain Mapping Matrix (DMM)), SAD Knowledge (mapping issues to CBT via DMM), and Emotion Knowledge (guiding output tone) [4].
- Strategy Control: Selects and adapts strategies. Strategy Selection chooses appropriate CBT based on SAD state/topic and wraps it in a personalized persuasion strategy based on the User Model and DMM. Strategy Adaptation refines strategies based on effectiveness (using Ratio Formulas for learning) [4].
- Natural Language Generation (NLG): Compiles the final text output. It uses Natural Language Templates, enriches the text stochastically using a Large Language Model (LLM), and employs a Language Generation Humanizer (risk thresholding) to filter potentially harmful outputs before presenting the text to the user [4].

3 Evaluation and results

The system was evaluated through computational experiments and an empirical interventional study.

1.1 Computational experiments

Machine learning models within the SAD State submodule were trained and evaluated using 10-fold subject-wise cross-validation on the novel dataset (see Section 2).

Detection: SAD levels and 15 symptoms (inability to relax, nervousness, fear, tightness in chest, lightheadedness, feeling hot or cold, trembling, pounding heart, sadness, self-hatred, anhedonia, hopelessness, indecisiveness, fatigue, emotional detachment, suicidality) were detected from single text diary entries. Using kNN (chosen for explainability), accuracies ranged from 72.58% to 91.41%. This surpassed SOTA systems reviewed [4,6], which detected fewer categories, often with lower or non-comparable accuracy [4].

Forecasting: The system forecasted SAD levels and 15 symptoms 7 days ahead from single text entries, with kNN accuracies ranging from 71.54% to 87.68%. SOTA systems reviewed lacked this forecasting capability. Forecasting from 21 days of quantitative questionnaire data was also performed, achieving high accuracy (e.g., 93.14% for anxiety using Logistic Regression).

1.2 Empirical interventional study

A study involving 42 participants compared this work's system ("Our system") against Woebot [7] in a simulated daily check-in scenario. SAD levels were measured using Single Item Screening Questions (SISQs) before and after the interaction. User experience was measured using the User Experience Questionnaire (UEQ).

SAD Reduction: Paired t-tests showed "Our system" significantly reduced participant stress (M_diff = -0.263, p = 0.048) and anxiety (M_diff = -0.263, p = 0.040). Woebot showed no significant change in stress (p = 0.484) or

anxiety ($p = 0.509$). Neither system significantly changed depression levels ($p > 0.6$).

User Experience: "Our system" was rated significantly more supportive than Woebot (M_ours = 5.368 vs M_woebot = 4.261, $p = 0.041$). There was no significant difference in perceived novelty ($p = 0.084$). Qualitative feedback indicated users appreciated the depth of assessment in "Our system" but preferred Woebot's user interface and friendly personality.

3 Discussion and conclusion

The system's performance was benchmarked against relevant SOTA systems identified in the literature review [5], ChatGPT [8], and through empirical comparison. Computationally, the CogA demonstrated superior detection capabilities, identifying a broader range of SAD levels and specific symptoms (18 categories) from single text entries with high accuracy (up to 91.41% using kNN) compared to the fewer categories and often lower or non-comparable accuracies reported by reviewed systems and recent evaluations of ChatGPT. Crucially, the system introduced a novel 7-day forecasting capability from text data, achieving accuracies up to 87.68%, a capability absents in the reviewed SOTA systems. The empirical interventional study directly compared "Our system" to Woebot, a prominent SOTA agent. While Woebot excels in user interface and personality, "Our system," leveraging its ToM simulation, achieved statistically significant short-term reductions in user-reported stress and anxiety, which Woebot did not. Participants also perceived "Our system" as significantly more supportive, although they noted Woebot's superior interface. These comparative results highlight the advantages of the developed CogA in assessment depth, predictive power, and intervention effectiveness for specific negative states.

This thesis therefore successfully developed and evaluated a novel computational psychotherapy system based on simulating Theory of Mind, supported by a unique mixed-methods panel dataset. The computational results demonstrate SOTA performance in detecting a wide range of SAD states from text and introduce novel forecasting capabilities. The empirical study confirmed the hypothesis that simulating ToM can lead to a system that performs comparably or better than established SOTA systems like Woebot.

Limitations include potential dataset biases, reliance on LLMs' capabilities, and the quasi-experimental nature of the interventional study. Future work involves user interface development, refining ML models, expanding the dataset, conducting larger and longitudinal clinical evaluations, and comparing different LLMs within the CogA framework. Overall, the work demonstrates the potential of integrating AI, cognitive science, and behavioral science through ToM simulation to create more effective and personalized digital mental health interventions.

Acknowledgements

I acknowledge the support of my supervisors Prof. Dr. Matjaž Gams and Prof. Dr. Günter Schiepek, the Jožef Stefan Institute, the Slovenian Research and Innovation Agency (ARIS) (research core funding No. P2-0209 and Young researchers postgraduate research funding), and the study participants.

References

[1] T. Kolenik, M. Gams, "Persuasive technology for mental health: One step closer to (mental health care) equality?" IEEE Technology and Society Magazine, 40(1):80–86, 2021. DOI: 10.1109/MTS.2021.3056288

[2] T. Kolenik, M. Gams, "Intelligent cognitive assistants for attitude and behavior change support in mental health: State-of-the-art technical review," Electronics, 10(11):1250, 2021. DOI: 10.3390/electronics10111250

[3] A. M. Leslie, O. Friedman, T. P. German, "Core mechanisms in 'theory of mind'," Trends Cogn Sci, 8(12):528–533, 2004. DOI: 10.1016/j.tics.2004.10.001

[4] T. Kolenik, G. Schiepek, M. Gams, "Computational Psychotherapy System for Mental Health Prediction and Behavior Change with a Conversational Agent,", Neuropsychiatr Dis Treat, 20:2465-2498, 2024. DOI: 10.2147/NDT.S417695

[5] B. Rammstedt, O. P. John, "Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german," J Res Pers, 41(1):203–212, 2007. DOI: 10.1016/j.jrp.2006.02.001

[6] T. Kolenik. Methods in digital mental health: Smartphone-based assessment and intervention for stress, anxiety and depression. In Integrating Artificial Intelligence and IoT for Advanced Health Informatics, C. Comito, A. Forestiero, and E. Zumpano, Eds., Springer, 2021. DOI: 10.1007/978-3-030-91181-2_7

[7] K. K. Fitzpatrick, A. Darcy, M. Vierhile, "Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial," JMIR Ment Health, 4(2): e19, 2017. DOI: 10.2196/mental.7785

[8] B. Lamichhane, "Evaluation of chatgpt for nlp-based mental health applications," arXiv:2303.15727 [cs.CL], 2023. DOI: 10.48550/arXiv.2303.15727

A Privacy Based Deep Learning Algorithm for Big Data Analytics

D. Franklin Vinod*, Neha Ahlawat
Department of Computer Science and Engineering, Faculty of Engineering and Technology, SRM Institute of Science and Technology, NCR Campus, Delhi-NCR Campus, Delhi-Meerut Road, Modinagar, Ghaziabad, UP, India.
E-mail: datafranklin@gmail.com, nehablow@gmail.com
*Corresponding author

Thesis summary

Keywords: Big Data, Deep Belief Network, Heterogeneous data, pattern recognition, privacy preserving

Received: March 31, 2025

This thesis addresses critical challenges in privacy-preserving feature selection and classification for big data analytics. Specifically, four novel methodologies are proposed: Hierarchical Classification Feature Selection (HCFS), Privacy-Preserving Classification Selection with p-stability (PPCS), Local N-ternary Pattern combined with Modified Deep Belief Network (LNTP-MDBN), and Privacy-Preserving Cosine Similarity integrated with Multi-Manifold Deep Metric Learning (PPCS-MMDML). These approaches collectively enhance classification accuracy, optimize feature extraction from heterogeneous image sets, and robustly preserve privacy, demonstrating significant improvements in data-driven analytical applications.

Povztek: V disertaciji razbiti algoritem globokega učenja omogoča učinkovito klasifikacijo in izbiro značilk pri analizi velikih podatkov.

1 Introduction

The exponential growth of big data has significantly increased the risks of privacy breaches. Traditional processing systems face challenges managing the high volume, velocity, and variety of data generated by modern technologies such as sensors and the Internet of Things (IoT). Distributed architectures like Hadoop facilitate efficient big data management; however, handling sensitive data, particularly in sectors such as healthcare, necessitates stringent privacy and security measures. This research focuses on innovative deep learning techniques that balance effective data utilization with essential privacy protection. Four advanced methodologies—HCFS, PPCS with p-stability, LNTP-MDBN, and PPCS-MMDML—are introduced, targeting privacy-preserving feature selection and efficient classification, especially in complex image datasets [1-3].

The thesis summery highlights the significance of big data analytics in the modern digital era., emphasizing the need for classification in big data environments, the role of deep learning in big data classification, and the privacy challenges in big data analytics.

2 Methodology and designs

The thesis introduces four distinct privacy-preserving methods:

HCFS (Hierarchical Classification Feature Selection) enhances classification by identifying optimal feature subsets, significantly improving accuracy [4].

PPCS with p-stability is a bi-level approach safeguarding individual and network-level privacy, effectively preventing intrusions and preserving data integrity [5].

LNTP-MDBN (Local N-ternary Pattern with Modified Deep Belief Network) specializes in feature extraction from heterogeneous images, minimizing reconstruction errors and maximizing classification accuracy through strategic modifications to deep belief networks [6].

PPCS-MMDML (Privacy-Preserving Cosine Similarity with Multi-Manifold Deep Metric Learning) addresses privacy-preserving classification challenges specifically in cancer image analysis, maintaining high classification accuracy while enforcing stringent privacy constraints [7].

3 Results and discussion

The OSIRIX viewer [8] and the Mammographic Image Analysis Society (MIAS) [9] are used to build datasets for the brain, breast, and bone in order to support the suggested performance. Every image used has a 1024 x 1024-pixel dimension. Figure 1 illustrates the metrics used to validate the performance of the proposed LNTP-MDBN, including classification accuracy, macro-averaged F1 score, and running time.

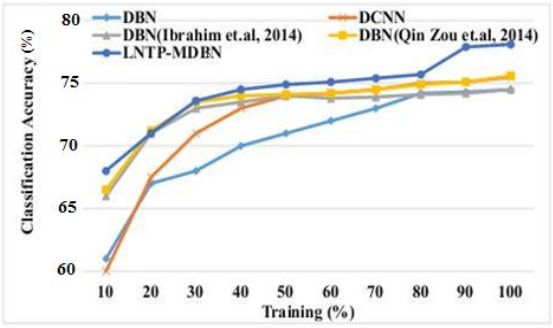


Figure 1: Classification accuracy analysis

With 100% training, the DBN and LNTP-MDBN have respective accuracy percentages of 75.6 and 78.1. The comparison analysis demonstrates that the suggested LNTP-MDBN work is superior to the current DBN.

Because of enhanced pattern extraction and the redesigned DBN that carries out heterogeneous image set classification, the suggested LNTP-MDBN uses less computing time for each level. LNTP spends 0.095, 1.25, and 0.023 seconds on feature extraction, training, and testing levels, respectively.

Table 1: Classification accuracy analysis of PPCS-MMDML

S.No	Data sets	Classification Accuracy (%)	
		MMDML	PPCS-MMDML
1	Bone Cancer	66.5	74.2
2	Brain Cancer	68.5	72.6
3	Breast Cancer	71.6	77.8

The performance assessments of the proposed PPCS-MMDML and the existing MMDML in three datasets related to cancer diseases are shown in Table 1. The PPCS-MMDML accuracy rates for breast, brain, and bone cancer diseases are 77.8%, 74.2, and 72.6, respectively. According to the comparison analysis, the suggested PPCS-MMDML improves the categorization of the bone, brain, and breast datasets by 7.7%, 4.1%, and 6.2%, respectively, over the current MMDML.

4 Conclusion and future work

The proposed methods, particularly LNTP-MDBN and PPCS with p-stability, achieve superior results in privacy preservation and classification accuracy within big data environments. Future research directions include integrating PPCS and LNTP-MDBN for enhanced privacy-preserving deep learning frameworks, exploring online learning techniques for real-time classification, and expanding application. The comparison analysis proves that the proposed work LNTP-MDBN provides improvement over the existent DBN which is suitable for heterogeneous cancer disease detection.

References

[1] W. Dou, X. Zhang, J. Liu and J. Chen, Hiresome-II: Towards privacy aware cross-cloud service composition for big data applications, IEEE Trans Parallel Distrib Syst., 6(2), (2014), 455–466.

[2] A.T. Azar and A.E. Hassanien, Dimensionality reduction of medical big data using neural-fuzzy classifier, Soft Computing, 19, (2015), 1115-1127.

[3] S. Gao, Z. Zeng, K. Jia, T.-H. Chan and J. Tang, Patch-Set-Based Representation for Alignment-Free Image Set Classification, IEEE Trans.Cir. and Sys. for Video Tech., 26, (2016), 1646-1658.

[4] D. F. Vinod and V. Vasudevan, "A filter-based feature set selection approach for big data classification of patient records," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, India, 2016, pp. 3684-3687, doi: 10.1109/ICEEOT.2016.7755397.

[5] Franklin Vinod, D., Vasudevan, V. (2017). A Bi-level Security Mechanism for Efficient Protection on Graphs in Online Social Network. In: Arumugam, S., Bagga, J., Beineke, L., Panda, B. (eds) Theoretical Computer Science and Discrete Mathematics. ICTCSDM 2016. Lecture Notes in Computer Science, vol 10398. Springer, Cham.

[6] Vinod DF, Vasudevan V. LNTP-MDBN: Big Data Integrated Learning Framework for Heterogeneous Image Set Classification. Curr Med Imaging Rev. 2019;15(2):227-236. doi: 10.2174/1573405613666170721103949. PMID: 31975670

[7] Franklin Vinod, D., Vasudevan, V. (2019). PPCS-MMDML: Integrated Privacy-Based Approach for Big Data Heterogeneous Image Set Classification. In: Satapathy, S., Joshi, A. (eds) Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 106. Springer, Singapore.

[8] OSIRIX. Available: http://www.osirix-viewer.com/resources/ dicom image-library

[9] J. Suckling, J. Parker, D. Dance, S. Astley, I. Hutt, C. Boggis and I. Ricketts, Mammographic Image Analysis Society(MIAS) database v1.21 [Dataset], (2015).

[10] R. Ibrahim, N. A. Yousri, M. A. Ismail and N. M. El-Makky, Multi-level gene/MiRNA feature selection using deep belief nets and active learning, In Proc Int. Conf. IEEE Engg. Med. Bio Society (EMBC), (2014), 3957-3960.

[11] Q. Zou, Y. Cao, Q. Li, C. Huang and S. Wang, Chronological classification of ancient paintings using appearance and shape features, Pattern Recognition Letters, 49, (2014), 146-154.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several sci- entific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute tem- perature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, en- ergy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research de- partments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the uni- versities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; ap- plied mathematics. Most of the activities are more or less closely connected to information sciences, in particu- lar computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automa- tion and control, professional electronics, digital communi- cations and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the in dependent state of S**love**nia (or S♡nia). The capital today isconsidered a crossroad bet between East, West and Mediter-ranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology Park “Ljubljana” has been proposed as part of the national strat- egy for technological development to foster synergies be- tween research and industry, to promote joint ventures be- tween university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology Park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.: +386 1 4773 900, Fax.: +386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2022 (Volume 46) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Blaž Mahnič, Gašper Slapničar; gasper.slapnicar@ijs.si
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Slavko Žitnik) in cooperation with the following societies (and contact persons):
Slovene Society for Pattern Recognition (Matej Kristan)
Slovenian Artificial Intelligence Society (Aleksander Sadikov)
Cognitive Science Society (Toma Strle)
Slovenian Society of Mathematicians, Physicists and Astronomers (Mojca Vilfan)
Automatic Control Society of Slovenia (Giovanni Godena)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Matjaž Mikoš) ACM Slovenia (Ljupčo Todorovski)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 49 Number 2 May 2025 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **49** (2025) Number 2, pp. 223–458