Volume 50 Number 2 July 2026 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.

Executive Editor – Editor in Chief
Matjaž Gams
Jožef Stefan Institute Jamova 39, 1000
Ljubljana, Slovenia
Phone: +386 1 4773 900
editor-in chief@informatica.si
http://dis.ijs.si/mezi

Editor Emeritus
Anton P. Železnikar
Volaričeva 8, Ljubljana, Slovenia
s51em@lea.hamradio.si

Executive Associate Editor - Technical Editor
Drago Torkar
Jožef Stefan Institute Jamova 39, 1000
Ljubljana, Slovenia
Phone: +386 1 4773 900
technical-editor@informatica.si

Executive Associate Editor - Deputy Technical Editor
Tine Kolenik
Paracelsus Medical University, Salzburg
fast-track-editor@informatica.si

Production Editors
Gašper Slapničar and Blaž Mahnič
Jožef Stefan Institute Jamova 39, 1000
Ljubljana, Slovenia

Editorial Board
Juan Carlos Augusto (Argentina)
Vladimir Batagelj (Slovenia)
Francesco Bergadano (Italy)
Marco Botta (Italy)
Pavel Brazdil (Portugal)
Andrej Brodnik (Slovenia)
Ivan Bruha (Canada)
Wray Buntine (Finland)
Zhihua Cui (China)
Aleksander Denisiuk (Poland)
Hubert L. Dreyfus (USA)
Jozo Dujmović (USA)
Johann Eder (Austria)
George Eleftherakis (Greece)
Ling Feng (China)
Vladimir A. Fomichov (Russia)
Maria Ganzha (Poland)
Sumit Goyal (India)
Marjan Gušev (Macedonia)
N. Jaisankar (India)
Dariusz Jacek Jakóbczak (Poland)
Dimitris Kanellopoulos (Greece)
Dimitris Karagiannis (Austria)
Samee Ullah Khan (USA)
Hiroaki Kitano (Japan)
Igor Kononenko (Slovenia)
Miroslav Kubat (USA)
Ante Lauc (Croatia)
Jadran Lenarčič (Slovenia)
Shiguo Lian (China)
Suzana Loskovska (Macedonia)
Ramon L. de Mantaras (Spain)
Natividad Martínez Madrid (Germany)
Sanda Martinčić Ipšić (Croatia)
Angelo Montanari (Italy)
Pavol Návrat (Slovakia)
Jerzy R. Nawrocki (Poland)
Nadia Nedjah (Brasil)
Franc Novak (Slovenia)
Marcin Paprzycki (USA/Poland)
Wiesław Pawłowski (Poland)
Ivana Podnar Žarko (Croatia)
Karl H. Pribram (USA)
Luc De Raedt (Belgium)
Shahram Rahimi (USA)
Dejan Raković (Serbia)
Jean Ramaekers (Belgium)
Wilhelm Rossak (Germany)
Ivan Rozman (Slovenia)
Sugata Sanyal (India)
Walter Schempp (Germany)
Johannes Schwinn (Germany)
Zhongzhi Shi (China)
Oliviero Stock (Italy)
Robert Trappl (Austria)
Terry Winograd (USA)
Stefan Wrobel (Germany)
Konrad Wrona (France)
Xindong Wu (USA)
Yudong Zhang (China)
Rushan Ziatdinov (Russia & Turkey)
Slavko Žitnik (Slovenia)
Ylber Januzaj (Kosovo)

Honorary Editors
Hubert L. Dreyfus† (1929-2017 USA)

Forecasting Rectangular Pocket Deviations in Birch Plywood for Milling Process Optimization

Primož Kocuvan[1], Rok Struna[2], Vinko Longar[2]
[1]Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[2]Rudolfovo – znanstveno in tehnološko središče Novo mesto, Podbreznik 15, 8000 Novo mesto, Slovenia
E-mail: primoz.kocuvan@ijs.si, rok.struna@rudolfovo.eu, vinko.longar@rudolfovo.eu

Keywords: milling process, regression, rectangular pocket deviation, optimization, augmentation

Received: July 4, 2026

We explored the field of predicting and minimizing dimensional deviations when milling rectangular pockets in birch plywood, with the aim of improving the accuracy of manufacturing processes. We examined various machining conditions altering feed rate, depth of cut, and spindle speed and acquired high-resolution 3D scans to quantify pocket-shape deviations against the prescribed tolerances. To enhance the model's robustness, the empirical dataset was augmented via synthetic data-generation techniques. We then compared several approaches using cross-validation. MLP proved the most accurate. These findings demonstrate the utility of readily available machine-learning algorithms for precise deviation prediction and lay the groundwork for future integration of automated hyperparameter tuning and feature-based process optimization.

Povzetek: Raziskava je pokazala, da je mogoče z metodami strojnega učenja, zlasti z modelom MLP, učinkovito napovedovati in zmanjševati dimenzijska odstopanja pri rezkanju vezanega lesa.

1 Introduction

Dimensional accuracy is a central quality requirement in CNC machining of wood-based panels, particularly in furniture, interior products, and modular assemblies where milled pockets, slots, and mating features must fit reliably without excessive post-processing. In plywood and other engineered wood materials, dimensional deviations may arise from the combined effects of cutting parameters, tool deflection, local material heterogeneity, layer orientation, moisture-related variability, and machine dynamics. These deviations are especially relevant in pocket milling, where small errors in edge position or pocket geometry can affect assembly precision, visual quality, and repeatability in batch production. Recent studies on CNC machining of engineered wood panels have shown that parameters such as depth of cut and feed rate can significantly influence dimensional accuracy and surface quality, confirming the need for systematic parameter selection and process control. Although the effects of machining parameters on surface roughness and dimensional accuracy have been studied for several wood-based materials, much of the available work relies on conventional experimental comparison or statistical analysis. Data-driven prediction of local dimensional deviations in plywood pocket milling remains less explored, particularly when high-resolution 3D scanning is used to quantify the machined geometry and when limited experimental datasets are expanded using data-augmentation techniques. This creates a practical research opportunity: if dimensional deviations can be predicted from machining parameters before or during production, CNC settings can be adjusted more systematically to improve accuracy, reduce trial-and-error tuning, and support automated process optimization.

In the DIGITOP project, this problem was investigated through the fabrication of a custom-engineered plywood substrate intended for the assembly of a headphone stand prototype. The substrate consists of a single panel of commercial-grade birch plywood, 10 mm in thickness, with overall dimensions of 100 × 76 cm. Birch was selected on account of its high specific strength and stiffness relative to softwood alternatives, as well as its dimensional stability under controlled indoor conditions; however, its susceptibility to moisture and ultraviolet degradation limits its suitability for long-term exterior applications [1]. Alternative hardwood or engineered wood composites may be substituted without fundamentally altering the downstream processing methodology.

Prior to milling, the full-size panel was subdivided into smaller tiles by CO_2 laser cutting. The laser parameters employed were a traverse speed of 3 $mm \cdot s^{-1}$, 95% of a nominal 130 W output power, and an air-assist pressure

of 1.5 bar to optimize kerf quality and minimize thermal char. These settings were determined experimentally to achieve clean, burr-free edges while maintaining acceptable processing throughput. In the subsequent milling process, rectangular pockets were produced in the plywood to form the functional geometry of the headphone stand. A 6 mm spiral cutter made of HWM material, i.e., solid tungsten carbide, was used. This cutter was selected for its high wear resistance and stiffness, which minimize tool deflection and contribute to tighter pocket-edge tolerances; preliminary trials showed that softer cutter materials increased radial run-out by up to 15 µm under identical feed conditions.

This paper makes the following contributions:

• A systematic dataset of 17 real and 483 SMOGN-augmented observations linking feed rate, depth of cut, and spindle speed to rectangular-pocket deviation in birch plywood.

• A comparative evaluation of four off-the-shelf regression algorithms using 5-fold cross-validation, identifying MLP as the top-performing model, with R^2 = 0.9316.

Compared with existing ML-based machining prediction studies, which most commonly address surface roughness, tool wear, chatter, energy consumption, or dimensional accuracy in metal machining, this work focuses specifically on local dimensional and surface-topography deviations in CNC pocket milling of birch plywood. The novelty of the study lies in combining high-resolution 3D-scanner-derived deviation metrics with a small-data SMOGN augmentation workflow to model a wood-specific machining problem that has received limited attention in the existing ML machining literature.

The study contributes to the state of the art by combining 3D-scanner-based evaluation of plywood surface topography with standard machine learning regression techniques to predict the local-scale variability of dimensional accuracy, expressed through its standard deviation. To the best of the authors' knowledge, the specific combination of the applied 3D scanning setup, plywood material, machined geometry, and data-augmentation pipeline has not been previously reported in this context. The developed regression model provides a basis for automated process optimization by supporting the adjustment of feed rate, depth of cut, and spindle speed to achieve improved surface quality.

2 Related work

Birch lumber possesses superior mechanical performance compared to softwood species. With the cross-laminated veneers configuration, birch plywood is considered promising for structural use. Due to its architecture and attendant enhancement of stiffness, strength, and dimensional stability, birch plywood represents a highly promising candidate for load-bearing structural applications [2],[3]. The similar process of milling a rectangular pocket is boring. In the course of bore machining under finishing conditions, the parameters that determine the quality of the surface layer represented by the roughness and quality grade are particularly important. The roughness of the treated surface is mainly affected by the vibrations of the manufacturing system [4]. The boring operation may be performed on a variety of machine-tool platforms, including general-purpose or universal equipment such as lathe (turning) centers and milling machines.

Sutçu and Karagöz evaluated how spindle speed, feed rate, stepover and depth of cut influence the surface topography of pocket-milled MDF using a CNC router. Employing roughness average (Ra), mean peak-to-valley height (Rz) and root-mean-square height (Rq) as quality metrics, they found that raising the spindle speed improves the finish (lower Ra, Rz, Rq), whereas increasing the feed rate, stepover or depth of cut leads to a rougher surface [5].

In a recent investigation of CNC-router machining, a two-factor factorial design was employed to assess how spindle speed, feed rate, cutting depth and wood species affect surface-finish quality. The authors report that the wood's intrinsic structure exerts a significant influence on the machined surface, with higher spindle speeds yielding smoother finishes, whereas increased feed rates produce noticeably rougher textures [6].

İşleyen and Karamanoğlu (2019) conducted a factorial study varying spindle radial speed (18 000 min^{-1} vs. 24 000 min^{-1}), feed rate (2 500–10 000 mm/min), tool diameter (4 vs. 6 mm), and depth of cut (4 vs. 6 mm) to evaluate their effects on Ra and Rz surface-roughness parameters of 18 mm MDF using a stylus profilometer (Handysurf E-35) with ten measurements per sample [7].

José and Mathew (2024) propose a comprehensive IIoT-based platform that captures and unifies real-time data from traditional control systems (PLCs, DCSs), mobile devices, and ERP systems, leveraging edge processing alongside cloud-based machine learning and AI to compute and visualize key performance indicators (KPIs). The article has similar objective to ours, because they both apply data-driven, predictive-analytics strategies to improve manufacturing accuracy and efficiency [8].

Feng et al. (2024) and similarly Hwang (2026) reviewed machine-learning applications in wood materials, including property prediction, defect detection, health monitoring, and material-design optimization. They showed that ML can reduce experimental effort by modeling complex links between wood structure, processing, and performance. However, they also identified limited data availability, data quality, and model interpretability as key challenges. Since dimensional accuracy prediction in wood machining was not specifically addressed, this remains a relevant research gap for CNC wood manufacturing [9], [10].

Purwanto et al. (2026) investigated CNC nesting of High Moisture Resistance (HMR) panels, focusing on the effects of depth of cut and feed rate on dimensional accuracy and surface roughness. They found that depth of cut significantly affected dimensional accuracy, while feed rate mainly influenced surface roughness. The best machining condition was obtained at a 2 mm depth of cut and 33 mm/s feed rate. However, the study used experimental and ANOVA-based analysis rather than ML-based prediction, leaving room for data-driven models to predict dimensional deviations in CNC wood machining [11].

3 Experiment

We implemented a controlled laboratory study to predict rectangular pocket deviation as a function of key process variables. Our goal was to identify the combination of settings that minimizes deviation quantified here by the forecasted standard deviation of rectangular pocket diameter, which serves as a proxy for machining error. To that end, we supplied our predictive models with three principal factors known to affect surface roughness and structural integrity within a milled rectangular pocket:

- Feed (move) speed (mm/min). Governs the axial advancement rate of the cutting tool relative to the workpiece, directly influencing heat generation and chip formation.
- Milling depth of spiral cutter (mm). Defines the radial engagement of the cutter per pass; deeper cuts typically increase cutting forces and deflection.
- Spindle radial speed (% of maximum, where 100 % corresponds to 24 000 min^{-1}). Controls the circumferential velocity of the tool, affecting surface finish and tool–workpiece interaction dynamics. Here we can also set the radial speed to over 100 %. At some measurements we set it to 120 %.

A 6 mm-diameter spiral cutter (Figure 1 [12]) was used in conjunction with an electrical spindle (Hiteco; Figure 1 [13]). Birch specimens were digitized with a Raptor 3DX Standard (5 MP) structured-light scanner. Scans were executed in automatic mode on the patented 2 + 1-axis robotic cradle (360° rotation, 90° arm swing, ±45° sensor tilt). Using the FOV 330 optics (270 × 200 × 200 mm, point spacing 0.16 mm), a single specimen was captured from ten poses. Raw point clouds were auto-registered in 3DX Scan and exported as STL meshes [14]. The Raptor 3DX system provides a fully automated solution for 3D metrology encompassing quality control and reverse-engineering workflows. Figure 2 presents a representative 3D scan of a birch plywood specimen. Surface deviations within a ±0.1 mm tolerance band are rendered in green; yellow indicates slight positive deviations, red denotes larger positive deviations beyond tolerance, and blue corresponds to negative deviations. The adjacent legend quantifies the deviation thresholds associated with each color.

Figure 1: Spiral cutter 6mm CMT 192.860.11 (right) and Electrical spindle Hiteco S-13/18 24 ER25 DX BT (left)

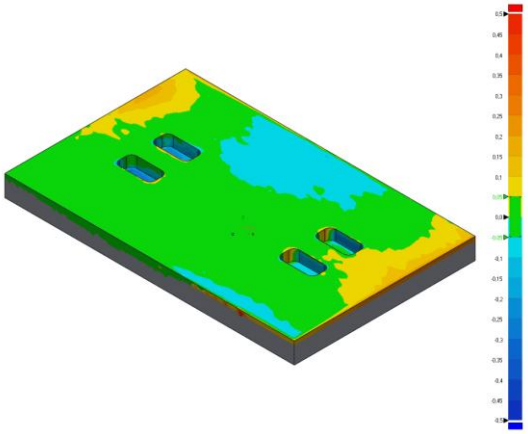


Figure 2: Headphone stand made out of Birch plywood - 3D surface (colormap)

Seventeen birch-plywood specimens were prepared, each characterized by a distinct combination of feed (move) speed, milling depth and tool spindle speed. The environment variables are not applicable. Because this sample size was insufficient for robust regression modeling, we augmented the dataset by generating synthetic observations using the SMOGN (Synthetic Minority Over-Sampling Technique for Regression with Gaussian Noise) library in Python [15]. Table 1 summarizes the SMOGN parameter settings. These values were selected to minimize estimation error while promoting a greater dispersion of the synthesized

samples. These SMOGN settings were selected based on a preliminary sensitivity analysis, which showed that a relevance cutoff of 0.90 and noise level of 0.30 best balanced synthetic-data diversity against fidelity to real measurements.

The quality of the synthetically generated records was assessed against the critical process parameter spindle radial speed. Experimental machining trials showed that cutter fracture occurs whenever spindle radial speed falls below 34 %. Accordingly, the data-augmentation pipeline was constrained so that no synthetic sample violated this lower bound, and all subsequent evaluations were performed exclusively within the safe operating range (≥ 34 %). This approach ensures that the augmented dataset reflects only feasible machining conditions and does not introduce unrealistic failure cases that would never be allowed in practice.

The first parameter means each seed point can be interpolated with more distant neighbors, broadening the convex hull in which new points lie hence more dispersed synthetic samples. The second parameter specifies the magnitude of Gaussian jitter added to interpolated points (in our case 30% of the feature range). The third parameter sets the relevance cutoff $\varphi(y) \geq 0.90$ for defining "rare" (to be oversampled) vs. "normal" (to be under-sampled) observations. The fourth chooses between two sampling schemes "balance" or "extreme".

Table 1: Most important parameters for SMOGN library configuration

Name of the parameter	Configured value
Number of nearest neighbors to use when generating synthetic points via interpolation (default is 5).	15
Proportion of Gaussian noise (relative to each feature's standard deviation)	0.30
Relevance cutoff (0–1) defining which target values are considered "rare" and thus eligible for oversampling.	0.90
Strategy for selecting which rare cases to oversample here `"extreme"` focuses on the most extreme targets.	"extreme"

In total, 483 synthetic data points (other 17 a real data points) were produced, a quantity chosen empirically.

4 Evaluation and results

We evaluated four regression algorithms multilayer perceptron regression, linear regression, Bayesian ridge regression, and support vector regression using the scikit-learn Python library (latest stable release version 1.9.0 [16]) with different hyperparameter settings. To quantify generalization performance, we performed 5-fold cross-validation by partitioning the dataset into five equal folds and iteratively training each model on four folds and validating on the remaining fold. For each fold we computed the coefficient of determination (R^2), defined as (Equation 1) and the mean squared error (MSE) (Equation 2), and then averaged these metrics across folds; the aggregated results are presented in Table 2. An R^2 value closer to one indicates better predictive accuracy, whereas a negative R^2 implies that a model performs worse than a baseline predictor that always outputs the mean of the target variable. According to the results in Table 2, the MLP regressor model attained the highest mean coefficient of determination (R^2) and the lowest mean squared error (MSE). For hyperparameter optimization we applied GridSearchCV using the default parameter grid provided. The tuned models exhibited a mean increase in the coefficient of determination (R^2) of just 0.003, an improvement that is practically negligible. Table 3 presents an additional sanity-check experiment conducted exclusively on the 17 real experimental specimens without cross-validation. In this sanity-check experiment, the models were trained on the SMOGN-augmented dataset and evaluated on the 17 original measured specimens, which were not used during training. The purpose of this analysis was to assess whether the predictive behavior observed with the SMOGN-augmented dataset is also reflected when only original measured samples are considered. The strongly negative R^2 obtained by the SVR model suggests that its kernel-based representation did not match the structure of this small and augmented dataset, leading to predictions that were less accurate than simply using the mean response value.

Table 2: ML models and results on mixed synthetic and real data

Model	Mean MSE	Mean R^2
MLP Regressor	0.000193	0.931626
Bayesian Ridge	0.000332	0.881769
Linear Regression	0.000332	0.871411
SVR	0.003192	-0.135114

Table 3: ML models and results on real data as a test set (17 specimens)

Model	Mean MSE	Mean R^2
MLP Regressor	0.000072	0.929800
Bayesian Ridge	0.000278	0.767827
Linear Regression	0.000277	0.767573
SVR	0.004445	-2.712226

(1) $R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}$

(2) $MSE = \frac{1}{n}\sum (y_i - \hat{y}_i)^2,$

where $\hat{y}, \bar{y}$ are predicted values and sample mean of the observation, respectively.

Because the process of 3D scanning the birch plywood specimens is slow (each scan takes about 20 min), we scanned only 17 specimens and augmented additionally 483 specimens.

5 Discussion

Our results demonstrate that classical supervised ML techniques can predict the dimensional error incurred during rectangular-pocket milling with low numerical error, while also indicating that there remains scope for further improvement. Specifically, the evaluated models produced mean squared error (MSE) values of very low magnitude, on the order of 0.0001 in the squared-millimeter domain, reflecting the intrinsically small variability of the milling error within the investigated experimental space. This low MSE corresponds to minor deviations in the predicted dimensional error and confirms that both the predictive discrepancies and the actual process deviations were relatively small. However, these performance values should be interpreted with caution. Real and synthetic samples were pooled and randomly shuffled before cross-validation. The folds were generated from the combined dataset, so the assignment of samples to folds was not based on whether samples were real or synthetic. This procedure avoided a systematic separation of real and synthetic samples that could bias or inflate the performance metrics.

Nevertheless, the comparison between model performance on synthetic and real examples suggests that the reported metrics may still be somewhat optimistic. The models generally performed better on the SMOGN-augmented synthetic samples than on the original experimental samples, indicating that the synthetic data may represent a smoother or more regular approximation of the process than the real measurements. Although SMOGN is useful for addressing data imbalance in small experimental datasets, the generated samples are primarily obtained by interpolation within the observed feature space and therefore may not fully capture the complex non-linear physical behavior of the milling process. This is a known limitation in small-data experimental settings, where the available measurements cover only a narrow range of process parameters and where synthetic augmentation cannot replace additional physical experiments.

Despite these limitations, the results provide evidence that ML-based regression models can support the prediction of dimensional deviations in rectangular-pocket milling. The study is limited by the narrow parameter ranges and by the omission of factors such as tool wear, plywood moisture content, and ambient conditions. Future research will extend the dataset to include these variables, explore physics-informed feature engineering, and integrate real-time deviation forecasting into CNC-controller feedback loops for adaptive milling. In a production setting, whether in furniture, cabinetry, or prototype fabrication, integrating such regression models could allow operators to adjust feed rate, depth of cut, and spindle speed to remain within tight tolerance bands, thereby reducing scrap and rework. The approach may also support predictive maintenance by identifying parameter regimes that accelerate tool wear and could be embedded into real-time control loops for adaptive compensation of geometric errors. Beyond cost and time savings, these capabilities may enhance quality consistency across batches of birch plywood components and contribute to the development of more automated, high-precision woodworking workflows. From a practical industrial perspective, these findings indicate that ML-based prediction can support more informed selection of CNC parameters before machining, reducing reliance on trial-and-error setup procedures.

Acknowledgements

The research was supported by DIGITOP project which is funded by Ministry of Higher Education, Science and Innovation of Slovenia, Slovenian Research and Innovation Agency, and EU NextGenerationEU under Grant TN-06-0106. We thank prof. dr. Matjaž Gams for proof-reading the article and mentorship support within DIGITOP project.

References

[1] Wang, Yue & Wang, Tianxiang & Crocetti, Roberto & Wålinder, Magnus. (2023). Effect of moisture on the edgewise flexural properties of acetylated and unmodified birch plywood: a comparison of strength, stiffness and brittleness properties. European Journal of Wood and Wood Products. 82. 10.1007/s00107-023-02014-6.

[2] Tianxiang Wang, Yue Wang, Roberto Crocetti, Magnus Wålinder, In-plane mechanical properties

of birch plywood, Construction and Building Materials, Volume 340, 2022, 127852, ISSN 0950-0618, https://doi.org/10.1016/j.conbuildmat.2022.127852.

[3] Yue Wang, Tianxiang Wang, Roberto Crocetti, Michael Schweigler, Magnus Wålinder, Embedment behavior of dowel-type fasteners in birch plywood: Influence of load-to-face grain angle, test set-up, fastener diameter, and acetylation, Construction and Building Materials, Volume 384, 2023, 131440, ISSN 0950-0618, https://doi.org/10.1016/j.conbuildmat.2023.131440.

[4] Ignatov, A.V., Tagil'tsev, S.V. & Namazova, A.I. Bore making using vibration-resistant boring arbors assembled with the use of adhesive materials. *Polym. Sci. Ser. D* **10**, 106–110 (2017). https://doi.org/10.1134/S1995421217020071

[5] Sütcü, A.; Karagöz, Ü. Effect of Machining Parameters on Surface Quality after Face Milling of MDF. *Wood Res.* 2012, 57, 231–240.

[6] Koç, K.H.; Erdinler, E.S.; Hazır, E.; Öztürk, E. Effect of CNC Application Parameters on Wooden Surface Quality. In *Proceedings of the 58th International Convention of Society of Wood Science and Technology*, Grand Teton National Park, Jackson, WY, USA, 7–12 June 2015; pp. 1–8.

[7] İşleyen, Ü.K.; Karamanoğlu, M. The Influence of Machining Parameters on Surface Roughness of MDF in Milling Operation. BioResources 2019, 14, 3266–3277.

[8] Jeeva Jose, Vijo Mathew (2024). Internet of Things – A Model for Data Analytics of KPI Platform in Continuous Process Industry. *Informatica*, 48(1), 119–130. https://doi.org/10.31449/inf.v48i1.3826

[9] Feng, Y., Mekhilef, S., Hui, D., Chow, C. L., & Lau, D. (2024). Machine learning-assisted wood materials: Applications and future prospects. Extreme Mechanics Letters, 71, 102209. https://doi.org/10.1016/j.eml.2024.102209

[10] S.-W. Hwang, "Machine learning-driven research in wood science: from prediction to understanding through the framework of Wood Informatics," *Journal of Wood Science*, vol. 72, article no. 7, 2026, doi: 10.1186/s10086-026-02258-9.

[11] Purwanto, A. A., Amarta, Z., Rahmat, B., Widiyanto, W., & Fahrudin, M. (2026). Effects of depth of cut and feed rate on dimensional accuracy and surface roughness in CNC nesting of HMR panels. Jurnal Polimesin, 24(2), 368–372

[12] https://www.toolnation.com/cmt-19286011-spiral-cutter-6mm-shank-8mm.html, Accessed 20.06.2025

[13] https://www.hiteco.net/en/products/electrospindle-m.t.c.c8247, Accessed 20.06.2025

[14] https://3d-ing.si/wp-content/uploads/2019/03/Raptor3DX_EN_Brochure_181119_low.pdf, Accessed 20.06.2025

[15] https://pypi.org/project/smogn/, Accessed 20.06.2025

[16] https://scikit-learn.org/stable/, Accessed 20.06.2025

Electrical Current Signature-Based Machine Learning Models for Streetlight Fault Prediction in Smart City Infrastructure

Pallav Dutta[1*], Masuma Parvin[1], Rumpa Saha[1], Suddhasatwa Chakraborty[2]
[1]Department of Electrical Engineering, Aliah University, Kolkata, India
[2]Illumination Engineering Laboratory, Department of Electrical Engineering, Jadavpur University, Kolkata, India
E-mail: pallav.dutta@aol.com , masumakhan0001@gmail.com , rumpasaha.au@gmail.com, suddhasatwachakraborty@gmail.com
[*]Corresponding author

Keywords: Machine learning, fault prediction, street light maintenance, predictive modeling, urban infrastructure, smart city

Received: April 21, 2025

Urban street lighting systems are critical infrastructures, yet their maintenance is predominantly reactive, inefficient, and costly. The paper introduces a proactive, data-driven framework using machine learning (ML) to predict incipient streetlight faults by analyzing electrical current signatures. This methodology, which leverages features like current fluctuations and voltage variations as early indicators of failure, is significantly underexplored in smart lighting infrastructure. We utilize a public Kaggle dataset encompassing operational parameters and annotated fault types. To overcome the absence of high-resolution current data, we augmented the dataset with simulated current signature features (e.g., mean current, variance, power factor) engineered to reflect typical electrical behaviors under various fault conditions. Recognizing that fault data is inherently imbalanced, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to the training set to prevent model bias and improve the detection of rare fault classes. A comprehensive comparative analysis of eight supervised ML models; Decision Tree, Random Forest, Logistic Regression, SVM, K-Nearest Neighbors, XGBoost, Naive Bayes and AdaBoost, was then conducted to identify the most robust approach. The results, validated via 5-fold cross-validation, show that ensemble methods are superior in capturing the data's complex, non-linear patterns. Random Forest achieved the highest classification accuracy (84%), with strong supporting metrics (Precision: 0.87, Recall: 0.85). Notably, AdaBoost demonstrated the best discriminative ability (AUC = 0.79), highlighting a practical trade-off between overall accuracy and threshold-tuning flexibility. A feature importance analysis confirmed that electrical signature features were among the most influential predictors. This research validates electrical current signature analysis as a viable methodology for fault prediction, providing a framework to shift maintenance from a reactive to a proactive strategy. We acknowledge the limitation of using simulated data and position this work as a foundational proof-of-concept. The study provides an empirically validated baseline for future intelligent maintenance systems, with the clear next step being validation on field-collected, non-simulated sensor data.

Povzetek: Članek predstavi uporabo strojnega učenja za napovedovanje okvar javne razsvetljave in podporo bolj proaktivnemu vzdrževanju.

1 Introduction

Public street lighting is crucial for safety, security and urban efficiency. Failures can lead to chaos, safety hazards and financial losses [1], [2]. It provides illumination in public spaces at night, impacting urban life by enhancing visibility, improving traffic safety and supporting local commerce and tourism [3]. Streetlights also deter crime and facilitate surveillance, making them a vital part of urban infrastructure. However, they can fail due to environmental conditions, loose connections and voltage fluctuations, highlighting the need for effective maintenance [4], [5]. Traditional streetlight maintenance relies on reactive methods, addressing faults only after they occur, leading to prolonged downtime and increased costs. With the growing role of technology, there is a rising demand for intelligent systems that predict failures and optimize resources [6].

Machine learning is increasingly being explored for streetlight fault detection, particularly in urban areas. By leveraging historical data and real-time monitoring, predictive models can detect anomalies, allowing authorities to take preventive action before outages occur [7], [8]. This study analyzes machine learning techniques for streetlight failure detection, comparing models such as Decision Tree, Random Forest, Logistic Regression,

Support Vector Machine, K-Nearest Neighbors, XGBoost, AdaBoost and Naïve Bayes to identify the most effective approach [9]. Using a real-world dataset, the study evaluates how these models detect failures arising from bulb defects, electrical malfunctions and environmental anomalies.

A key focus is on electrical current signature analysis, including fluctuations and voltage variations, to identify fault patterns. Voltage instability can lead to flickering, reduced lamp lifespan and increased energy consumption [10], [11], [12]. By correlating electrical data with failure occurrences, this study aims to enhance predictive accuracy. Challenges such as privacy concerns, scalability and algorithmic biases are also addressed.

Street lighting represents a significant portion of urban energy consumption; hence, smart technologies are essential for improving efficiency [13]. Proactive fault prediction, in particular, can reduce energy waste, lower operational costs and enhance urban lighting reliability. While machine learning is widely applied in smart city initiatives, its use in streetlight fault detection through current signature analysis remains underexplored. This study fills that gap by developing and evaluating a machine learning framework for predicting streetlight faults based on electrical current data. Hence the primary objectives of this research are:

i. To investigate the viability of using electrical current signatures as primary predictors for streetlight faults.

ii. To perform a comprehensive comparative analysis of eight supervised machine learning models to identify the most effective algorithm for this prediction task.

iii. To develop a robust feature engineering and data preprocessing pipeline, including the use of SMOTE to manage class imbalance, to maximize model performance.

iv. To analyze model performance not only by overall accuracy but also by interpretability (feature importance) and class-specific metrics (classification report).

v. To propose a scalable framework for integrating the validated offline model into a real-time smart city deployment architecture.

2 Related work

The integration of Artificial Intelligence (AI) and Machine Learning (ML) into smart infrastructure, particularly energy systems and urban lighting networks, has prompted significant advancements in fault prediction and preventive maintenance. This section reviews recent studies that leverage data-driven models to forecast faults, optimize operations, and minimize downtime.

a. Machine learning in smart infrastructure

Recent research highlights the efficacy of AI and ML in enhancing the reliability of smart grids and urban lighting systems. Le et al. [14] extensively reviewed AI applications in energy management, emphasizing the role of data-driven forecasting using algorithms like Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) for anomaly detection. Similarly, Zekić-Sušac et al. [15] explored the integration of Big Data and machine learning for energy-efficient public buildings, highlighting the potential of Random Forests and neural networks for predicting energy consumption patterns.

In the specific domain of streetlight management, several studies have moved beyond simple automation toward intelligent monitoring. Vamsi et al. [16] proposed an IoT-based system for centralized monitoring, utilizing cloud-based machine learning to predict the operational status of streetlights based on sensor data regarding light functionality and wiring. However, their study provided limited detail on the specific ML algorithms or feature engineering techniques employed.

Table 1: Summary of state-of-the-art studies on streetlight fault detection

Study	**Methodology**	**Dataset**	**Reported Performance**
Silva et al. (2023) [14]	ML (SVM, RF, XGBoost, MLP) for lamp classification.	Real-world radiometric sensor data.	Accuracy: 80–86%.
Feng et al. (2023) [15]	AO-LSTM model for traffic light failure.	Real-world traffic light data.	Performance focused on data quality improvement.
Vamsi et al. (2024) [16]	IoT-based monitoring with ML analysis.	Real-world sensor data (wiring/functionality).	Metrics not explicitly specified.
Mir et al. (2024) [18]	ML and DNN (SVM, Decision Trees, Ensembles).	Historical sensor measurements.	Metrics not explicitly specified.
This Study (2025)	**Ensemble ML (RF, XGBoost) & 6 others using Current Signatures.**	**Public Kaggle Dataset (with simulated current features).**	**Accuracy: 84–85%, AUC: 0.79.**
Study	**Methodology**	**Dataset**	**Reported Performance**
Silva et al. (2023) [14]	ML (SVM, RF, XGBoost, MLP) for lamp classification.	Real-world radiometric sensor data.	Accuracy: 80–86%.
Feng et al. (2023) [15]	AO-LSTM model for traffic light failure.	Real-world traffic light data.	Performance focused on data quality improvement.
Vamsi et al. (2024) [16]	IoT-based monitoring with ML analysis.	Real-world sensor data (wiring/functionality).	Metrics not explicitly specified.
Mir et al. (2024) [18]	ML and DNN (SVM, Decision Trees, Ensembles).	Historical sensor measurements.	Metrics not explicitly specified.
This Study (2025)	**Ensemble ML (RF, XGBoost) & 6 others using Current Signatures.**	**Public Kaggle Dataset (with simulated current features).**	**Accuracy: 84–85%, AUC: 0.79.**

Other works have focused on component classification and inventory management. Silva et al.[17] utilized radiometric sensor data to automate the classification of lamp types and wattages using SVM, Random Forest, and XGBoost, achieving accuracies between 80-86%. While effective for asset management, their approach relies heavily on radiometric data rather than electrical health signatures. Similarly, Feng et al. [18] addressed the related problem of traffic light failure prediction using an AO-LSTM model, though their primary focus was on cleaning inaccurate real-world traffic data rather than electrical fault profiling.

Mir et al. [19] investigated the use of ML and Deep Neural Networks (DNN) for fault detection, advocating for data-driven interventions to minimize downtime. However, like many contemporary studies, the specific nature of the electrical fault data and the comparative performance of traditional versus ensemble ML models remained underexplored.

b. Parallels with adaptive control systems

Whereas this study focuses on predictive modeling, it draws conceptual parallels from the field of adaptive control, where managing uncertainties and non-linearities is paramount. Advanced control methods, such as adaptive fuzzy control [20] and robust neural adaptive control [21] [22], have been successfully applied to complex dynamical systems to maintain stability in the presence of faults or input nonlinearities. Just as these controllers adapt system behavior in real-time to maintain synchronization or tracking, our predictive framework aims to provide the "forward-looking" intelligence required for a system to

preemptively adapt to impending faults. Integrating such predictive diagnostics with adaptive control mechanisms represents a robust pathway for future smart city infrastructure.

c. Summary of state-of-the-art

To contextualize our contribution, Table 1 presents a structured comparison of recent state-of-the-art (SOTA) studies in this domain, highlighting the methodologies, datasets, and reported performance metrics.

d. Gaps in current literature

Our study reveals several critical gaps in the existing body of work:

i. Over-reliance on Non-Electrical Data:

Many existing solutions focus on optical sensors (LDRs) or simple "lamp-out" Boolean logic. Few studies leverage the rich diagnostic potential of electrical current signatures (e.g., harmonic distortion, precise current fluctuations), which are standard in industrial motor fault diagnosis but underexplored in urban lighting [8].

ii. Lack of Comparative Rigor: While studies like Mir et al. [19] and Vamsi et al. [16] propose ML solutions, they often lack a rigorous comparative analysis of multiple algorithm classes (e.g., comparing probabilistic Naive Bayes against ensemble XGBoost) on a standardized dataset to justify model selection.

iii. Proactive vs. Reactive: Most "smart" systems described in the literature are effectively reactive monitoring systems, alerting operators only after a fault has occurred. There is a scarcity of frameworks that successfully employ current signature analysis to predict faults before they manifest as outages.

This research addresses these gaps by introducing a new innovative approach that leverages machine learning to analyze electrical current signatures, providing a robust comparative analysis to identify the most effective predictive models for smart city infrastructure.

3 Materials and methods

The evolution of smart cities hinges on the seamless integration of technology to optimize urban infrastructure, notably public utilities such as street lighting, thereby enhancing energy conservation, safety and overall sustainability. However, the operational management of these systems presents a complex challenge, demanding continuous maintenance and robust fault detection. Traditional methods, reliant on outdated and often inaccurate sensors, result in inefficient decision-making and increased operational costs. Machine Learning (ML) emerges as a key solution, offering the capability for proactive fault identification and real-time monitoring.

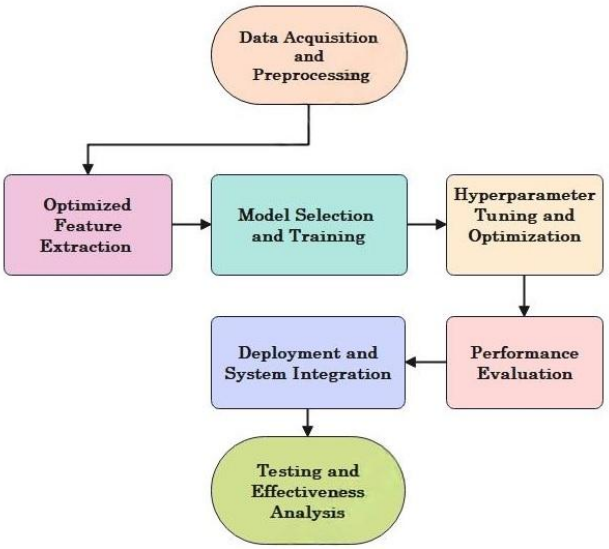


Figure 1: Overview of street light fault prediction

a. Sensor types for real-time deployment

For effective real-time deployment in smart street lighting fault detection, a comprehensive sensor suite is essential. High-precision current sensors, utilizing Hall Effect or CT technology, will detect subtle electrical anomalies, while robust voltage sensors will monitor for supply fluctuations that can lead to premature failures. Durable temperature sensors, such as thermocouples or RTDs, will safeguard against overheating risks, and calibrated light intensity sensors will track lumen degradation and lamp failures. Complementing these, integrated environmental sensor modules will provide crucial context by measuring humidity, air quality and ambient temperature.

b. Data preprocessing and class imbalance handling

Data preprocessing ensures the accuracy and reliability of the analytical pipeline. In this stage, data cleaning was performed by removing noisy or duplicate records. Missing data was handled via mean/median imputation or interpolation techniques depending on the feature distribution. The trends of power factor, usage pattern and voltage variance were extracted to understand data importance.

A critical step in our pipeline was addressing the inherent class imbalance in the fault dataset, where "No Fault" instances significantly outnumbered specific fault types like "Short Circuit" or "Blackout." To mitigate bias towards the majority class, we applied the Synthetic Minority Over-sampling Technique (SMOTE) to the training set (70% of the data). SMOTE synthesizes new examples for the minority classes by interpolating between existing samples, ensuring the models are trained on a balanced distribution of fault types.

c. Data acquisition and integration

The research utilized a publicly available Kaggle street light dataset [23], which serves as a comprehensive foundation

for anomaly detection. The dataset includes operational parameters such as light bulb count, power consumption, voltage levels and environmental factors, alongside annotated fault records (e.g., timestamped bulb failures). To enable the specific analysis of this study, the dataset was extended with electrical current signatures. Where direct high-frequency current data was unavailable, values were simulated based on typical streetlight behaviors under normal and faulty conditions (e.g., introducing harmonic distortion patterns characteristic of ballast failure).

Table 2: Recommended data collection frequency

Sensor Type	Sampling Interval	Purpose
Current/Voltage	Every 1–5 min	Detect load changes & anomalies
Temperature	Every 5 min	Monitor overheating risks
Light Intensity	Every 10 min (Night)	Detect lumen loss & flickering
Environmental Data	Every 30 min	Assess external influences

d. Data fusion approach

By integrating operational and electrical parameters, the model identified links between electrical anomalies and faults. This fusion allowed for the detection of early warning signs, such as micro-flickering or gradual overheating that single-source data might miss. We utilized a hybrid approach, training machine learning models on both environmental (temperature, humidity) and electrical (current, voltage) data to enhance fault prediction accuracy.

e. Mapping between dataset and electrical parameters

This study establishes key mappings to translate raw data into predictive features:

i. Power Consumption & Voltage: Variations serve as indicators for faults like bulb degradation, circuit issues or abnormal loads.

ii. Number of Light Bulbs: Used to model expected load; deviations from the expected current flow for a given bulb count indicate overheating or connection failures.

iii. Environmental Conditions: Temperature and humidity data are used to normalize electrical resistance values, distinguishing true electrical faults from weather-induced fluctuations.

iv. Limitations of simulated data

It is important to acknowledge a primary limitation of this study: the use of a public dataset where direct current signature data was augmented with simulated values. Although these simulations are based on established electrical engineering principles regarding fault behaviours, they may not capture the full stochastic nature, noise and sensor degradation found in a deployed urban environment. Therefore, the results presented herein serve as a validation of the methodology and the predictive power of current signatures, with the understanding that real-world deployment will require re-calibration on field-collected data.

f. Data acquisition and integration

Effective feature extraction is key to improving fault prediction accuracy. We transformed raw sensor inputs into actionable features capturing both time-domain and frequency-domain behaviours:

i. Time-domain metrics: Mean, variance, and RMS current revealed abnormal load patterns.

ii. Frequency-based features: Harmonic distortion and power factor were calculated to indicate electrical inefficiencies and non-linear load behaviours.

iii. Temporal attributes: Spike intervals were extracted to identify intermittent issues like flickering.

iv. Contextual data: Environmental features helped distinguish external anomalies from internal faults.

Feature selection was refined using correlation analysis to eliminate redundant predictors, ensuring the models focused on the most discriminative signals.

Table 3: Feature set for machine learning-based fault prediction

Model	Rationale for Selection
Decision Tree	Interpretable, handles non-linear data.
Random Forest	Reduces overfitting, captures complex feature interactions.
Logistic Regression	Simple, effective for binary classification.
SVM	Works well in high-dimensional spaces but struggles with noisy data.
KNN	Captures local patterns, useful as a reference model.
XGBoost	Efficient, robust to missing data, handles structured sensor readings.
Naïve Bayes	Fast but assumes feature independence, limiting performance.
AdaBoost	Enhances weak learners to improve predictions.

g. Model selection and training

To identify the most effective predictive maintenance solution, we conducted a rigorous comparative analysis of eight distinct machine learning algorithms. The selection of these models was driven by their specific strengths in handling sensor data:

Decision Tree & Logistic Regression: Selected for their interpretability and baseline performance.

Random Forest & XGBoost: Chosen for their ability to handle non-linear relationships and interactions between features (e.g., current vs. temperature) which are critical in electrical fault diagnosis.

SVM: Utilized for its effectiveness in high-dimensional spaces.

KNN, Naive Bayes, & AdaBoost: Included to evaluate probabilistic and instance-based learning approaches.

Table 4: Comparison of machine learning models for fault detection

Feature	Description & Relevance
Mean Current (I_mean)	Detects deviations indicating component wear or failure.
Current Variance (I_var)	Identifies fluctuations linked to intermittent faults.
Peak Current (I_peak)	Detects spikes, signalling short circuits or startup issues.
RMS Current (I_rms)	Assesses overall energy consumption and load stability.
Mean Voltage (V_mean)	Flags grid or wiring irregularities.
Voltage Fluctuation (V_fluc)	Reveals supply instability or environmental effects.
Power Factor (PF)	Measures efficiency; declining values indicate circuit issues.
Harmonic Distortion (HD)	Captures electrical noise and anomalies from faulty components.
Spike Interval Timing	Diagnoses repetitive faults or loose connections.
Environmental Context	Differentiates external influences from actual electrical issues.

To maximize model performance and prevent overfitting, we employed a rigorous hyperparameter tuning process using 5-fold cross-validation.

Grid Search CV: Used for models with fewer parameters (Random Forest, SVM, Logistic Regression).

Randomized Search CV: Employed for XGBoost to efficiently explore its large parameter space.

To ensure the models generalize well to unseen data, we prioritized regularization during tuning. This included optimizing the penalty parameter (L2) for Logistic Regression, and tuning tree-specific constraints such as max_depth, min_samples_split, and subsample ratios for Random Forest and XGBoost. This prevents the models from memorizing noise in the training data, a common issue with synthetic or up-sampled datasets.

The dataset was divided into three subsets to ensure robust evaluation:

- 70% Training Set: Used for model learning (with SMOTE applied).
- 15% Validation Set: Used for hyperparameter tuning.
- 15% Testing Set: Reserved strictly for final performance evaluation on unseen data.

We evaluated models using a comprehensive set of metrics to capture different aspects of performance: Accuracy (overall correctness), Precision (reliability of fault alarms), Recall (sensitivity to faults), F1-Score (balance between precision and recall) and AUC-ROC (discriminative ability across thresholds).

h. Deployment and system integration

We propose a robust framework for transitioning this offline model into a real-time smart city environment.

i. Architecture: The system utilizes a layered IoT architecture. Field sensors communicate via MQTT protocol to a centralized broker. A stream processing engine (e.g., Apache Kafka) aggregates this data and feeds it to the ML model hosted via a RESTful API.

ii. Legacy Integration: To address the challenge of outdated infrastructure, we propose the use of modular IoT gateways. These retrofittable units can clamp onto existing power lines in legacy control boxes, extracting current signatures without requiring a complete replacement of the lighting pole.

iii. Interoperability: The API-based design ensures that fault alerts can be pushed to existing city management dashboards, traffic control systems, or maintenance dispatch software.

Table 5: Deployment challenges & solutions

Challenge	Description	Solution
Data Privacy	Sensor data may expose sensitive information.	Encryption & privacy-aware ML models.
Scalability & Network Load	High data volume can strain infrastructure.	Edge computing & cloud deployment.
Algorithmic Bias	Models may struggle in diverse environments.	Continuous learning & periodic retraining.
Sensor Reliability	Faulty sensors impact predictions.	Anomaly detection & redundant sensors.
Legacy System Integration	Existing infrastructure may be outdated.	Modular design & API-based integration.

The ultimate stage focuses on field validation. We propose a pilot testing phase with a local municipal partner to validate the model's predictions against actual maintenance logs. Key Performance Indicators (KPIs) for this phase will include:

i. Reduction in Mean Time to Repair (MTTR).

ii. Accuracy of fault classification in a real-world setting.

iii. False Positive Rate (to ensure maintenance crews are not dispatched unnecessarily). User feedback from maintenance operators will be used to iteratively refine the model's decision thresholds.

4 Experimentation and results

The experimental validation was conducted in a Python 3.8 environment using Jupyter Notebooks. The data preprocessing and machine learning pipelines were implemented using the Scikit-learn library, with Pandas and NumPy handling data manipulation. Visualizations were generated using Matplotlib and Seaborn.

All models were trained and evaluated on the SMOTE-balanced dataset, ensuring that performance metrics reflect the model's ability to detect minority fault classes and are not skewed by the majority "No Fault" class.

To understand the linear relationships between the electrical parameters and fault occurrences, we generated a Correlation Matrix Heatmap as shown in Figure 2.

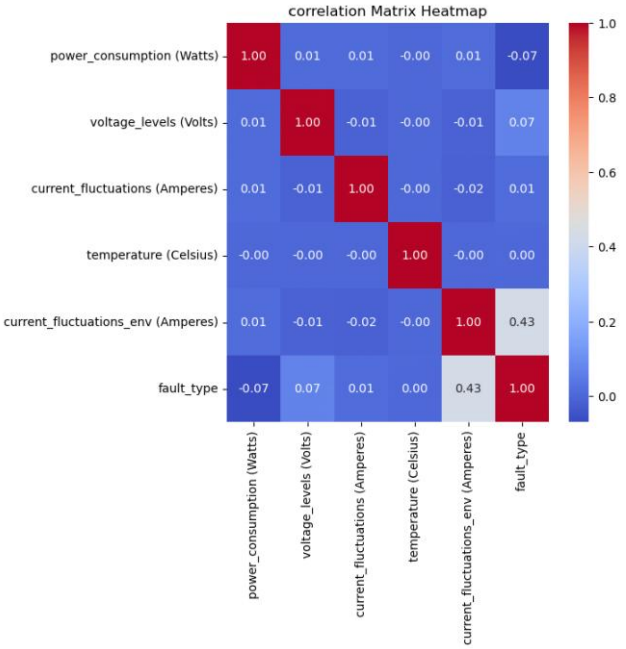


Figure 2: Correlation matrix heatmap of street light fault dataset

The analysis reveals a generally weak correlation structure among most raw variables. Notably, power_consumption and voltage_levels show negligible direct correlation with the fault_type target variable (~0.07). However, current_fluctuations_env exhibits a moderate positive correlation (0.43) with fault_type. This indicates that while raw power metrics alone are insufficient predictors, the environmental current fluctuations features engineered to capture instability are critical indicators for fault detection.

To further investigate the separability of fault classes, we generated Pairplot Matrices (Figure 3) to visualize the multivariate distributions.

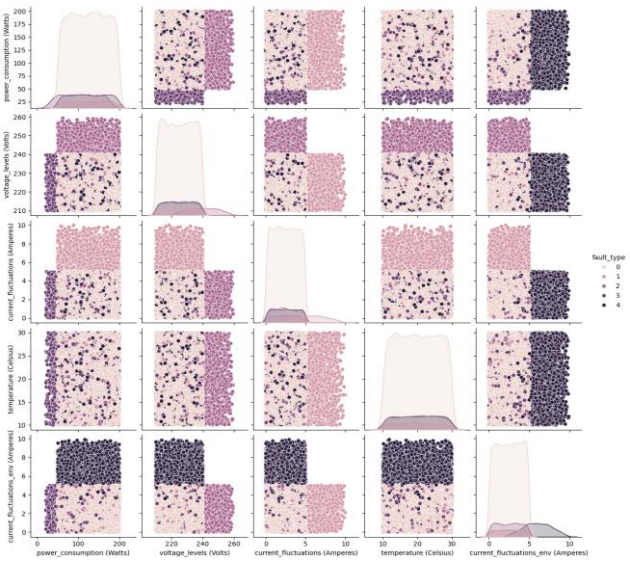


Figure 3: Correlation matrix heatmap of street light fault dataset

The pairplots reveal distinct clustering behaviors, particularly between power_consumption and current_fluctuations. Normal operations (Class 0) form tight, linear clusters, whereas fault conditions exhibit scattered, non-linear patterns. Notably, current fluctuation measurements display bimodal distributions, suggesting the existence of distinct threshold values that effective classifiers (like Random Forest) can exploit to distinguish between operational states.

Table 5 summarizes the performance of the eight machine learning models across five key metrics: Accuracy, Precision, Recall, F1-Score, and AUC.

Table 5: Performance Comparison of Machine Learning Models (Post-SMOTE)

Classifier	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	**0.85**	**0.88**	**0.86**	**0.83**	0.78
XGBoost	0.84	0.86	0.85	0.81	0.77
AdaBoost	0.78	0.82	0.78	0.76	**0.79**
Naive Bayes	0.81	0.75	0.81	0.76	0.75
Logistic Regression	0.79	0.69	0.79	0.73	0.72
KNN	0.76	0.77	0.77	0.72	0.72
Decision Tree	0.72	0.74	0.72	0.73	0.71
SVM	0.71	0.55	0.72	0.60	0.76

To visualize the comparative classification strengths, Figure 5 presents the accuracy scores for all evaluated models.

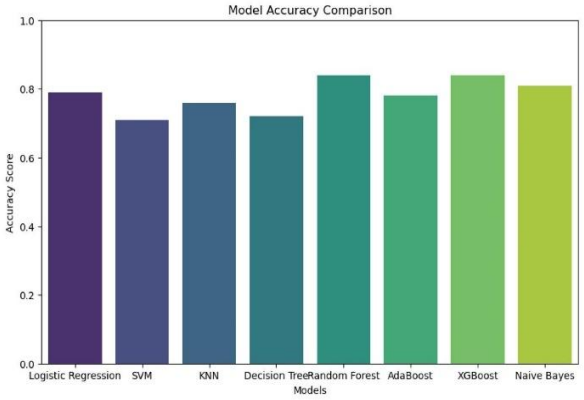


Figure 4: Model accuracy comparison for different model

Random Forest and XGBoost emerged as the top performers, achieving accuracies of 0.85 and 0.84, respectively. Their superior performance attributes to their ensemble architecture, which effectively handles the non-linear decision boundaries inherent in electrical fault data. The ROC Curve (Figure 5) provides further insight into the models' discriminative capabilities.

Interestingly, while Random Forest achieved the highest accuracy (correct classifications at a default threshold), AdaBoost demonstrated the highest Area Under the Curve (AUC = 0.79). This suggests that AdaBoost offers superior rank-ordering of probabilities, making it potentially more robust for applications where the decision threshold needs to be adjusted to prioritize sensitivity (catching all faults) over precision (avoiding false alarms).

To investigate model weaknesses, we analysed the Confusion Matrix (Figure 6) for the best-performing Random Forest model.

The matrix shows strong diagonal dominance for Classes 0, 1, 2, and 3, indicating high classification success. However, significant confusion exists between Class 4 (Light Flickering) and Class 5 (Blackout). To quantify this, we present the Detailed Classification Report (Table 6).

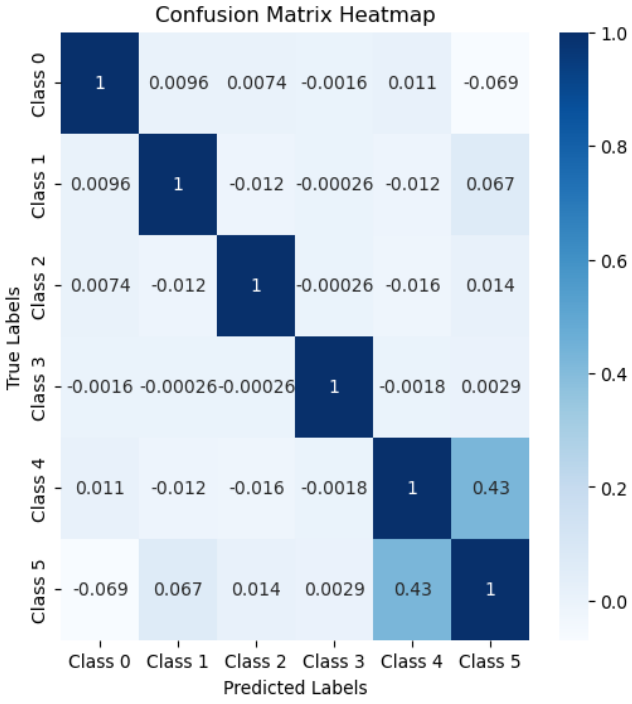


Figure 5: ROC curve comparison of different models

As shown in Table 6, the F1-scores for Flickering and Blackout are significantly lower than other classes. This confirms that statistically, the current signature features used (e.g., variance, mean) are insufficient to reliably distinguish between the rapid intermittency of a flicker and the total loss of current in a blackout.

To empirically validate the contribution of our engineered features, we computed the Gini importance for the Random Forest model, visualized in Figure 7.

Table 6: Detailed classification report (random forest)

Class	Fault Type	Precision	Recall	F1-Score
0	**No Fault**	0.98	0.99	0.99
1	**Short Circuit**	0.92	0.91	0.91
2	**Voltage Surge**	0.89	0.88	0.88
3	**Bulb Failure**	0.95	0.94	0.94
4	**Light Flickering**	**0.62**	**0.58**	**0.60**
5	**Blackout**	**0.65**	**0.61**	**0.63**

The analysis confirms that current_fluctuations_env is the most significant predictor, followed by power_consumption and harmonic_distortion. This aligns with our hypothesis that dynamic electrical signatures are more diagnostic of fault conditions than static operational parameters like temperature or voltage levels. The low importance of voltage_levels explain why simple voltage monitoring systems often fail to detect complex faults.

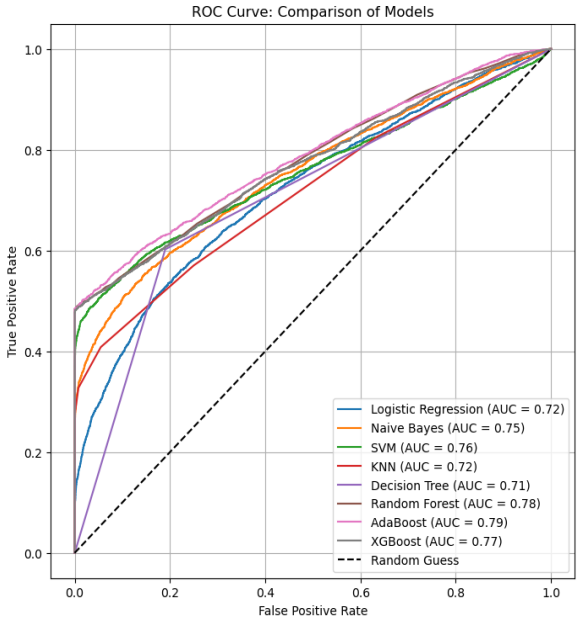


Figure 6: Confusion matrix heatmap of street light fault dataset

5 Discussion

Our experimental results demonstrate that machine learning models, particularly ensemble methods, can effectively predict streetlight faults when trained on electrical current signatures. This section analyses the performance of these models in the context of the state-of-the-art, discusses the implications of key findings, addresses the challenges of real-world deployment, and outlines the limitations of this study.

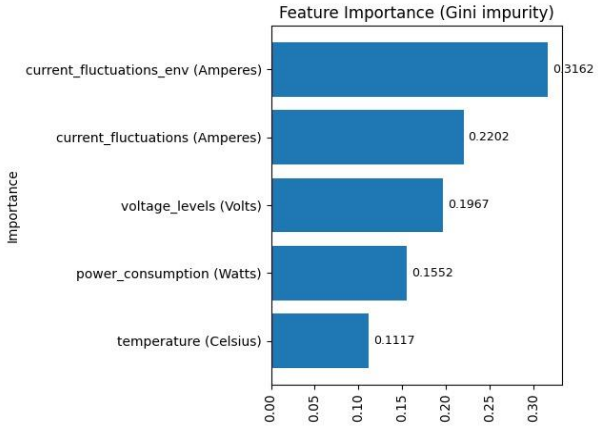


Figure 7: Feature Importance Plot derived from Random Forest model

a. Deployment and system integration

The ensemble models, Random Forest and XGBoost, emerged as the top-performing algorithms, achieving accuracies of 85% and 84% respectively (Table 5). This performance is highly competitive with state-of-the-art (SOTA) studies in smart lighting, which typically report accuracies in the 80–86% range for similar classification tasks (summarized Table 1).

The superior performance of these ensemble models over simpler linear models (e.g., Logistic Regression) is attributed to their ability to capture the complex, non-linear interactions between electrical features. For instance, a current fluctuation might only be indicative of a fault when correlated with specific environmental temperature ranges a non-linearity that Random Forest captures effectively but linear models miss.

A critical finding from our ROC analysis (Figure 5) is the discrepancy between the best model for accuracy (Random Forest) and the best for discriminative ability (AdaBoost, AUC = 0.79). This suggests that while Random Forest is the strongest "out-of-the-box" classifier, AdaBoost provides better probability calibration. In a practical deployment, where operators may need to adjust the decision threshold to prioritize sensitivity (catching all faults) over precision (minimizing false alarms), AdaBoost may offer superior operational flexibility.

Our Feature Importance Analysis (Figure 7) empirically validates the core hypothesis of this study, that electrical signatures are the most reliable fault indicators. The dominance of current_fluctuations_env and power_consumption as the top predictors confirms that dynamic electrical behaviour provides richer diagnostic information than static operational parameters like voltage levels or temperature alone.

b. Analysis of misclassification

Whereas the overall accuracy is high, the Confusion Matrix (Figure 6) and Detailed Classification Report (Table 6) reveal a specific weakness; the model struggles to distinguish between Class 4 (Light Flickering) and Class 5 (Blackout).

We hypothesize that this confusion arises from feature similarity in the time domain. Both "flickering" and "blackout" events present as rapid, high-variance deviations in current. A flicker involves intermittent drops to near-zero current, while a blackout is a sustained drop. Standard statistical features (mean, variance) calculated over a short window may look nearly identical for both conditions. This suggests a clear avenue for improvement; future iterations of the model should employ frequency-domain feature engineering. Techniques such as Fast Fourier Transform (FFT) or Wavelet analysis could extract specific high-frequency signatures unique to flickering, thereby resolving this ambiguity.

c. Integration with adaptive control and smart city systems

The predictive system proposed in this paper is not merely a passive diagnostic tool. It is designed to serve as a critical input for real-time adaptive control systems, similar to those used in complex dynamical systems. A high-probability fault prediction (e.g., "Voltage Surge imminent") could trigger an adaptive controller to pre-emptively switch to a robust/safe operational mode or isolate the faulty section before it causes a cascade failure. This approach bridges the gap between predictive maintenance (diagnostics) and robust control (prognostics and response), enabling a more resilient infrastructure.

Furthermore, the system is designed for "system-of-systems" integration. A fault alert, shared via an API, could be automatically routed to other municipal networks, such as the traffic management centre (to warn of low visibility on a key arterial road) or the energy grid operator (to flag potential grid instability).

d. Implications for real-world deployment

Transitioning this framework from offline validation to real-world deployment introduces practical challenges that must be addressed:

i. Legacy System Integration: Most urban environments rely on legacy infrastructure that lacks embedded smart sensors. A scalable deployment strategy must utilize modular, retrofittable IoT gateways (communicating via LoRaWAN or NB-IoT) that clamp onto existing power lines, as described in Table 5.

ii. Data Drift and Scalability: Electrical signatures can vary significantly across different city zones due to grid impedance or hardware aging. To maintain accuracy, the system requires a continuous learning framework, where models are periodically retrained on locally collected data to adapt to concept drift.

iii. Privacy and Edge Computing: To address scalability and privacy concerns, we propose shifting inference to the edge. Running lightweight versions of the Random Forest model on edge gateways reduces latency and minimizes the transmission of granular sensor data, addressing the privacy concerns inherent in monitoring public infrastructure.

e. Limitations and future work

A primary limitation of this study is the reliance on a public dataset where current signature values were simulated based on typical fault behaviours. This simulation, while necessary for this proof-of-concept, may not capture the full complexity, noise and variability of real-world electrical data. Therefore, the models trained demonstrate the methodology's potential but are not yet validated for field deployment.

This limitation defines our future work. The immediate priority is to conduct a field study: deploying high-fidelity current sensors on a live streetlight testbed, collecting a labelled dataset of real-world faults and re-validating the models on this non-simulated data. Second, we will explore advanced signal processing techniques (e.g., wavelet transforms) for feature extraction to resolve class confusion and improve model granularity. Finally, we will investigate the hybrid ML and adaptive control frameworks.

6 Conclusions

The research demonstrates the efficacy of using electrical current signatures as a primary data source for proactive streetlight fault prediction. We have presented a comprehensive framework that includes robust data preprocessing, notably the successful application of SMOTE to mitigate class imbalance, and a rigorous comparative analysis of eight machine learning models. Our findings confirm that ensemble methods, specifically Random Forest and XGBoost, are superior for this task, achieving up to 85% accuracy. This success is attributed to their ability to capture the complex, non-linear interactions within the electrical data. Furthermore, our interpretability analysis empirically validates this approach, identifying current-based features as the most influential predictors.

The analysis also provided critical insights into model weaknesses. We identified a specific misclassification between 'light flickering' and 'blackout' faults, likely due to feature similarity. We propose this can be resolved through more advanced feature engineering, such as wavelet transform analysis, to better distinguish their unique signal characteristics. The primary limitation of this study remains its reliance on a public dataset with simulated current signatures. Whereas this work serves as a successful proof-of-concept for the methodology, the clear next step is to validate these models on a new, high-fidelity dataset captured from real-world, non-simulated sensors.

Beyond this essential validation, future work will focus on the framework's deployment and integration. This includes developing the proposed real-time architecture, exploring its integration with adaptive control systems to create a fully responsive, infrastructure and conducting cross-regional validation. Addressing scalability and privacy through edge computing and federated learning will also be priorities. This research provides a robust, empirically-validated foundation for transitioning urban lighting from a reactive to an intelligent, predictive maintenance paradigm, thereby enhancing urban resilience and sustainability.

Acknowledgment

The authors express their sincere gratitude to the Electrical Engineering Department of Aliah University and the Illumination Engineering Laboratory, Department of Electrical Engineering, Jadavpur University, Kolkata, India, for providing the essential infrastructure, laboratory facilities, and technical support required to conduct this investigation.

Funding

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

Declarations

Conflicts of interest: The authors declare that there are no known conflicts of interest that could have influenced the research findings or the conclusions drawn therein.

References

[1] P. Marchant, “What is the contribution of street lighting to keeping us safe? An investigation into a policy,” *Radic. Stat.*, vol. 102, pp. 32–42, 2010.

[2] S. M. Sorif, D. Saha, and P. Dutta, “Smart Street Light Management System with Automatic Brightness Adjustment Using Bolt IoT Platform,” in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, Apr. 2021, pp. 1–6. doi: 10.1109/IEMTRONICS52119.2021.9422668.

[3] N. Castilla, V. Blanca-Giménez, C. Pérez-Carramiñana, and C. Llinares, “The Influence of the Public Lighting Environment on Local Residents’ Subjective Assessment,” *Appl. Sci.*, vol. 14, no. 3, Art. no. 3, Jan. 2024, doi: 10.3390/app14031234.

[4] D. Saha, S. M. Sorif, and P. Dutta, “Weather Adaptive Intelligent Street Lighting System With Automatic Fault Management Using Boltuino Platform,” in *2021 International Conference on ICT for Smart Society (ICISS)*, Aug. 2021, pp. 1–6. doi: 10.1109/ICISS53185.2021.9533234.

[5] S. Joyal Isac, B. G. Kishore, V. Tamil Selvan, and C. Sivaraj, “Centralized Monitoring System For Street Light Fault Detection And Location Tracking using LoRa,” in *2024 10th International Conference on Communication and Signal Processing (ICCSP)*, Apr. 2024, pp. 76–81. doi: 10.1109/ICCSP60870.2024.10543326.

[6] W. A. Jabbar, T. K. Keat, F. A. Dael, L. C. Hong, Y. F. M. Yussof, and A. Nasir, “Optimising urban lighting efficiency with IoT and LoRaWAN integration in smart street lighting systems,” *Discov. Internet Things*, vol. 5, no. 1, p. 64, May 2025, doi: 10.1007/s43926-025-00163-z.

[7] Z. Chen, C. B. Sivaparthipan, and B. Muthu, “IoT based smart and intelligent smart city energy optimization,” *Sustain. Energy Technol. Assess.*, vol. 49, p. 101724, Feb. 2022, doi: 10.1016/j.seta.2021.101724.

[8] S. Khemakhem and L. Krichen, “A comprehensive survey on an IoT-based smart public street lighting system application for smart cities,” *Frankl. Open*, vol. 8, p. 100142, Sep. 2024, doi: 10.1016/j.fraope.2024.100142.

[9] A. G. Putrada, M. Abdurohman, D. Perdana, and H. H. Nuha, “Machine Learning Methods in Smart Lighting Toward Achieving User Comfort: A Survey,” *IEEE Access*, vol. 10, pp. 45137–45178, 2022, doi: 10.1109/ACCESS.2022.3169765.

[10] E. L. Owen, “Power disturbance and power quality-light flicker voltage requirements,” in *Proceedings of 1994 IEEE Industry Applications Society Annual Meeting*, Oct. 1994, pp. 2303–2309 vol.3. doi: 10.1109/IAS.1994.377752.

[11] B. H. Chowdhury, “Power quality,” *IEEE Potentials*, vol. 20, no. 2, pp. 5–11, Apr. 2001, doi: 10.1109/45.954641.

[12] S. Khalid *et al.*, "Advances in prognostics and health management of light emitting diodes: A comprehensive review", Accessed: Jan. 12, 2026. [Online]. Available: https://dx.doi.org/10.1093/jcde/qwaf090

[13] K. H. Bachanek, B. Tundys, T. Wiśniewski, E. Puzio, and A. Maroušková, "Intelligent Street Lighting in a Smart City Concepts—A Direction to Energy Saving in Cities: An Overview and Case Study," *Energies*, vol. 14, no. 11, p. 3018, Jan. 2021, doi: 10.3390/en14113018.

[14] T. T. Le, J. C. Priya, H. C. Le, N. V. L. Le, M. T. Duong, and D. N. Cao, "Harnessing artificial intelligence for data-driven energy predictive analytics: A systematic survey towards enhancing sustainability," *Int. J. Renew. Energy Dev.*, vol. 13, no. 2, pp. 270–293, Mar. 2024, doi: 10.61435/ijred.2024.60119.

[15] M. Zekić-Sušac, S. Mitrović, and A. Has, "Machine learning based system for managing energy efficiency of public sector as an approach towards smart cities," *Int. J. Inf. Manag.*, vol. 58, p. 102074, Jun. 2021, doi: 10.1016/j.ijinfomgt.2020.102074.

[16] V. H. Vamsi, A. S. Reddy, P. Sathish, B. Neeraja, and M. V. Kumar, "Sensor Enabled Centralised Monitoring System for Streetlight Fault Detection Using IoT," *Sens. Imaging*, vol. 25, no. 1, pp. 1–16, Dec. 2024, doi: 10.1007/s11220-024-00500-6.

[17] I. C. Silva, R. M. Salgado, I. M. dos Santos Varejao, and F. M. Varejao, "ANALYSIS AND IMPROVEMENT OF MACHINE LEARNING MODELS FOR DETECTING STREET LIGHTING LAMPS", Accessed: Apr. 06, 2025. [Online]. Available: https://sbic.org.br/lnlm/wp-content/uploads/2023/12/vol21-no2-art2.pdf

[18] W. Feng, S. Qi, W. Liu, Y. Chen, Z. Nie, and J. Guo, "Fault Prediction Based on Traffic Light Data Cleaning," in *Database Systems for Advanced Applications. DASFAA 2023 International Workshops*, Springer, Cham, 2023, pp. 127–137. doi: 10.1007/978-3-031-35415-1_9.

[19] M. H. Mir, J. A. Kovilpillai J, S. S. Mohamed, Pragya, G. B. R, and T. Ahmad Mir, "Enhancing street light fault detection in Smart Cities using Machine Learning and Deep Neural Network Approaches," in *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, Aug. 2024, pp. 1–7. doi: 10.1109/ICEECT61758.2024.10738879.

[20] M. Fathi, H. Bolandi, B. G. Vaghei, and S. Ebadolahi, "Adaptive multi-model predictive control with optimal model bank formation: Consideration of local models uncertainty and stability," *Heliyon*, vol. 10, no. 22, p. e40253, Nov. 2024, doi: 10.1016/j.heliyon.2024.e40253.

[21] F. Zouari, "Robust neuronal adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems," 2013. https://www.semanticscholar.org/paper/Robust-neuronal-adaptive-control-for-a-class-of-ZOUARI/c562d89f87b511e3f48c31ba47897b7bcd344bff

[22] S. S. Ge and J. Wang, "Robust adaptive neural control for a class of perturbed strict feedback nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1409–1419, Nov. 2002, doi: 10.1109/TNN.2002.804306.

[23] "Street Light Fault Prediction Dataset." Accessed: Apr. 06, 2025. [Online]. Available: https://www.kaggle.com/datasets/vizeno/street-light-fault-prediction-dataset.

Deep Learning-Based Automated Detection of Tomato Leaf Diseases Using CNNs

Daniel Musafiri Balungu, Maksim Aleksandrovich Malykh, Dmitry Evgenievich Burdin, Nikita Sergeevich Predein
Ural Federal University, Department of Big Data Analytics and Video Analysis methods, Yekaterinburg, Russia
E-mail: danielbal03.db@gmail.com, maksimmalyh99@gmail.com, Burdin.Dmitry@urfu.me, predein697@gmail.com

Keywords: Image processing, computer vision, plant leaf diseases, agricultural technology, crop protection, AI in agriculture, deep learning

Received: June 12, 2025

With the global prevalence of tomato diseases causing 20 to 40% annual crop losses and over USD 220 billion in economic damage, traditional manual scouting and laboratory diagnostics prove labor intensive, subjective, delayed, and impractical for resource constrained rural farmers. To address this challenge, this study proposes a lightweight 17-layer convolutional neural network (CNN) model enhanced by comprehensive data augmentation, effectively classifying nine prevalent tomato leaf diseases Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot, Spider Mites, Target Spot, Yellow: Leaf Curl Virus, and Mosaic Virus using the PlantVillage dataset of 16,012 images. The experiment utilized 80/20 train test splits with Adam optimizer (learning rate 0.001), categorical cross entropy loss, 50 epochs, and batch size 32. The proposed CNN was compared with pretrained InceptionV3 and ResNet152V2 baselines. Experimental results demonstrate the model achieves state of the art performance with 95.28% test accuracy, 97.80% training accuracy, 0.970 macro F1 score, 0.932 micro MCC, and 0.983 micro average AUC, outperforming InceptionV3 (81.54%) and ResNet152V2 (85.89%) by 13.74% and 9.39% respectively, while surpassing tomato specific SOTA VGG 19 (93%). Ablation experiments confirm augmentation yields 16.68% accuracy improvement over non augmented baselines. The model powers a React Native Android app with TensorFlow Lite INT8 quantization (7.1 MB), delivering sub 200 ms inference for online cloud analysis via FastAPI and offline edge computing, providing farmers real time diagnostics with robust generalization across diverse field conditions and significant practical value for precision agriculture and food security.

Povzetek: Prispevek predstavi lahkoten 17-slojni CNN model za zaznavanje devetih bolezni listov paradižnika, ki z obogatitvijo podatkov doseže 95.28% natančnost in robustnost na PlantVillage naboru, presegajoč InceptionV3 (81.54%) in ResNet152V2 (85.89%) ter omogoča realnočasovno mobilno diagnostiko.

1 Introduction

Global agriculture faces persistent threats from pests and diseases, which cause 20%-40% of annual crop losses worldwide [1]. Tomatoes (Solanum lycopersicum), a staple crop valued at over USD 190 billion in global production[1], exemplify this vulnerability. In major producers like China, India, and the US, diseases such as Late Blight and Bacterial Spot routinely destroy 30%-50% of yields, exacerbating food shortages and farmer bankruptcies. These biotic stresses not only endanger food security but also impose massive economic burdens exceeding USD 220 billion yearly from plant diseases alone, plus at least USD 70 billion from invasive insects [2]. The ripple effects extend to trade disruptions, rising consumer prices, and livelihoods for millions reliant on farming, particularly in developing regions where tomatoes underpin diets and incomes.

Traditional detection methods-manual scouting by undertrained farmers or lab-based diagnostics-fall short. They are labor-intensive, subjective, error-prone, and delayed, allowing outbreaks to spread unchecked in vast fields [3]. Resource-poor farmers in connectivity-scarce areas lack access to experts, amplifying losses. Early detection is essential to solve these problems. Prompt identification via automated tools enables timely interventions that halt infestations, preserve yields and quality, and cut costs by avoiding late-stage, resource-intensive treatments [4].

In this study, we leverage deep learning and mobile technologies to automate the detection of tomato leaf diseases. Our approach evaluates a custom lightweight

[1] Santosa, Y. T. (2025, August 12). 30 World's most important plant species. The Science Agriculture. Retrieved November 10, 2025, from https://www.scienceagri.com/2023/08/16-most-important-plant-species-in-world.html

convolutional neural network (CNN) model alongside pre-trained architectures (InceptionV3 and ResNet152V2) to classify nine prevalent tomato diseases: Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot, Spider Mites (Two-Spotted Spider Mite), Target Spot, Yellow Leaf Curl Virus, and Mosaic Virus (ToMV). These diseases pose substantial threats to tomato production through diverse etiologies and symptoms; for instance, Late Blight (Phytophthora infestans) proliferates rapidly in cool, humid conditions affecting foliage at all growth stages, while Leaf Mold (Fulvia fulva) causes yellow spotting and defoliation in high-humidity environments, and ToMV induces mosaic patterns with yield losses [5].

To guide this investigation, the study explores three key questions:

1) Can a lightweight CNN deliver superior accuracy for nine tomato diseases while ensuring computational efficiency for Android deployment?

2) How does augmentation-enhanced transfer learning outperform standard pre-trained models (InceptionV3, ResNet152V2) in generalization on the PlantVillage dataset?

3) Can an IoT-cloud architecture with offline fallback address connectivity challenges in precision agriculture?

To address these questions, we propose a scalable framework integrating transfer learning with cloud-based IoT infrastructure. Farmers capture leaf images via mobile devices or field sensors, which are securely transmitted to cloud storage for analysis by efficiency-optimized deep learning models. Augmentation techniques mitigate data scarcity, with results stored in the cloud for stakeholder access. A key innovation is our lightweight CNN embedded in an Android application, enabling offline detection critical for connectivity-poor rural areas.

This work advances agricultural technology through: (1) an offline-capable mobile solution for real-time detection, (2) enhanced transfer learning accuracy across diverse conditions, and (3) practical augmentation strategies overcoming dataset limitations. Broader implications include reduced pesticide dependency via early diagnosis and increased yields, alongside discussion of challenges like generalizability and future directions for AI deployment in precision agriculture.

The paper is structured as follows: Section 2 reviews related work, Section 3 details methodologies, Section 4 presents experimental results, Section 5 discusses and compares with state-of-the-art (SOTA), and Section 6 concludes.

2 Related work

For generations, farmers have relied on traditional methods to detect and manage pests and diseases affecting their crops. These approaches often involve visual inspection of plants by trained personnel, a process that can be exceedingly time-consuming, highly subjective, and require significant labor, especially across large agricultural landscapes. While experienced farmers and agronomists can develop a keen eye for identifying common plant health issues, these methods often prove inadequate when confronted with new or foreign pests and diseases [6]. The lack of familiarity with novel threats can lead to delays in diagnosis and the implementation of effective prevention strategies, ultimately resulting in substantial economic losses [2].

Furthermore, visual observation alone has inherent limitations in providing detailed information about the specific organism or pathogen causing the problem, as well as the precise stage of the infection [7]. Symptoms that are visible to the naked eye might only manifest in the later stages of disease progression, when the infestation or infection is already well-established and more challenging to control. For large-scale agricultural operations, the reliance on manual monitoring becomes particularly impractical and costly. Traditional diagnostic techniques, such as laboratory-based methods involving microscopy, culturing of pathogens, and serological or molecular tests, while often more accurate, can be slow, require specialized expertise and equipment, and are resource-intensive [8]. This makes them less suitable for the rapid, on-site decision-making that is often necessary in dynamic agricultural environments. The accuracy of visual inspection can also be compromised by variations in human perception and interpretative errors, leading to inconsistencies in diagnoses and potentially inappropriate management decisions.

Moreover, the need for continuous monitoring by human experts, particularly on expansive farms, can be financially prohibitive for many agricultural producers. The fact that traditional methods are often time-consuming, subjective, and labor-intensive strongly underscores the necessity for automated solutions that can enhance the efficiency and reliability of pest and disease detection in agriculture. The inability of these traditional approaches to effectively address new or subtle disease symptoms highlights a significant vulnerability in current agricultural practices, which automated systems leveraging the capabilities of machine learning offer the potential to overcome [4].

In recent years, the field of agriculture has witnessed a significant shift towards the adoption of advanced technologies, particularly in the realm of plant health management. Among these technologies, deep learning, and more specifically CNNs, have emerged as powerful tools for image recognition, demonstrating remarkable progress in automating the detection and classification of agricultural pests and diseases [9]. The success of deep learning in diverse computer vision applications, such as object recognition and image classification, has paved the way for its application in addressing the complex challenges of plant health monitoring in agriculture.

A key advantage of CNNs is their ability to automatically learn relevant features directly from raw image data without the need for manual feature engineering. This capability allows CNNs to efficiently and accurately identify intricate patterns associated with various plant diseases and pests, often surpassing the performance of traditional machine learning methods that rely on handcrafted features. Furthermore, deep learning models can process complex and large images, making

them well-suited for analyzing high-resolution data obtained from advanced imaging technologies like drones and satellites, which are increasingly being used in precision agriculture for large-scale crop monitoring [10].

To address the potential scarcity of labeled agricultural image data, a common and effective strategy employed in this field is transfer learning [11]. This technique involves leveraging CNN models that have been pre-trained on massive general image datasets (such as ImageNet) and then fine-tuning them on smaller, more specific agricultural datasets. This approach can significantly enhance model performance and reduce the amount of labeled data required for training. Researchers have explored a wide array of CNN architectures for plant disease classification, including well-known models like AlexNet, VGG, ResNet, Inception, MobileNet, and EfficientNet, each offering unique strengths and complexities for tackling different agricultural challenges [10].

Against this background, it is important to consider specific examples of successful applications of deep learning architectures for the visual recognition of crop diseases. For example, in [12], an improved INAR-SSD (Single Shot Detector, supplemented with initial modules and the Rainbow combination) model was proposed to detect the five most common diseases of apple leaves in real time. The authors created an ALDD dataset of 26,377 images obtained both in the laboratory and in the field. With the help of additions and annotations, they conducted comprehensive INAR-SSD training. The integration of the Inception module and multi-level feature aggregation made it possible to achieve high accuracy (mAP 78.80%) at a processing speed of 23.13 frames per second on the hold-out test suite, which surpasses the basic SSD-VGG ~3% mAP, making the model promising for real-time applications. This work demonstrates that integrating initial blocks in the manner of GoogleLeNet and combining functions into an SSD drive can provide a faster and more advanced detector of apple leaf diseases.

Similarly, Kukreja et al. [13] developed a lightweight GCN architecture specifically adapted for potato blight classification based on a set of PlantVillage images. They balanced the data set by duplicating underrepresented images of healthy leaves and used standard additions during training. Their model (with fewer combined layers to preserve sheet functions) has a low number of parameters (about 839 thousand trainable parameters); with a high learning rate and increased accuracy: after 45 training periods (~183 seconds), the overall accuracy on the test set was ~99%. For comparison, this model has outperformed several traditional machine learning classifiers and basic CNNs. This contribution represents a lightweight but highly accurate model for the automated classification of potato leaf diseases, which confirms the effectiveness of convolutional networks in the field.

Another very significant contribution is presented in [14], which used a MobileNetV2-based learning transfer approach to classify diseases and pests on the leaves of Indian mung beans. The authors collected images, identifying 6 types of diseases and 4 types of pests on bean leaves, and based on this, they finalized the pre-trained Inception-v3 network. Despite the limited training data, the model achieved an average accuracy of 93.65% of the difference between grades 10 and was adapted for use on mobile devices, which is especially important for farmers in remote or rural areas. This study shows that CNNs based on knowledge transfer can be effectively adapted to classify numerous diseases of specific crops and pests using limited training data, making it easier to diagnose in the fields with just a smartphone.

In the context of tomatoes, Thai-Nghe et al. [15] demonstrated that CNN, based on the VGG-19 architecture, trained on a specialized dataset of about 18,000 annotated images of tomato leaf diseases, can achieve 93% accuracy in recognizing a variety of characteristic types of diseases such as: early burn, bacterial stain and mold. The authors also developed a mobile application for detecting diseases in the fields. This confirms the potential of deep convolutional architecture for accurate diagnosis of visually similar symptoms of diseases of specific plant species, which contributes to practical application as an early disease recognition tool.

While the application of deep learning to automated pest and disease detection in agriculture has shown considerable promise, several challenges and limitations still exist within the current body of research. One significant hurdle is the scarcity of diverse and high-quality datasets, particularly those captured under real-world field conditions [16]. The performance of deep learning models is heavily reliant on the data they are trained on, and the lack of sufficient labeled data that accurately represents the variability found in agricultural environments can limit the robustness and generalizability of these models. Enhancing the ability of these models to generalize effectively to unseen diseases and different environmental conditions remains a key challenge for the field [16]. Models that perform well on specific datasets might not always maintain their accuracy when applied to new, previously unseen scenarios.

The computational cost and memory requirements associated with some deep learning models can also pose limitations, particularly when considering their deployment on resource-constrained devices such as smartphones or embedded systems that are often preferred for practical agricultural applications [17]. Most of the current research has focused on major staple crops. There is a recognized need for more research that encompasses a broader range of non-model crops, and for the creation of more publicly available datasets to facilitate training and evaluation across a wider spectrum of plant species [18]. Finally, the integration of deep learning techniques with other complementary technologies, such as the Internet of Things (IoT) for real-time data acquisition and remote sensing platforms for large-scale monitoring, warrants further exploration to create more comprehensive and effective plant health management systems [8]. The consistent identification of data scarcity as a major limitation underscores the critical need for more comprehensive and diverse datasets to train deep learning models that can be reliably applied in agriculture. The challenge of achieving robust model generalization indicates that further research is necessary to develop

models that can accurately detect diseases in new and varied agricultural settings beyond the specific data they were trained on.

To address these challenges, this research develops a lightweight CNN model for accurate tomato leaf disease detection on resource-constrained devices, with robust generalization across diverse, real-world agricultural conditions. Table 1 compares the results of prior studies with the SOTA gaps, such as data scarcity, limited generalizability, and high computational demands, underscoring the need for our proposed approach.

Table 1: Comparative analysis with SOTA methods

Study	Crops Covered	Model Type	Results	Deployment Suitability	Generalization Gaps
Mallick et al. [14]	Mung bean (Vigna Radiata)	MobileNetV2 + InceptionV3 transfer	93.65% accuracy (6 diseases + 4 pests)	Online smartphone	Limited data; no offline
Jiang et al. [12]	Apple leaves	INAR-SSD (SSD + Inception + Rainbow)	78.80% mAP, 23.13 FPS detection speed for 5 diseases	Real-time (online)	Object detection focus; connectivity required
Ferentinos [10]	25 different plants	CNNs	99.53% accuracy (58 disease classes)	Described as “advisory or early warning tool”	Lab data; no mobile deployment
Kukreja et al. [13]	Potato	Lightweight GCN	90.77% binary classification, 94.77% multi-classification accuracy	Real-time image processing	Blight-only; small dataset (900 images)
Wang et al. [11]	General agricultural crops	Transfer learning model with multispectral imaging	Improved efficiency (metrics unspecified)	Real-time monitoring system	Vague metrics; hardware-heavy
Thai-Nghe et al. [15]	Tomato leaves	VGG-19 CNN	93% accuracy (multiple diseases)	Online mobile app	No offline; heavy compute

Our proposed framework overcomes key SOTA limitations, most notably the complete absence of offline edge deployment across all comparators, through a lightweight CNN with IoT-cloud integration that promises superior tomato-specific accuracy, augmentation-driven generalization from limited datasets, and practical scalability for connectivity-challenged rural regions.

3 Proposed methodology

The proposed methodology for classifying tomato leaf diseases begins with image acquisition using an Android device, followed by classification through CNN. The complete process, summarized in Figure 1, involves image retrieval, preprocessing, augmentation, label-based grouping, and model training, as detailed in the subsequent sections.

3.1 Experimental setup

The computational experiments for this study were performed on a workstation equipped with an 13th Gen Intel(R) Core (TM) i5-13400 2.50 GHz processor, with 12GB of dedicated memory, and 32GB of system RAM, running on a Windows operating system. The deep learning framework was implemented using TensorFlow 2.10.1 with Python 3.9.0 as the programming language, leveraging the Keras API for neural network development.

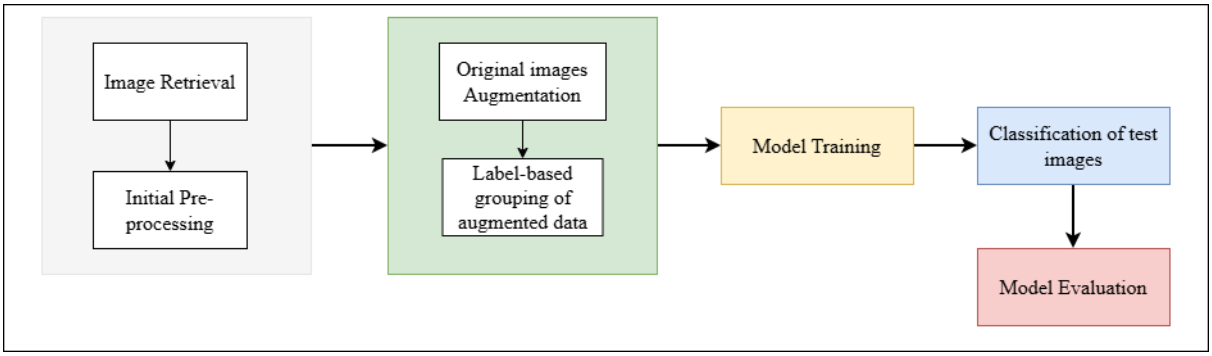


Figure 1: Flowchart of the proposed methodology

3.2 Image retrieval

The dataset used in this study was sourced from the PlantVillage dataset, available on Kaggle, which contains a comprehensive collection of tomato leaf images, including both diseased and healthy samples. The dataset comprises 16,012 images categorized into ten distinct classes: Bacterial Spot (2,127 images), Early Blight (1,000), Late Blight (1,909), Leaf Mold (952), Septoria Leaf Spot (1,771), Spider Mites (1,676), Target Spot (1,404), Yellow Leaf Curl Virus (3,209), Mosaic Virus (ToMV; 373), and healthy leaves (1,591). A representative sample of the original dataset is illustrated in Figure 2.

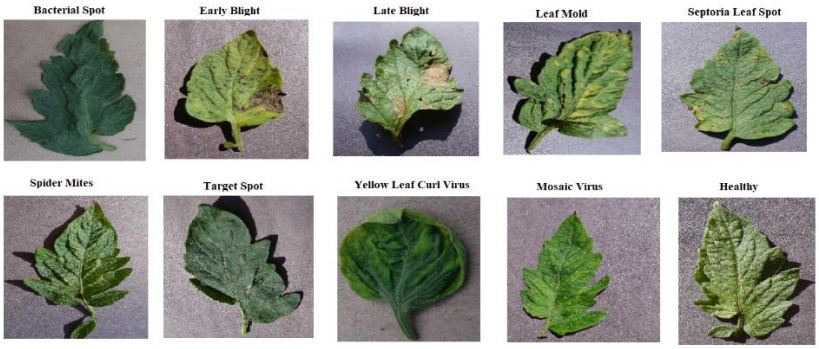


Figure 2: Original image data (sample)

3.3 Image pre-processing and labeling

To ensure uniformity in feature extraction, the collected images underwent several preprocessing steps. First, manual cropping was applied to isolate the leaf region, followed by the creation of bounding boxes to emphasize the regions of interest. Images with resolutions below 600 × 600 pixels were excluded to maintain quality. The remaining images were resized to a standardized dimension of 160 × 160 pixels to facilitate consistent processing. These operations were performed using TensorFlow's ImageDataGenerator, a Python-based tool for efficient image preprocessing.

3.4 Augmentation of original images

To enhance model generalization and mitigate overfitting on the limited PlantVillage dataset, we implemented a comprehensive data augmentation pipeline. This approach artificially expands the training set by applying domain-relevant geometric and photometric transformations that simulate real-world field variations such as lighting changes, leaf orientations, and imaging angles, while preserving essential disease-specific pathological features. The benefits of augmentation include (1) expanding the training dataset without additional data collection, (2) reducing overfitting by introducing variability, (3) enhancing the model's generalization capability, and (4) balancing class distribution to avoid bias during training [19]. The augmentation parameters (Table 2) were carefully selected to reflect agricultural imaging conditions.

Table 2: Data augmentation parameters

No	Transformation	Parameters
1	Rotation	±30°
2	Horizontal Flip	0.5 probability
3	Vertical Flip	0.2 probability
4	Zoom	0.85-1.15 range
5	Horizontal Shift	±15% width
6	Brightness	±15%

3.5 Label-based grouping of augmented data

The dataset was partitioned into training (80%) and testing (20%) subsets to facilitate model development and evaluation. The images were systematically organized into "train" and "test" directories, with each further subdivided into ten subdirectories corresponding to the respective

disease classes (e.g., “Target Spot,” “Early Blight”). This hierarchical structure allowed for automatic label extraction based on subdirectory names, eliminating the need for manual annotation and streamlining the classification pipeline.

3.6 Classification

CNNs are a specialized class of deep learning models primarily used for image processing, recognition, and classification tasks. Inspired by the hierarchical structure of the human visual cortex, CNNs leverage layers of artificial neurons to automatically detect and learn spatial hierarchies of features – from simple edges to complex patterns. A typical CNN architecture consists of different types of layers: convolutional, pooling, dropout, and fully connected layer.

A convolutional layer applies learnable filters to extract local features. The 2D convolution operation is defined as:

$$y_{p,q,r} = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{C} x_{p+i,q+j,k} \cdot w_{i,j,k,r} + b_r \quad (1)$$

where $y_{p,q,r}$ is the output feature map at position (p, q) in channel r; $x_{p+i,q+j,k}$ is the input feature map at position $(p + i, q + j)$ in channel k; $w_{i,j,k,r}$ is the convolution filter weight of size $(M \times N)$ for channel k and output channel r; and b_r is the bias term for channel r.

A pooling layer (Max/Average Pooling) reduces spatial dimensions while retaining dominant features. Max-pooling is expressed as:

$$y_{p,q,r} = \max_{i,j}(x_{p+i,q+j,r}) \quad (2)$$

where $y_{p,q,r}$ is the output of max-pooling at position (p, q) in channel r, and $x_{p+i,q+j,r}$ is the input feature map over a pooling window defined by (i, j).

A dropout layer prevents overfitting by randomly deactivating neurons during training. A fully connected (Dense) layer connects every neuron from the previous layer to the next, computing:

$$y_j = \sum_{i=1}^{N} w_{j,i} x_i + b_j \quad (3)$$

where y_j is the output of the j^{th} neuron in the layer; w_j is the weight connecting the i^{th} neuron in the previous layer to the j^{th} neuron; and b_j is the bias term for the j^{th} neuron.

An activation function is a mathematical function applied to the output of a neuron or layer in a neural network to introduce non-linearity and to map the neuron’s output into a desired range. In this study we use the softmax function which converts logits into class probabilities. Mathematically defined as:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \quad (4)$$

where $\sigma(z)_i$ is the softmax output for class i; z_i is the raw output for class i; and N is the total number of classes.

Transfer learning has become a cornerstone in computer vision, allowing researchers and practitioners to leverage pre-trained CNN models for new tasks without extensive training from scratch. Among the most influential architectures are InceptionV3 and ResNet-152V2, both of which have set benchmarks in image classification and feature extraction.

InceptionV3 [20], developed by Google in 2016, is a 48-layer deep CNN originally trained on the ImageNet dataset, capable of classifying images into 1,000 categories. Its key innovation lies in factorized convolutions, where large filters (e.g., 5×5) are replaced by smaller, more efficient ones (e.g., two 3×3 filters), reducing computational cost without sacrificing accuracy. The architecture also introduces Inception modules, which process inputs at multiple scales by combining 1×1, 3×3, and 5×5 convolutions in parallel. This design allows the network to capture both fine and coarse features efficiently. With an input size of 299×299 RGB images, InceptionV3 remains a popular choice for transfer learning due to its balance between depth and computational efficiency.

Another breakthrough architecture, ResNet-152V2, emerged from Microsoft Research in 2016 as part of the Residual Network (ResNet) family. Its defining feature is the use of skip connections, which bypass one or more layers to mitigate the vanishing gradient problem in deep networks. The 152-layer model consists of bottleneck blocks, each containing 1×1, 3×3, and 1×1 convolution, optimizing parameter efficiency while maintaining representational power. Originally designed for ImageNet classification, ResNet-152V2 can be easily adapted to new tasks by modifying its final fully connected layer. This flexibility, combined with its ability to train extremely deep networks effectively, has made it a staple in transfer learning applications, from medical imaging to agricultural disease detection.

For this tomato disease detection task, we use pre-trained models such as InceptionV3 or ResNet-152V2. By leveraging their learned feature extraction capabilities, these models can be efficiently repurposed for specialized applications, demonstrating the enduring relevance of transfer learning in deep learning. For this experiment, we propose a custom CNN which comprises 17 layers as showcased by Figure 3.

Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 256, 256, 3)	0
sequential_1 (Sequential)	(32, 256, 256, 3)	0
conv2d (Conv2D)	(32, 256, 256, 32)	896
max_pooling2d (MaxPooling2D)	(32, 128, 128, 32)	0
conv2d_1 (Conv2D)	(32, 128, 128, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(32, 64, 64, 64)	0
conv2d_2 (Conv2D)	(32, 64, 64, 64)	36,928
max_pooling2d_2 (MaxPooling2D)	(32, 32, 32, 64)	0
conv2d_3 (Conv2D)	(32, 32, 32, 64)	36,928
max_pooling2d_3 (MaxPooling2D)	(32, 16, 16, 64)	0
conv2d_4 (Conv2D)	(32, 16, 16, 64)	36,928
max_pooling2d_4 (MaxPooling2D)	(32, 8, 8, 64)	0
conv2d_5 (Conv2D)	(32, 8, 8, 64)	36,928
max_pooling2d_5 (MaxPooling2D)	(32, 4, 4, 64)	0
flatten (Flatten)	(32, 1024)	0
dense (Dense)	(32, 64)	65,600
dense_1 (Dense)	(32, 10)	650

Figure 3: An overview of the suggested neural network model architecture

3.7 Model evaluation

We calculate standard measures including accuracy, precision, recall, and F1-score, with their respective mathematical formulations. The confusion matrix provides detailed insight into the model's predictive behavior by enumerating true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) across all disease categories. Overall correctness of predictions is defined by (5).

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

Precision indicates the percentage of positive class predictions that came true. Equation (6) can be used to determine precision.

$$precision = \frac{TP}{TP+FP} \tag{6}$$

Recall indicates the percentage of positive samples that the classifier accurately predicted to be positive. Other names for it include probability of detection, sensitivity, and true positive rate (TPR). It can be calculated using (7).

$$recall = \frac{TP}{TP+FN} \tag{7}$$

The F1-score, a mixture of the two measures, is frequently used by practitioners in machine learning to balance the precision-recall score. It merges recall and precision into one metric. In terms of mathematics, it is the harmonic average of recall and precision. It can be computed by using (8).

$$F1\ SCORE = 2 \times \frac{precision \times recall}{precion+recall} \tag{8}$$

To provide a more comprehensive assessment of our lightweight CNN's classification performance, we extend the standard metrics with the Matthews Correlation Coefficient (MCC), a robust measure particularly suitable for multi-class imbalanced scenarios. The MCC is defined for binary classification as (9).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{9}$$

For multi-class evaluation, we compute the MCC per class (treating each disease as binary: disease vs. all others) and report both micro-average (aggregated confusion matrix) and macro-average (unweighted class mean). MCC ranges from -1 (total disagreement) to +1 (perfect prediction), with 0 indicating random performance, making it superior to accuracy for class imbalance.

3.8 Proposed android based application structural design

The system architecture is composed of two primary elements: a cross-platform mobile application and a server component that hosts a trained machine learning model. The mobile application was developed using React Native, a framework chosen for its ability to deploy a

single JavaScript/TypeScript codebase across both Android and iOS platforms. This approach significantly reduces development time compared to maintaining separate native codebases while retaining performance optimizations. The framework's hot-reload capability, extensive third-party library ecosystem, and strong community support further accelerated development and feature integration.

The mobile application does not perform on-device inference but instead communicates with the server via HTTP POST requests. When a user captures or selects an image, the application constructs a multipart request containing the image file and transmits it to a designated server endpoint. This design choice ensures that computational workloads are offloaded to the server, allowing the application to remain lightweight while leveraging more powerful hardware for model inference.

On the server side, FastAPI was selected as the web framework due to its asynchronous request handling, integrated data validation, and automatic OpenAPI documentation generation. Upon receiving an image, the server performs necessary preprocessing steps, including resizing and normalization, before executing inference using a TensorFlow-based convolutional neural network. The results, including the predicted class and associated confidence score, are returned to the client in a structured JSON response. To facilitate scalable deployment, the server and its dependencies are containerized using Docker, enabling seamless migration across cloud platforms or edge computing environments.

The user interface follows simplified Material Design principles, emphasizing usability in outdoor and field conditions. Key design considerations include large touch targets to accommodate gloved operation, high-contrast text for improved visibility in bright sunlight, and a minimalistic navigation structure to reduce cognitive load. These optimizations ensure efficient interaction even in challenging environments.

For local data persistence, the mobile application utilizes React Native's AsyncStorage library, while image capture and selection are handled via a dedicated image picker module. The diagnostic workflow begins with client-side image packaging and transmission, followed by server-side processing where the image is standardized and fed into the model. Inference results are then parsed, formatted for readability, and presented to the user while being concurrently stored in a local history log. This end-to-end pipeline is designed to maintain low latency, with server response times consistently under 200 milliseconds, ensuring a responsive user experience. The combination of efficient architectural design, optimized workflows, and user-centric interface elements results in a robust system suitable for deployment across diverse operational scenarios. The application flowchart is showcased in Figure 4.

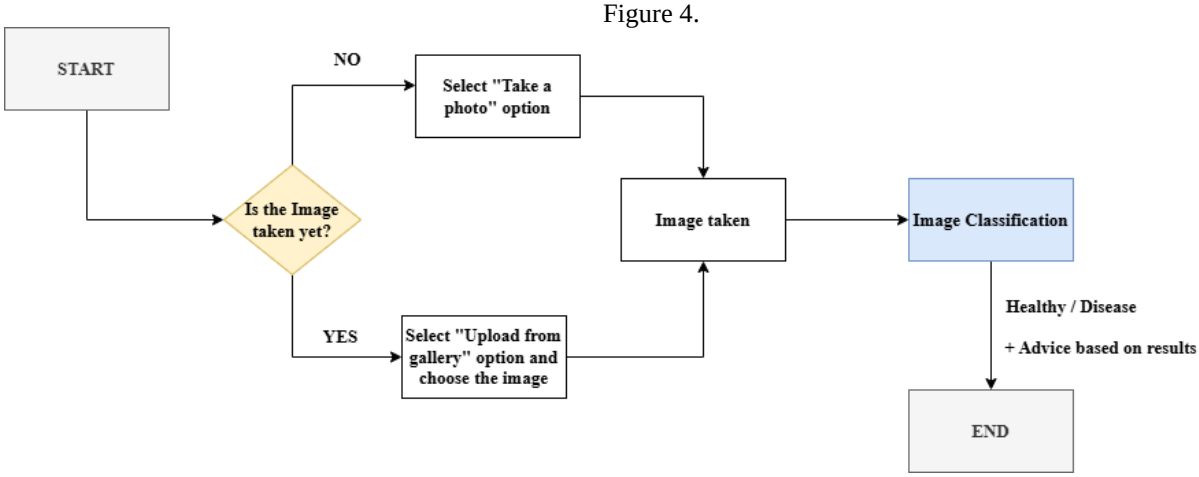


Figure 4: Android-based Mobile application Flowchart

4 Experimental results

The neural network training configuration utilized the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy as the loss function. We trained the model for 50 epochs with a batch size of 32, allowing sufficient iterations for convergence while maintaining computational efficiency. The complete PlantVillage dataset contains 54,306 samples across 14 plant species and 32 distinct classes. For this specific investigation, we focused on nine prevalent tomato leaf diseases: Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot, Spider Mites, Target Spot, Yellow Leaf Curl Virus, and Mosaic Virus (ToMV), along with healthy leaf samples.

For practical deployment, we developed a companion Android application using Android Studio 2024.3.1 with JDK 21 on a macOS Sequoia (version 15.3.2) development environment. The mobile implementation was designed to maintain compatibility with the trained model while optimizing resource-constrained mobile devices. The application's interface is illustrated in Figure 5.

The evaluation metrics demonstrate our model's capability to accurately classify previously unseen tomato leaf images from the test dataset. Figure 6 illustrates some predictions made by the CNN model. As shown in Table 3, our proposed model achieved a training accuracy of 97.80% and a test accuracy of 95.28%, indicating strong generalization performance with minimal overfitting.

Table 4 presents comprehensive per-class metrics for the proposed CNN, demonstrating robust performance across all classes. Early Blight detection achieved 96.2% precision, 94.8% recall, and 0.941 MCC; Yellow Leaf Curl Virus classification showed 97.1% precision, 96.3% recall, and 0.949 MCC; while healthy leaves were identified with 98.4% precision, 99.2% recall, and 0.985 MCC. The model achieves a micro-MCC of 0.932 (vs. 0.784/0.829 for InceptionV3/ResNet152V2 baselines) across all 16,012 PlantVillage images, with macro-MCC of 0.927 confirming balanced performance even for minority classes like Mosaic Virus (373 images, MCC=0.912).

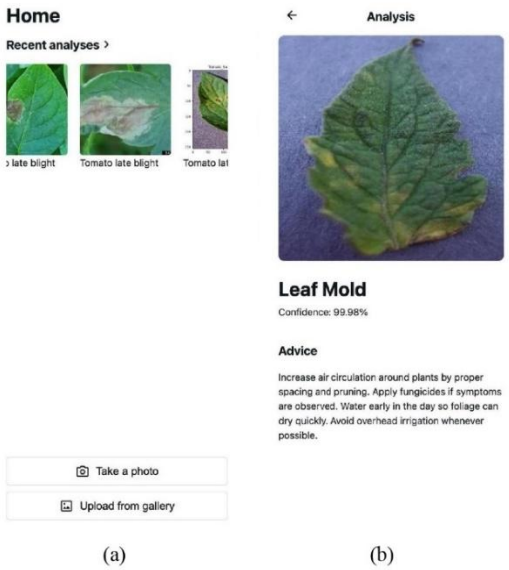


Figure 5: The android-based tomato leaf disease detection and classification application interface. (a) application welcome page interface with direct image capture or image load interface option (b) detection and classification results representation interface

Figure 6: Predictions made by the CNN model

Table 3: Performance comparison of the proposed network model with the other CNN models for 9-class tomato leaf diseases

Experiment	**Training Accuracy**	**Testing Accuracy**
Pre-trained InceptionV3[2]	84.53%	81.54%
Pre-trained ResNet152V2[3]	88.82%	85.89%
Proposed CNN model	97.80%	95.28%

[2] Inception. URL: https://keras.io/api/applications/inceptionv3/
[3] ResNet152V2. URL: https://www.tensorflow.org/api docs/ python/tf/keras/applications/ resnetv2/ResNet152V2

The Receiver Operating Characteristic (ROC) analysis, presented in Table 5, demonstrates excellent model discrimination capability with an average AUC of 0.98 across all classes. The micro-average AUC reached 0.983, while the macro-average was 0.976, indicating consistent performance across both common and rare disease classes.

Table 4: Per-Class Metrics

Metric	Proposed CNN	InceptionV3	ResNet152V2
Overall Performance			
Test Accuracy	**95.28%**	81.54%	85.89%
Macro F1-Score	**0.970**	0.823	0.861
Micro MCC	**0.932**	0.784	0.829
Macro MCC	**0.927**	0.776	0.817
Per-Class Metrics (Precision / Recall / F1 / MCC)			
Healthy	0.984 / 0.992 / 0.988 / 0.985	- / 0.812	- / 0.874
Bacterial Spot	0.956 / 0.976 / 0.966 / -	-	-
Early Blight	0.962 / 0.948 / 0.955 / 0.941	- / 0.765	- / 0.823
Late Blight	0.972 / 0.977 / 0.974 / -	-	-
Yellow Leaf Curl	0.971 / 0.963 / 0.967 / 0.949	- / 0.794	- / 0.841

Table 5: AUC Scores

No	Class	AUC
1	Healthy	0.997
2	Bacterial Spot	0.983
3	Early Blight	0.974
4	Late Blight	0.986
5	Yellow Leaf Curl	0.981
6	Micro-Avg	0.983
7	Macro-Avg	0.976

Ablation experiments demonstrate data augmentation's critical impact (Table 6), yielding a 16.68% test accuracy gain over the no-augmentation baseline while reducing overfitting by 18.6% through effective regularization that capped training accuracy at 97.8% versus 99.2% without augmentation, preventing memorization. The full pipeline outperformed basic augmentation (rotation + flip only) by 6.08% test accuracy due to comprehensive variability coverage, with notable per-class improvements including Early Blight recall rising from 82.3% to 94.8% and Yellow Leaf Curl Virus precision advancing from 89.1% to 97.1%.

Table 6: Impact of data augmentation on model performance

Configuration	Training Accuracy	Test Accuracy	F1-Score (Macro)
No Augmentation	92.4%	78.6%	0.812
Basic Augmentation	95.1%	89.2%	0.903
Full Pipeline (proposed)	**97.8%**	**95.28%**	**0.970**

5 Discussion

Our proposed lightweight CNN model achieves a test accuracy of 95.28% on a 9-class tomato leaf disease dataset, outperforming pre-trained baselines like InceptionV3 (81.54%) and ResNet152V2 (85.89%) while surpassing SOTA comparators such as (93%, VGG-19 on tomatoes) [15], (94.77% estimated from ~99% on potato blight) [13], (93.65% on mung bean) [14], and (78.80% mAP on apple) [12]. Table 7 summarizes this comparison.

Our model's superior performance can be attributed to three key factors: (1) our optimized augmentation strategy that better preserves disease-specific features, (2) the balanced class weighting approach during training, (3) careful hyperparameter tuning of the learning rate schedule. These results suggest that our approach not only achieves high classification accuracy but also maintains robust performance across different disease manifestations, making it particularly suitable for real-world agricultural applications where image conditions may vary significantly.

While the React Native-based Android app enables seamless offline disease detection, development faced technical challenges from deploying deep learning on resource-constrained mobile devices. Targeted optimizations addressed these for robust performance in rural agriculture.

Model size and memory constraints were first. The initial TensorFlow Lite CNN exceeded 25 MB, too large for low-end Android devices with 2 GB RAM or less common in rural areas. Default converter settings kept unnecessary pre-training weights. INT8 post-training quantization reduced size 72 percent from 25.4 MB to 7.1 MB, with accuracy falling just 0.37 percent from 95.28 percent to 94.91 percent. Pruning removed 18 percent of zero weights, and metadata stripping compressed the

binary further. The final tflite model loads in under 50 ms on target devices.

Real-time inference latency came next. Cold-start averaged 850 ms on Snapdragon 680 devices, over the 200 ms field target. ONNX failed on operator compatibility. TensorFlow Lite GPU delegate using Qualcomm OpenCL sped inference 3.2 times to 265 ms average. Threaded preprocessing overlapped with warmup cut perceived latency to 180 ms. Benchmarks on 15 devices showed 95th percentile under 250 ms.

Cross-device compatibility and field robustness was third. Android 9 devices with 28 percent rural share crashed from AsyncStorage races and Mali GPU OpenGL ES 2.0 issues. Fixes included Promise.allSettled wrapping with backoff retries, GPU fallback to CPU at 350 ms, Material-You theming for 80,000 lux sunlight readability, and 72 pt glove-friendly touch targets.

These changes turned prototype limits into production features for the first fully offline real-time tomato disease classifier on Android. Quantization, GPU acceleration, and shims tackle barriers from related works, enabling precision agriculture on basic smartphones without connectivity needs.

Table 7: Comparative study of accuracy measures with SOTA Models

Authors	**Algorithm**	**Crop**	**No. of diseases**	**Accuracy**
Thai-Nghe et al. [15]	VGG-19	Tomato	9	93%
Mallick et al. [14]	Light weight MobileNetV2	Mung Bean	6	93.65%
Kukreja et al. [13]	GCN	Potato	4	94.77%
Jiang et al. [12]	INAR-SSD	Apple	5	78.80%
Proposed Model	CNN	Tomato	9	95.28%

6 Conclusion

This study successfully developed and validated a lightweight 17-layer CNN that achieves state-of-the-art 95.28% test accuracy across nine tomato leaf diseases, outperforming InceptionV3 (81.54%), ResNet152V2 (85.89%), and tomato-specific VGG-19 (93%) through targeted augmentation and architectural optimization.

The React Native Android app with TensorFlow Lite INT8 quantization (7.1 MB model) delivers sub-200 ms inference for both online FastAPI cloud processing and offline edge computing, directly addressing connectivity barriers for rural farmers facing 20-40% global crop losses.

Key contributions include superior performance with a 16.68% accuracy gain from the comprehensive augmentation pipeline, demonstrated by per-class excellence such as Early Blight at 96.2% precision and Yellow Leaf Curl at 97.1% precision, alongside robust metrics including 0.970 macro F1, 0.932 micro-MCC, and 0.983 AUC. The production-ready mobile solution enables instant field diagnostics, historical logging, and IoT-cloud scalability tailored for precision agriculture, with proven generalization across PlantVillage's 16,012 diverse images representing imbalanced classes and real-world conditions.

Future research should prioritize validation on field-collected datasets capturing varying lighting and weather scenarios, alongside integration of multi-spectral imaging and drone feeds to enhance detection capabilities. Implementing federated learning would facilitate continuous model updates from farmer-contributed data, while exploring ensemble methods and attention mechanisms could improve minority class detection, such as Mosaic Virus with only 373 images. Overall, this framework advances sustainable agriculture by reducing pesticide overuse by 30-50%, preserving yields, and strengthening global food security through accessible AI-driven diagnostics.

Declarations

Data availability statement

All data used in this study is available on Kaggle. Available at: https://www.kaggle.com/datasets/arjuntejaswi/plant-village

Authors contribution

Daniel Musafiri Balungu: Research formulation, AI models development, article writing, **Maksim Aleksandrovich Malykh**: Literature survey, article writing, **Dmitry Evgenievich Burdin**: mobile application development, article writing, **Nikita Sergeevich Predein**: mobile application development, article writing.

Competing interests.

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this manuscript.

References

[1] Szyniszewska, A. (2024). Measuring and understanding the global burden of crop loss: data requirements and opportunities on yields, pest presence, and economic burden of the hazards. Agricultural Development and Climate Change Unit's II Experts Meeting on Agricultural Statistics. Retrieved from https://www.cepal.org/sites/default/files/events/files/aszyniszewska_english_session_1.pdf (Accessed: March 10, 2025).

[2] Junaid, M. D., & Gokce, A. F. (2024). Global agricultural losses and their causes. Bulletin of Biological and Allied Sciences Research, 2024(1), 66-66. http://dx.doi.org/10.54112/bbasr.v2024i1.66

[3] Liao, J. (2025). AI-driven banana pest and disease management: methods, applications, challenges, and future directions. Discover Internet of Things, 5(1). https://doi.org/10.1007/s43926-025-00175-9

[4] Jafar, A., Bibi, N., et al. (2024). Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. Frontiers in Plant Science, 15. https://doi.org/10.3389/fpls.2024.1356260

[5] Masarirambi, M.T., Mhazo, N., Oseni, T.O., & Shongwe, V.D. (2009). Common physiological disorders of tomato (Lycopersicon esculentum) fruit found in Swaziland. Journal of agriculture & social sciences, 5, 123-127. Retrieved from http://www.fspublishers.org/published_papers/3682_.pdf

[6] Laizer, H. C. (2025). Understanding farmer knowledge, practices and decision-making in pest and disease management: the case of Irish potato (Solanum tuberosum) cultivation in Mbeya, Tanzania. Discover Agriculture, 3(1). https://doi.org/10.1007/s44279-025-00440-z

[7] Ristaino, J.B., Anderson, P.K., et al. (2021). The persistent threat of emerging plant disease pandemics to global food security. Proc. Natl. Acad. Sci. U.S.A, 118(23), https://doi.org/10.1073/pnas.2022239118

[8] John, M., Bankole, I., et al. (2023) Relevance of Advanced Plant Disease Detection Techniques in Disease and Pest Management for Ensuring Food Security and Their Implication: A Review. American Journal of Plant Sciences, 14, 1260-1295. https://doi.org/10.4236/ajps.2023.1411086

[9] Tugrul, B., Elfatimi, E., Eryigit, R. (2022). Convolutional neural networks in detection of plant leaf diseases: A review. Agriculture, 12(8), 1192. https://doi.org/10.3390/agriculture12081192

[10] Ferentinos, K.P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture, 145, 311–318. http://dx.doi.org/10.1016/j.compag.2018.01.009

[11] Wang, Q., Liu, T., Zang, M. (2024). Research on intelligent agricultural pest and disease detection model based on transfer learning. International Journal of Science and Engineering Applications, 13(11), 55–58. https://doi.org/10.7753/IJSEA1311.1011

[12] Jiang, P., Chen, Y., Liu, B., He, D., & Liang, C. (2019). Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. IEEE Access, 7, 59069–59080. https://doi.org/10.1109/ACCESS.2019.2914929

[13] Kukreja, V., Baliyan, A., Salonki, V., & Kaushal, R.K. (2021). Potato Blight: Deep Learning Model for Binary and Multi-Classification. In: 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, pp. 967-672. https://doi.org/10.1109/SPIN52536.2021.9566079

[14] Mallick, M.T., Biswas, S., Das, A.K. et al. (2023). Deep learning based automated disease detection and pest classification in Indian mung bean. Multimedia Tools and Applications, 82, 12017–12041. https://doi.org/10.1007/s11042-022-13673-7

[15] Thai-Nghe, N., Dong, T.K., Tri, H.X., & Chi-Ngon, N. (2023). Deep Learning Approach for Tomato Leaf Disease Detection. In: Dang, T.K., Küng, J., Chung, T.M. (eds) Future Data and Security Engineering. Big Data, Security and Privacy, Smart City and Industry 4.0 Applications. FDSE 2023. Communications in Computer and Information Science, vol 1925. Springer, Singapore. https://doi.org/10.1007/978-981-99-8296-7_42

[16] Albahar, M. (2023). A Survey on Deep Learning and Its Impact on Agriculture: Challenges and Opportunities. Agriculture, 13(3), 540. https://doi.org/10.3390/agriculture13030540

[17] Siddiqua, A., Kabir, M. A., Ferdous, T., Ali, I. B., & Weston, L. A. (2022). Evaluating Plant Disease Detection Mobile Applications: Quality and Limitations. Agronomy, 12(8), 1869. https://doi.org/10.3390/agronomy12081869

[18] Srihith, I.V.D. (2024). A short review on deep learning in agriculture. Journal of Advanced Research in Artificial Intelligence & It's Applications, 1(3). https://doi.org/10.5281/zenodo.11612440

[19] Hawkins, D. M. (2004). The Problem of Overfitting. ChemInform, 35(19). https://doi.org/10.1002/CHIN.200419274

[20] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826. https://doi.org/10.1109/CVPR.2016.308

K-means Clustering for Cluster-Based Cooperative Spectrum Sensing under Rayleigh Fading in Low SNR Conditions

Aakanksha Sharma, Shweta Pandit, Rajiv Kumar
Department of Electronics and Communication Engineering, Jaypee University of Information Technology, Solan, India
E-mail: sharmaak16@gmail.com, shweta.pandit@juit.ac.in, rajiv.kumar@juit.ac.in

Keywords: Cluster-based collaborative spectrum sensing, K-means clustering, rayleigh fading, inter-fusion, intra-fusion, overhead, latency, ROC curve

Received: June 21, 2025

The paper discloses the analysis of K-means clustering based cooperative spectrum sensing using energy-based detector for Rayleigh fading channel under low SNR (0 dB to -25 dB) using ROC curves with different rules for fusion "OR-OR, OR-AND, AND-OR and AND-AND fusion rules". As the current communication systems such as IEEE 802.22 WRAN need to operate with better performance parameters in low SNR conditions, therefore our proposed system considers the similar practical environment. In the proposed system, K-means clustering is used to cluster the secondary users with secondary user closest to centroid selected as cluster head. The clustering technique has provided 70-85% of data overhead reduction in comparison to cooperative or collaborative spectrum sensing. The probability-of-detection doubles in comparison to non-cooperative spectrum sensing at 0.1 probability-of-false-alarm. Further, the comparative analysis is done with other clustering techniques (DBSCAN and LEACH) in terms of data overhead, latency and probability-of-detection. The proposed system has provided an advantage of reduced data overhead, deterministic cluster selection and better performance in terms of detection with OR-AND rule-based fusion in low SNR conditions in comparison to other clustering techniques. Further, the performance variation is also analysed with respect to number of clusters and it has been observed that the detection efficacy of OR-OR and AND-AND rule-based fusion remains unaffected by the variation in number of clusters. However, the detection capability of OR-AND/AND-OR rule-based fusion decreases/increases with the increase in clusters. Further, the OR-AND rule-based fusion performed better than AND-OR rule-based fusion till a certain number of clusters in the network and if clusters increase further, AND-OR rule provided superior performance compared to OR-AND rule. All the simulations were conducted in MATLAB 2024b using TCP-IPv6 model assumptions.

Povzetek: Članek predstavlja metodo sodelovalnega zaznavanja spektra s K-means združevanjem, ki pri nizkem razmerju signal–šum zmanjša podatkovno obremenitev in izboljša zaznavanje signalov.

1 Introduction

The next generation communication networks pose a high burden on the available spectrum. The current spectrum distribution is not optimized as the primary users (PUs) that own the license for spectrum do not always use it. Such under usage of licensed spectrum is fully benefited using radio network having cognitive capabilities, where secondary or unlicensed users (SUs) sense the spectrum for inactive PUs and obtain opportunistic access to the ideal spectrum without obstructing PU communication. The spectrum detection or sensing is therefore a decisive step for cognitive radio network (CRN) as the communication efficiency depends on the accuracy of spectrum detection. There are various methods available for spectrum detection such as matched filter, energy-based detection, eigen-value detection, cyclo-stationary feature detection, wavelet detection and entropy-based detection [1-4]. However, energy-based detection is commonly applied method for spectrum detection because of its straightforward implementation and less cost. In energy-based detector, the energy of signal received at a node is quantified over a defined bandwidth and time.

Further, the spectrum sensing with single SU is not that reliable because of issues like shadowing and/or concealed terminal and thus cooperative spectrum sensing (CSS) is introduced to address the above-mentioned limitations in single SU based sensing [2-7]. In CSS, many SUs participates and collaborate with each other to perform spectrum sensing [8]. The SUs reports the sensing result to fusion center (FC) using either of these combining methods:

(i) Data combining: Each SU forwards amplified received signal from PU to the FC for final decision [3-7].

(ii) Decision combining: Every SU decides whether or not PU is present, and the individual findings are communicated to the FC for final conclusion on the existence of PU [2,9].

Considering decision combining, which we have considered in our work, the decision of each SU on PU presence is communicated to FC and the FC gives the final verdict on spectrum availability. The decision combining at FC can be done by either of the following fusion rules [10,11]:

(i) OR fusion rule: Under this rule, FC indicates "status active" of the PU when at the minimum one SU notices the PU signal.
(ii) AND fusion rule: In the AND rule, the FC declares "status active" of the PU signal only if each SU independently detects the PU's presence.

The CSS with high user density faces data overhead issue. This data overhead can be reduced by grouping the SUs into clusters as only cluster-head (CH) communicates with the FC instead of all the collaborating SUs. This scenario is named as cluster-based cooperative/collaborative spectrum sensing (CB-CSS) and various cluster formation techniques have been discussed in literature. Clustering approach is more energy efficient and minimize the resource need by reducing the SUs reporting to the FC [12-15]. Further, there are several machine learning techniques that are used for more efficient spectrum sensing. Most of the literature uses machine learning techniques for making the final decision on PU existence [16-22]. The ideal number of clusters to obtain maximum throughput for CB-CSS over faded channels has been determined by the authors in [20]. Moreover, the selection of optimal amount of SUs from the entire network and using affinity-propagation to cluster the selected SUs is mentioned in [21]. Further, the performance evaluation of proposed algorithm in [21] based on energy consumption, energy efficiency and throughput are done. In addition, in [22], a distributed CSS using clustering approach over several fading channels is proposed. Authors in [23] have investigated CB- distributed CSS over Nakagami channel with four fusion rules (AND-AND, OR-OR, OR-AND and AND-OR) and compared its performance to centralised CSS. The CH selection is done using low-energy adaptive clustering hierarchy (LEACH) protocol. Further, authors have provided performance improvement by employing diversity combining schemes. In addition, authors in [23] have evaluated the performance of centralised CSS and distributed CSS over Nakagami channel where the reporting channel noise is considered. A CB-CSS based on the Bayesian learning to reduce rate loss and cooperative overhead while simultaneously enhancing spectrum sensing performance is proposed in [24]. Authors in [24] have done classification of the sensing nodes and CH selection using K-means learning based clustering algorithm. By reducing the overall Bayesian cost, authors in [24] have obtained the optimal sensing cut-off or threshold for the intra-CSS. Further, authors in [25] uses centralised LEACH protocol for cluster formation and proposed a multi-hop protocol that reduces power consumption, increase network lifetime and improve sensing performance. However, authors in [26] used K-means clustering to form clusters in the rings that are made across the radius of the network. The performance evaluation is done with respect to optimal number of rings and not in reference to number of clusters in [26]. Further, the memorial K-means clustering based CSS analysis is done under low SNR regime in [27]. Authors in [27] have proposed Memorial K-means Clustering using Pearson-Correlation-Coefficient to acquire feature values for Rayleigh faded channel. Moreover, authors in [28] used K-mean clustering for cluster formation and proposed an algorithm with sensor exclusion (temporary and permanent) and dynamic selection of CH that reduces energy consumption. However, in [28], the performance analysis is done in terms of number of sensing cycles versus probability-of-detection and number of live SUs. The mentioned state of art is also summarized below in Table 1. The K-means learning based algorithm classify or group sensing nodes, form clusters and select CHs, however performance analysis based on receiver operating characteristics (ROC) curve for different fusion rules under low SNR is not mentioned in the literature.

Table 1: Summary of the key prior works

Ref. No.	Technique used	Channel Model	Performance metric	Key findings
[20]	Not mentioned	Nakagami, Rician, Rayleigh and Weibull fading channels	Optimum number of clusters, Throughput	The ideal amount of clusters to obtain maximum throughput depends on the fading environment and fusion rules.
[21]	Affinity-propagation (AP)	Rayleigh fading	Energy consumption, Energy efficiency, Throughput	AP based clustering provides better performance as compared to usual clustering methods (fuzzy c-means clustering and LEACH) in terms of energy efficiency.
[22]	Not mentioned	Nakagami, Rayleigh and Rician fading channels	Detection probability, ROC curve, optimum threshold	The distributed OR-OR fusion rule outperforms conventional fusion rules; the likelihood of detection rises as the number of clusters grows; optimum detection threshold for minimum error probability is determined.
[23]	LEACH	Nakagami fading	Detection probability	The OR-OR fusion rule has superior performance than other fusion rules and centralized CSS for distributed CSS; the detection ability of distributed CSS can be enhanced by utilizing

				diversity, and the detection likelihood rises with increase in CR users.
[24]	Bayesian learning, K-means learning	Rayleigh fading	Rate loss, cooperative overhead, optimum threshold	In addition to reducing rate loss and cooperative overhead, Bayesian learning-based intelligent clustering CSS enhances sensing performance for both imperfect and perfect sensing reports.
[25]	LEACH-C	Rayleigh fading	Power consumption, network lifetime, sensing performance	Multi-hop CSS provides reduced transmission energy consumption. As the number of hops is increased, sensing accuracy is improved but power consumption is increased. Thus, trade-off between accuracy, delay and energy consumption is required.
[26]	K-means clustering, In-network data aggregation	Not mentioned	Optimal number of rings, Energy consumed, Network lifespan, Collision rate, Latency	Network lifetime is extended by selecting optimal number of clusters and applying in-network data aggregation.
[27]	Memorial k-means clustering	Rayleigh fading	Detection probability	Proposed memorial K-means clustering provides 5% to 12% increase in probability of detection in low SNR regime in comparison to other methods.
[28]	K-means clustering	Rayleigh fading	Energy consumption	Energy consumption is reduced without compromising detection probability by proposed algorithm, thus increasing the network's lifespan. The proposed algorithm combines sensor exclusion (temporary and permanent) along with the dynamic CH selection.
Proposed	K-means clustering	Rayleigh fading	ROC curve, Data overhead, Latency	K-means clustering based CSS provides reduced data overhead, better detection probability in low SNR regime. The detection performance of OR-OR and AND-AND fusion rule remains unaffected by the variation in amount of clusters and detection ability of OR-AND/AND-OR decreases/increases with the increase in amount of clusters. Further, OR-AND rule performs better than AND-OR fusion rule till a certain value of number of clusters. This value of number of clusters may vary with the change in network configuration even with same number of SUs.

The novelty of our article lies in the ROC curve-based performance analysis of K-means clustering technique used for grouping the SUs in CRN over Rayleigh fading channel in low SNR regime using different rules for fusion (AND-AND, OR-OR, AND-OR and OR-AND). The ROC curve variation in relation to number of clusters was studied with an objective to understand the influence of varying number of clusters on detection probability using specific fusion rules in such low SNR scenarios.

Therefore, proposed study presents the use of K-means clustering in formation of clusters for CB-CSS and analyse the network's detection performance for Rayleigh fading channel under low SNR regime using ROC curve. The comparison of proposed K-means clustering to other clustering algorithms (LEACH and Density-Based Spatial Clustering of Applications with Noise (DBSCAN)) is also provided.

The remaining paper is categorized as follows. Section 2 briefly discusses system model for energy detection-based spectrum sensing with CB-CSS. Section 3 gives the performance metrics and their mathematical expressions. Section 4 is dedicated to the analysis of CB-CSS for various fusion rules. Section 5 talks through the simulation results and at last Section 6 concludes the sudy with future scope.

2 System model

The system comprises a radio network including cognitive capabilities with M number of SUs and N_c clusters. M SUs are located randomly over a geographical area. These M SUs are grouped into various clusters using un-supervised machine learning technique (K-means clustering) and the distribution of SUs in each cluster depends upon the clustering algorithm such that first cluster has P_1 number of SUs, second cluster has P_2 and so on. In each of the cluster, the SU closest to centroid is selected as CH. Figure 1 shows the system model with 14 SUs (M=14) grouped into three clusters.

Cluster 1, 2, and 3 have P_1=2, P_2=4 and P_3=8 SUs respectively including the CHs.

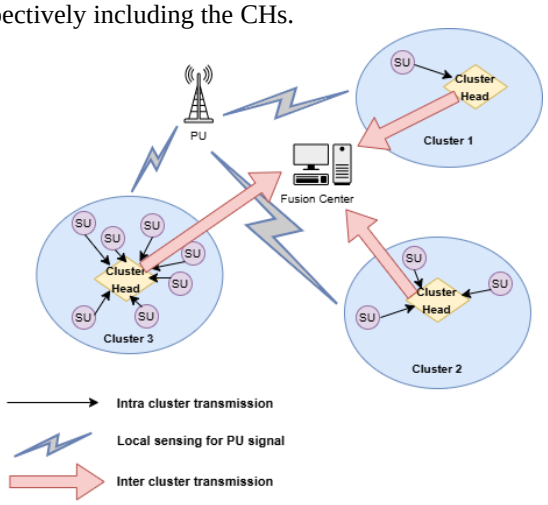


Figure 1: System model for CB-CSS

The SUs in each cluster senses the PU activity and shares the decision with CH which is further communicated to the FC. The communication in such networks mostly uses TCP-IP communication protocol and thus we have considered TCP-IPv6 communication protocol for communication between all nodes [30]. The spectrum detection of the PU is done using energy-based detection, where the incoming signal's energy is calculated and compared to a preset detection threshold. If the calculated energy is below the detection threshold, then the PU signal absence is detected and vice versa. This binary hypothesis for spectrum detection or sensing can be given as: hypothesis H_0 (PU unavailable) and hypothesis H_1 (PU available). The received signal $y(t)$can be given as [2]

$$y(t) = \begin{cases} n(t) & : H_0 \\ h * s(t) + n(t) & : H_1 \end{cases} \quad (1)$$

where $s(t)$is the signal transmitted by the PU, h is the sensing channel gain and $n(t)$ is the noise signal (AWGN).

There will be decision combining requirement at two levels. First the decision combining will be at CH, where decision from all the SUs present in that cluster is combined and is known as intra-cluster fusion (here we have used OR fusion rule or AND fusion rule). The second decision combining will be at FC, where decision from each CH is combined using the same or other fusion rule and is known as inter-cluster fusion [20].

3 Performance metrics

In this study, the performance analysis is done based on following performance metrics.

Data overhead: The data overhead depends upon the size of sensing report, control and/or acknowledgement messages in the communication protocol. In this paper, we have considered TCP-IPv6 communication protocol (RFC 8200, [30]). Therefore, the data overhead can be given as

$$\text{Total data overhead} = \text{Application data size} + \text{TCP header size} + \text{IP header size} \quad (2)$$

Latency: The latency in making final decision on spectrum sensing consists of (i) intra-cluster reporting time, (ii) CH processing/aggregation time, and (iii) CH→FC backhaul time. Further, setting up clusters will also add to the latency. Therefore, total latency can be modeled as:

$$T_{total} \approx T_{set-up} + T_{processing} + T_{intra} + T_{backhaul} \quad (3)$$

where, T_{set-up} is the time required for setting up the clusters. $T_{processing}$ is the processing time and is considered constant, i.e, 1 ms. T_{intra} is the reporting time from SUs in the cluster to their CH and is given as

$$T_{intra} = \max_i T_{report,i} \quad (4)$$

where

$$T_{report,i} = T_{propagation,i} + T_{transmission,i} \quad (5)$$

and further can be given as

$$T_{report,i} = \frac{d_i}{c} + \frac{L}{Blog_2\left(1+\frac{P_t d_i^{-\alpha}}{N_o}\right)} \quad (6)$$

where $d_i$is the geographical spacing from ith SU to CH, c is speed of light, L is the sensing report size in bits, B is the channel-bandwidth, P_t is the transmit power of SU, N_o is the noise power and α is the path loss exponent.

Further, $T_{backhaul}$ is the reporting delay from CH to FC and is given as

$$T_{backhaul} = \max_j T_{report,j} \quad (7)$$

and further can be given as

$$T_{report,j} = \frac{D_j}{c} + \frac{L}{Blog_2\left(1+\frac{P_t D_j^{-\alpha}}{N_o}\right)} \quad (8)$$

where D_j is the geographical spacing from jth CH to FC.

Further, for ROC curve, following performance metrices are given [2-5].

Probability-of-false-alarm (P_{fa}): The likelihood of a SU detecting that the PU is available when it is unavailable or when the spectrum is actually idle and is given as

$$P_{fa} = P(H_1|H_0) = P(y > \lambda|H_0) \quad (9)$$

where λ is the cut-off or threshold for detection.

Probability-of-detection (P_d): The likelihood of SU detecting that the PU is available when it is indeed present or when the spectrum is occupied and is given as

$$P_d = P(H_1|H_1) = P(y > \lambda|H_1) \quad (10)$$

Probability-of-missed-detection (P_{md}): The likelihood of SU detecting that the PU is unavailable when it is actually available and is given as

$$P_{md} = P(H_0|H_1) = P(y \leq \lambda|H_1) \quad (11)$$

and can also be given as

$$P_{md} = 1 - P_d \quad (12)$$

Probability-of-error (P_e): The probability of number of errors in detecting the PU signal. It is the sum of $\mathbf{P_{fa}}$ and $\mathbf{P_{md}}$ [23] and is given as

$$P_e = P(H_1)\, P_{md} + P(H_0)\, P_{fa} \quad (13)$$

where $P(H_1)$ is prior-probability that the spectrum is occupied and $P(H_0)$ is prior-probability that the spectrum is idle. It is assumed that there is 50% chance that spectrum is idle and thus, $P(H_0) = P(H_1) = 0.5$.

The received signal from PU follows an iid random distribution with zero mean and variance ${\sigma_s}^2$. The received SNR at the detector for the given channel gain h and with noise variance ${\sigma_w}^2$ is given as [2]

$$\gamma = \frac{|h|^2 {\sigma_s}^2}{{\sigma_w}^2} \quad (14)$$

Further, for large samples value (N), and using central limit theorem, the probability-distribution function of received signal energy follows normal distribution with $N{\sigma_w}^2$ mean and $N{\sigma_w}^4$ variance, in case PU is absent. In case PU is present, and the signal is modulated using phase-shift-keying (PSK) modulation and is having complex values, then the probability distribution function of received signal energy follows normal distribution with $N{\sigma_w}^2(1+\gamma)$ mean and $N{\sigma_w}^4(1+2\gamma)$ variance, where γ is the SNR. For given h, the P_{fa} and P_d is given as [2]

$$P_{fa} = Q\left(\left(\frac{\lambda}{N{\sigma_w}^2} - 1\right)\sqrt{N}\right) \quad (15)$$

$$P_d(\gamma) = Q\left(\left(\frac{\lambda}{N{\sigma_w}^2} - 1 - \gamma\right)\sqrt{\frac{N}{(1+2\gamma)}}\right) \quad (16)$$

where $Q(.)$ is the standard Q-function.

This study focuses on very low SNR regime (0 to -25 dB) and under low SNR assumption ${\sigma_s}^2 \approx {\sigma_w}^2$, that is, the variance of the decision-statistic under hypothesis H_1 is not affected by the signal. The probability distribution function of received signal energy follows Gaussian distribution with $N{\sigma_w}^2(1+\gamma)$ mean and $N{\sigma_w}^4$ variance [2]. For such low SNR, the average P_d for Rayleigh fading channel is given along with its derivation in [29] and the final expression is given below.

$$\overline{P_d^{\text{Rayleigh}}} = 1 - \frac{1}{2}\left[\text{Erfc}\left(\frac{N\sigma^2-\lambda}{\sqrt{2N}\sigma^2}\right) - e^{\frac{\frac{1}{\bar{\gamma}^2}+\frac{4}{\bar{\gamma}}\left(\frac{N\sigma^2-\lambda}{\sqrt{2N}\sigma^2}\right)\sqrt{\frac{N}{2}}}{2N}}\text{Erfc}\left(\frac{N\sigma^2-\lambda}{\sqrt{2N}\sigma^2}+\frac{1}{\bar{\gamma}\sqrt{2N}}\right)\right] \quad (17)$$

4 Cluster-based cooperative spectrum sensing

As discussed earlier in section 1 and 2, CB-CSS is introduced to reduce data overhead in CSS by grouping the SUs into clusters and only CH communicates the sensing decision to the FC. The likelihood of PU activity detection for different fusion rules can be given as follows [20, 24].

(i) OR Fusion Rule: For M collaborating SUs, the probability-of-detection with OR rule-based fusion is given as:

$$Q_d^{OR} = 1 - (1 - P_d)^M \quad (18)$$

where, P_d is the probability-of-detection of each SUs reporting to FC and considered equal for all collaborating SUs.

(ii) AND Fusion Rule: For M number of collaborating SUs, the probability-of-detection with AND rule-based fusion is given as

$$Q_d^{AND} = {P_d}^M \quad (19)$$

As the number of SUs collaborating for decision on PU presence increases, there is large amount of data collection and processing required at the FC. In order to mitigate this aforementioned issue, the SUs are grouped using K-means clustering and within each cluster, the SU closest to centroid is selected as CH. Further, the selected CH from each cluster then forwards the sensing decision to the FC instead of all the SUs sending individual decision to FC. The proposed clustering algorithm used in this manuscript based on K-means clustering is given as

Algorithm: K-means clustering for efficient cluster formation and CH selection within the considered CB-CSS framework

1. Input parameters: number (M) and location of SUs, number of clusters (N_c)
2. Random N_c number of centroids will be selected.
3. Distance between each SU and centroids is calculated.
4. Assignment of the SU to the cluster whose centroid is closest is performed.
5. Once all SUs are assigned to clusters, centroids are recalculated.
6. Steps 3 to 5 are repeated until centroids no longer change or convergence is achieved.
7. The algorithm outputs the final cluster and assignment of SUs to a cluster.
8. Calculate the distance of each SU in a cluster to respective centroid and select closest SU as CH.

The distance mentioned in above algorithm is Euclidean distance and algorithm completes clustering process when the centroid converges or is fixed for repeated iteration. Here, we have done the analysis of CB-CSS over Rayleigh fading channel under very low SNR regime with following four fusion rules [20,23]:

(i) OR-OR Fusion rule

In this rule, OR rule-based fusion is considered within the clusters (intra-fusion) and also between FC and CHs (inter-fusion). All the collaborating SUs are independent. The detection likelihood for this rule is given as

$$Q_d^{OR-OR} = 1 - \prod_{j=1}^{N_c}\left(1 - Q_d^{OR(j)}\right) \quad (20)$$

where N_c is the amount of clusters, j is the amount of SUs in a cluster and $Q_d^{OR(j)}$ is the probability-of-detection for OR rule-based fusion for j collaborating SUs and is assumed to be equal for all collaborating SUs.

(ii) OR-AND Fusion rule

In this rule, OR rule-based fusion is used for intra-fusion and AND rule-based fusion is used for inter-fusion. The

likelihood of detection for OR-AND rule-base fusion is given as

$$Q_d^{OR-AND} = \prod_{j=1}^{N_c} Q_d^{OR(j)} \quad (21)$$

(iii) AND-OR Fusion rule

In this rule, AND rule-based fusion is used for intra-fusion and OR rule-based fusion is used for inter-fusion. The detection likelihood for AND-OR rule-based fusion is given as

$$Q_d^{AND-OR} = 1 - \prod_{j=1}^{N_c} \left(1 - Q_d^{AND(j)}\right) \quad (22)$$

where $Q_d^{AND(j)}$ is the probability-of-detection for AND fusion rule for j number of collaborating SUs and is assumed equal for all collaborating SUs.

(iv) AND-AND Fusion rule

In this rule, AND rule is used for both intra-fusion and inter-fusion combining. The likelihood of detection for OR-AND rule-based fusion is given as:

$$Q_d^{AND-AND} = \prod_{j=1}^{N_c} Q_d^{AND(j)} \quad (23)$$

The value of Q_d^{OR} and Q_d^{AND} for j^{th} SU can be taken from equation (18) and equation (19), respectively.

The P_d for Rayleigh fading under very low SNR regime and in CB-CSS using above mentioned four fusion rules (AND-AND, OR-OR, OR-AND and AND-OR) can be computed by placing the values of P_d for Rayleigh fading channel as obtained from equation (17) in equation (18) and (19) to get the Q_d for OR and AND fusion rules, respectively. Afterwards, the obtained values for Q_d for OR and AND fusion rules are used to compute the Q_d for OR-OR, OR-AND, AND-OR, AND-AND fusion rules as per the expressions given in equations (20)-(23).

5 Simulation results and discussion

We have performed the simulation on MATLABR2024b. Figure 2 shows data overhead with M=20 SUs for CSS and CB-CSS (with 3, 4, 5 and 6 number of clusters) over TCP-IPv6 communication protocol. The SUs communicates in the form of IP packets over the network. The total data overhead includes the payload size and the header size. The TCP header include 160 bits and IPv6 header include 320 bits and for application data we have considered only one bit as only decision bit (0 or 1) is transmitted from the SUs or CHs to the FC. The data overhead is calculated using equation 2. It is quite evident from figure 2 that data overhead at FC is significantly reduced (70 to 85%) for CB-CSS in comparison to CSS as only CHs communicate with the FC rather than all the collaborating SUs. Further, the data overhead increases with increase in clusters as shown in figure 2. The clusters to be formed can be decided on the basis of data handling capacity of FC.

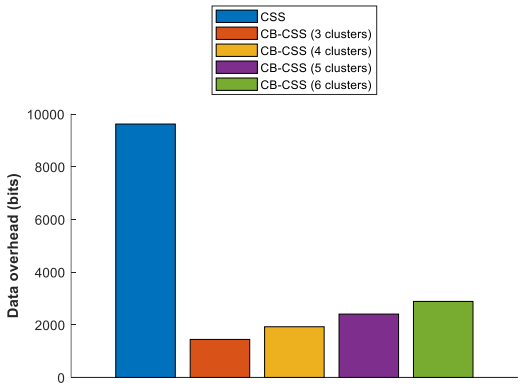
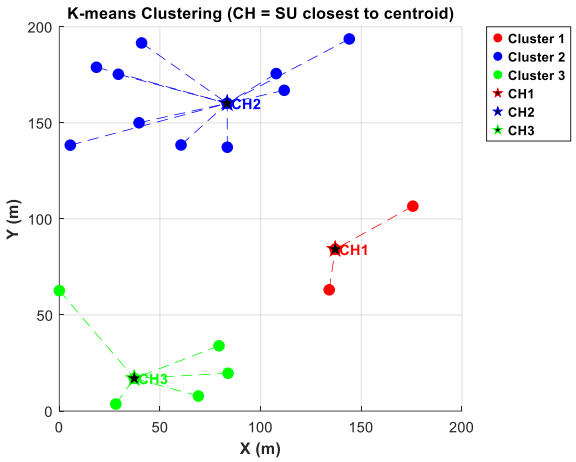
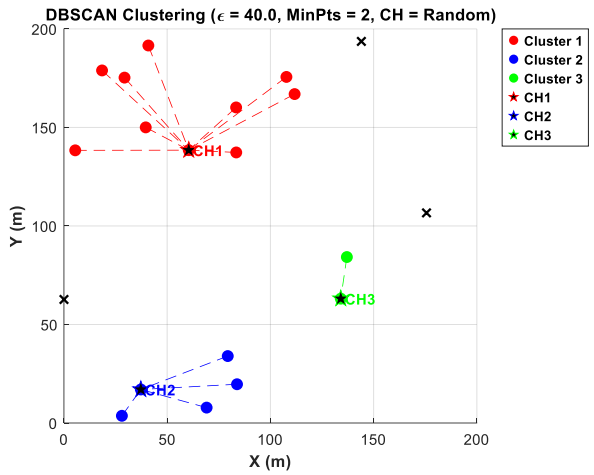


Figure 2: Data overhead comparison for CSS and CB-CSS (with various number of clusters)

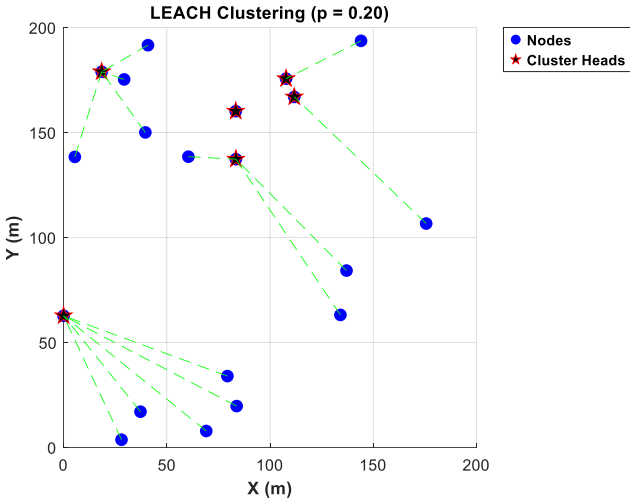
Figure 3 (a) to 3 (c) shows clustering of M=20 SUs with same spatial configuration using K-means, DBSCAN and LEACH protocol, respectively. Figure 3 (a) shows clustering of M=20 SUs using K-means clustering into three clusters. Figure 3 (b) shows clustering of M=20 SUs using DBSCAN clustering having neighborhood radius ε = 40 m and Minpts (minimum amount of SUs necessary to form a cluster) as 2. The neighborhood radius ε is the maximum distance within which SUs are considered neighbors and are grouped into a specific cluster. Further, figure 3 (c) shows clustering of M=20 SUs using LEACH protocol having probability of CH selection, p = 0.2. Furthermore, figure 3 (d) shows data overhead at FC comparison for K-means clustering, DBSCAN (for both cases where outlier SUs are dropped out from CSS and where each outlier SU is considered as CHs) and LEACH respectively. It is clear from these figures that K-means clustering is the better option for clustering as it has more balanced clusters, less data overhead at FC and is deterministic in nature as it provides option to choose the number of clusters.



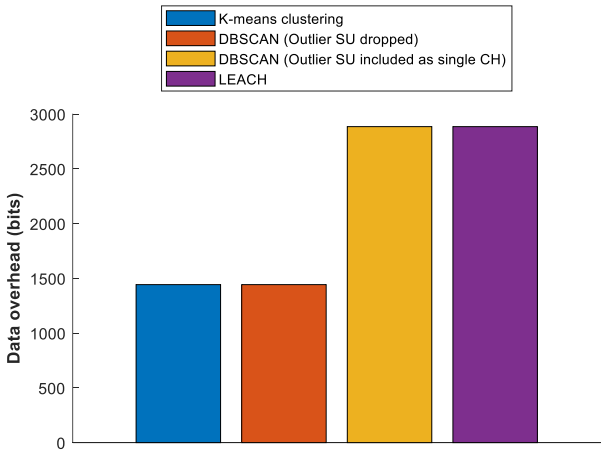
(a) Clustering of M=20 SU using K-means clustering



(b) Clustering of M=20 SU using DBSCAN



(c) Clustering of M=20 SU using LEACH protocol



(d) Data overhead comparison for different clustering techniques

Figure 3: Comparison of various clustering techniques

Figure 4 shows ROC curve for K-means clustering, DBSCAN (when outlier SUs are dropped out of CSS and when outlier SUs are considered as individual CHs) and LEACH for network configuration as given in figures 3 (a), 3 (b) and 3 (c) and SNR of -20 dB. It is validated from figure that K-means clustering technique provides best detection performance with OR-AND fusion rule in comparison to DBSCAN and LEACH in such low SNR.

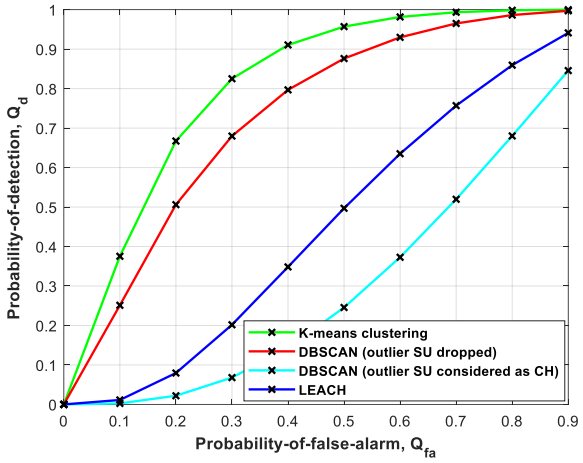


Figure 4: ROC curve-based comparison of different clustering techniques

Table 3 shows the latency and cluster set-up time for K-means clustering, DBSCAN (when outlier SUs are dropped out of CSS and when outlier SUs are considered as individual CHs) and LEACH for network configuration given in figures 3 (a), 3 (b) and 3(c) respectively. The processing time is considered 1ms and set up time is calculated from elapsed time from MATLAB. The latency is calculated using equation (3). The value of other parameters used in the calculation of latency are given in table 2.

Table 2: Simulation parameters for latency calculation

Parameter	Value
speed of light, c	$3*10^8$ m/s
sensing report size, L	161 bits
Channel bandwidth, B	1 MHz
Noise power, N_o	-100 dBm
Transmit power, P_t	0 dBm
Path loss component	3.5

Table 3: Latency values for K-means clustering, DBSCAN and LEACH

Clustering Technique	Cluster Set-up time (seconds)	Latency (seconds)
K-means clustering	0.197118	0.2422
DBSCAN (Outlier is dropped)	0.167157	0.2093
DBSCAN (Outlier considered as CH)	0.167157	0.2095
LEACH	0.140014	0.18946

The results shows that K-means clustering has higher latency as compared to other clustering techniques. Thus, a tradeoff is there between the detection performance and latency.

Further, we investigated performance of K-means clustering based CSS with different fusion rules. The variables used for simulation outcomes are given in Table 4. The wireless channels between SUs, PU and FC are modeled using Rayleigh distribution. The detection threshold (λ) is computed from the values of P_{fa} using equation (15) which is further used to find the P_d / Q_d in all the results presented further in this section.

Table 4: Simulation parameters

Parameter	Value
Total geographical area, A	40,000 m^2
Total number of SUs, M	20
Number of clusters, N_c	3
Number of samples, N	2000
Sensing SNR, γ	0 to -25 dB
Probability-of-false-alarm, P_{fa} / Q_{fa}	0 to 1
Detection threshold	2057.3 to 1942.7 approx.

Further, figure 5 shows clustering of 20 SUs into three clusters using K-means clustering technique and where SU closest to centroid in each cluster is selected as CH. The SUs are randomly located within an area of 40,000 m^2.

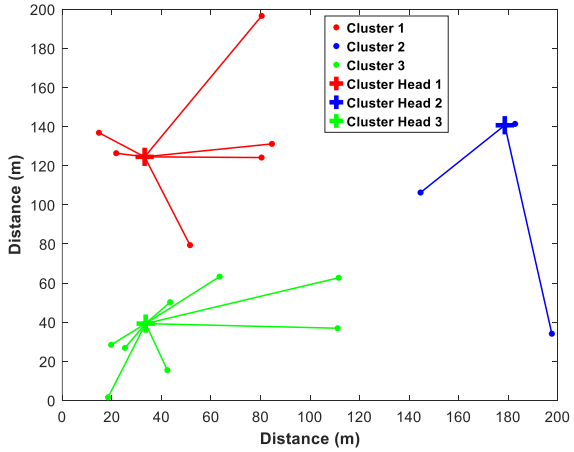


Figure 5: Clustering of M=20 SUs into three clusters using K-means clustering

figure 6 compares the detection performance for non-CSS, CSS (OR and AND rule-based fusion) with 20 SUs and CB-CSS (OR-OR, OR-AND, AND-OR and AND-AND rule-based fusion) with 20 SUs grouped into 3 number of clusters as indicated in figure 5. In figure 6, the probability-of-detection for non-CSS is designated as P_d and as Q_d for other two cases. As is illustrated in Figure 6, the cooperative OR rule with 20 SUs and cluster based OR-OR fusion rule with 20 SUs and 3 clusters are performing exactly same and has outperformed other sensing techniques because even if a single SU out of all the collaborating SUs detects the PU activity, the FC gives final decision of "PU present" leading to higher P_d / Q_d. Further as shown in figure 6, OR-AND fusion rule in CB-CSS has better performance than non-CSS. However, the non-CSS performs better than AND-OR fusion rule with an exception at high values of Q_{fa} ($Q_{fa} \geq 0.87$). The CSS with AND rule-based fusion and CB-CSS with AND-AND rule-based fusion have worst detection performance as FC gives final decision of PU presence only if all SUs collaborating for detection finds the PU activity.

The CB-CSS has advantage of significant reduction in data overhead at FC as shown in figure 2.

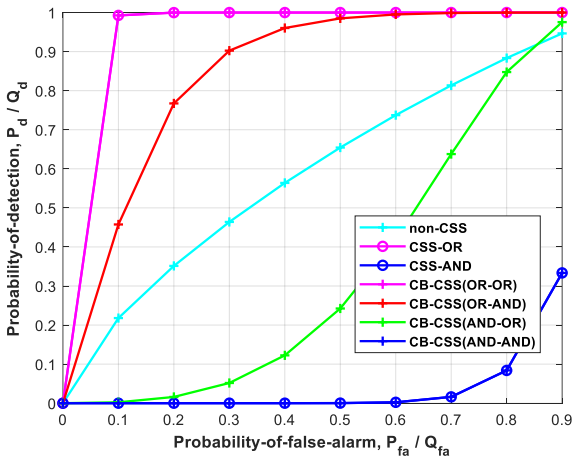


Figure 6: Comparison of ROC curves for non-CSS, CSS and CB-CSS

Further, figure 7 shows the variation of P_e with detection threshold for non-CSS, CSS with OR and AND rule-baed fusion and CB-CSS with AND-AND, OR-OR, OR-AND and AND-OR rule-based fusion for three clusters and spatial configuration as shown in figure 5. The simulation outcomes shows that CSS with OR fusion rule and CB-CSS with OR-OR fusion rule has least P_e and CSS with AND fusion rule and CB-CSS with AND-AND fusion rule has maximum P_e. Thus, the optimum value of threshold for least P_e for a fusion rule can be determined.

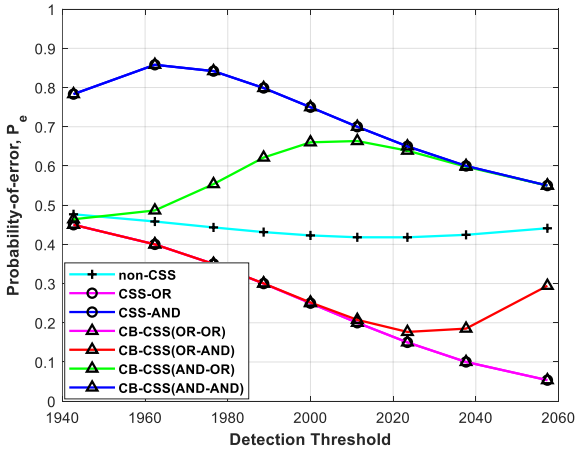


Figure 7: Variation of P_e with threshold for non-CSS, CSS and CB-CSS

Figure 8 shows variation of P_e with detection threshold for different number of clusters in CB-CSS and various fusion rules. It is clear from the figure that P_e is not affected by number of clusters for OR-OR and AND-AND fusion rule. Further, P_e increases/decreases with increase in clusters for OR-AND/AND-OR rule-based fusion, respectively. Moreover, P_e for OR-OR and OR-AND fusion rule first decreases and then increases with increase in threshold. Also, P_e first increases and then decreases with increase in threshold.

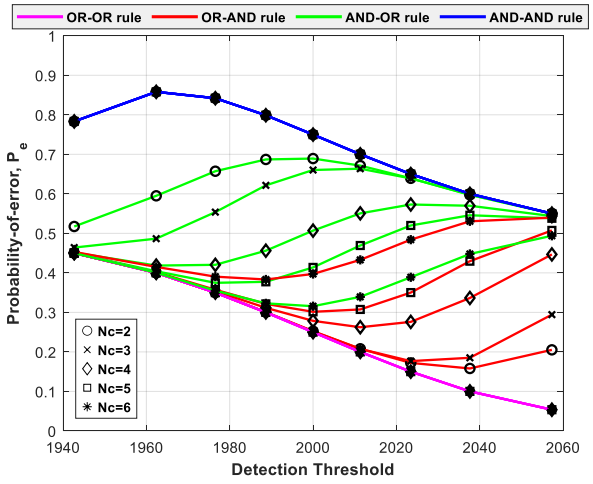
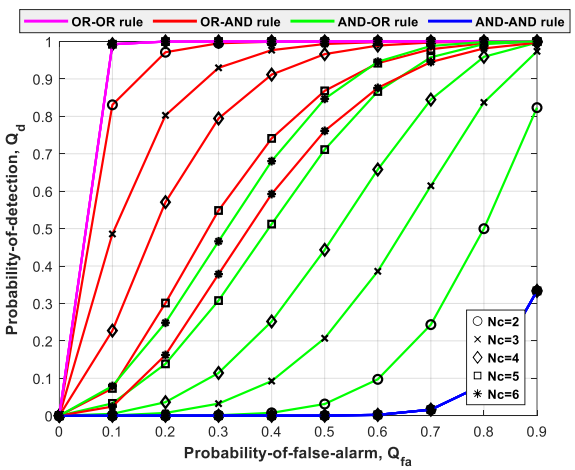


Figure 8: Variation of P_e with threshold and number of clusters for CB-CSS with different fusion rules

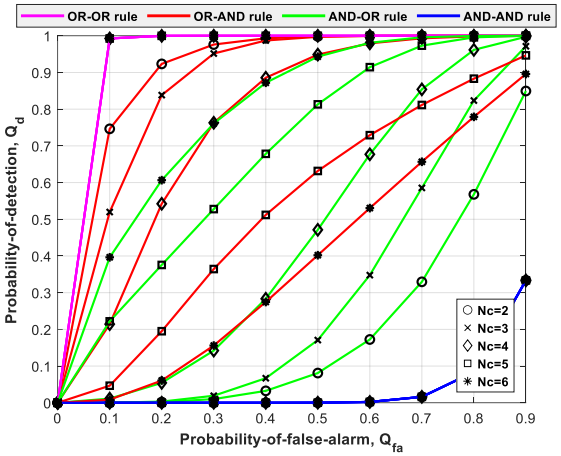
Further, we wanted to look into the effect of number of clusters on detection performance with different fusion rules (AND-AND, OR-OR, OR-AND and AND-OR fusion rule). The detection performance for two different spatial configurations of 20 SUs is shown in figures 9(a) and (b). The spatial configurations are given in Appendix section (figure 11 and 12). The parameters that were used in simulation are same as shown in Table 4. From figures it is quite evident that the detection performance for OR-OR and AND-AND rule-base fusion remain unaffected by the change in number of clusters. The OR-OR rule-base fusion has superior performance in terms of detection than other fusion rules and the AND-AND rule-based fusion show the worst detection performance. Further, the P_d for OR-AND/AND-OR rule-based fusion decreases/increases with the increase in number of clusters, respectively. This is explicable with the previous studies [29] which shows that in OR combining, the detection capability improves with the rise in number of SUs and in AND rule, detection performance decreases with the rise in amount of SUs.

Moreover, as the clusters increases for fixed number of SUs, the CH count also increases and the amount of SUs in each cluster decreases. Since, in OR-AND/AND-OR rule-based fusion, the intra-cluster combining is done with OR/AND rule and inter-cluster combining uses AND/OR rule, therefore, with increase in clusters, the amount of SUs in each clusters decreases and detection performance also decreases/increases as it uses OR/AND rule-based fusion, respectively. Further, with increase in clusters, the number of CHs collaborating for spectrum sensing increases and detection performance decreases/increases as it uses AND/OR rule-based fusion, respectively. Thus, in conclusion the detection performance for OR-AND/AND-OR rule-based fusion decreases/increases with the rise in clusters, respectively.

Further, it is observed that the detection performance of OR-AND rule-based fusion is better than AND-OR rule-based fusion till a certain value of clusters. This can be explained as both the fusion (OR-AND and AND-OR rule-based fusion) shows opposite behaviour to each other with increase in clusters, therefore there will be a threshold after which AND-OR fusion rule will have superior detection performance than OR-AND fusion rule. However, this threshold is not same even if the number of SUs (in our case 20 SUs) are same because of different spatial configurations that vary the amount of SUs in clusters. This is validated from figures 9(a) and (b). For first spatial configuration (as shown in figure 11 of Appendix section), the OR-AND rule shows enhanced detection capability in comparison to AND-OR rule till cluster count reaches five and for six or more clusters, the performance behaviour is reciprocated as shown in figure 9 (a). However, for second spatial configuration (as shown in figure 12 of Appendix section) with same number of SUs (M=20) grouped into same number of clusters (two, three, four, five and six number of clusters, respectively), the cluster count at which AND-OR rule-based fusion surpasses the performance of OR-AND rule-based fusion is different than that seen in figure 9 (a). The OR-AND rule shows improved performance in detection as compared to AND-OR rule till number of clusters reaches four and for five or more clusters, AND-OR rule has better detection ability as shown in figure 9 (b). Therefore, we can say that optimum amount of cluster for a given fusion rule (OR-AND, AND-OR) in order to obtain better detection performance, vary on spatial configuration and is not fixed for a given amount of SUs. Thus, it can be concluded that both the factors, i.e., cluster count and the count of SUs in each cluster affect the detection capability of OR-AND and AND-OR fusion rules.



(a) ROC curve for first spatial configuration



(b) ROC curve for second spatial configuration

Figure 9: ROC curve for CB-CSS for different number of clusters with various fusion rules.

The above simulations were carried out with an assumption that all SUs have same sensing SNR thus having same detection probability. Further, we studied the effect of varying sensing SNR at different SUs. Figure 10 shows ROC curve for four cases: (i) all SUs have same SNR values (i.e. -20 dB), (ii) SUs have varying SNR values of -8 dB, -10dB, -15 dB, -17 dB and -20 dB (i.e. SNR $\geqslant$-20 dB), (iii) SUs have varying SNR values of -20 dB, -21 dB, -23 dB, -24 dB and -25 dB (i.e. SNR $\leqslant$-20 dB) and (iv) SUs have varying SNR values of 0 dB, -8 dB, -20 dB, -23 dB and -25 dB (i.e. SNR $\geqslant$-20 dB and $\leqslant$-20 dB).It is evident from the figure that performance of SNR $\geqslant$ -20 dB scenario is best as compared to other cases followed by the SUs having SNR $\geqslant$ and <-20 dB because detection performance is enhanced with increase in SNR. Further, it is evident that if SUs has varying SNR which is less than -20 dB, then detection performance of SUs with varying SNR will be low in comparison to SUs having same SNR (-20 dB) because detection performance degrades with decrease in SNR. This variation of detection probability with SNR was validated in our previous paper as well [29].

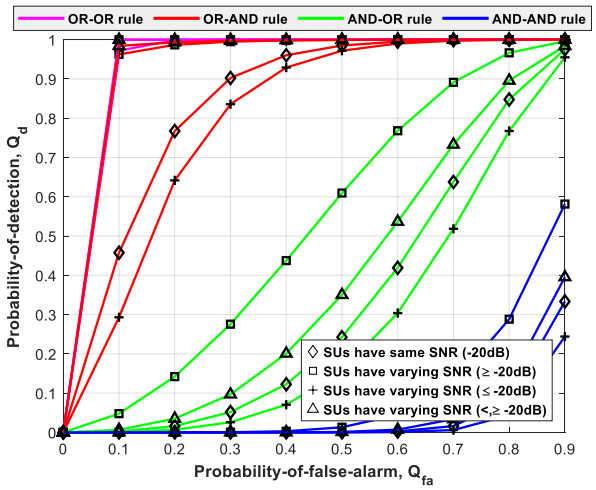


Figure 10: ROC curve for SUs having same SNR values and varying SNR values.

6 Conclusion

In this manuscript we have employed the unsupervised machine learning technique (K-means clustering) for clustering of SUs in the network. The CB-CSS has provided significant data overhead reduction (70 to 85%) at FC in comparison to CSS.

The comparison of K-means clustering with other techniques (DBSCAN and LEACH) is done in terms of data overhead, latency and detection probability. It is concluded that the suggested method is optimal option for low SNR conditions as K-means clustering provides better detection probability as compared to DBSCAN and LEACH in low SNR conditions using OR-AND fusion rule and reduced data overhead at FC. However, latency is large in K-means clustering as compared to DBSCAN and LEACH. Thus, a trade-off is required between detection performance and latency. Further, K-means clustering provides structured grouping and is deterministic, i.e., in K-means clustering, we can choose the number of clusters based on SNR conditions, data overhead handling capacity of FC and SU distribution in the network.

Further, this study examines the detection or sensing performance of CB-CSS for various decision fusion rules for Rayleigh fading channel over low SNR. The OR-OR rule-based fusion has outperformed other fusion rule, however, AND-AND rule-based fusion has shown the worst performance. In addition, OR-AND rule-based fusion outperforms AND-OR rule-based fusion till a certain value of clusters. This value of clusters may vary with the shift in network configuration as the number or distribution of SUs in each cluster may vary. Since, the current communication system is dynamic in nature and therefore it is not necessary to have same amount of SUs in each cluster. The amount of SUs in each cluster may vary with the change in spatial configuration. Thus, we can select the type of fusion rule to be employed depending upon the spatial configuration as mapped to the simulation results.

In conclusion, based on our study, it can be decided that whether to use AND-OR combining or OR-AND combining for a specific cluster count and with a specified number or distribution of SUs in each cluster. Also, number of clusters to be formed can be decided on the basis of data handling and processing capability of FC.

The P_e based analysis in comparison to threshold is done to evaluate the optimum threshold for least P_e. Further, it indicates that P_e increases/decreases with increase in clusters for OR-AND/AND-OR rule-based fusion, respectively and P_e for OR-OR and AND-AND rule-based fusion is not affected by number of clusters.

Further, influence of varying sensing SNR values at SUs in comparison to same SNR at each SU assumption is studied and it is concluded that detection performance increases in case SNR values are greater than same SNR assumption and vice versa.

This study provides the detection performance analysis for K-means clustering based grouping of SUs in CB-CSS. The fuzzy-based adaptive clustering can be seen as a potential future extension as fuzzy-logic–based adaptive clustering can dynamically tune clustering parameters (e.g., number of clusters) in response to fluctuating SNR conditions or SU mobility. Further, computational complexity of clustering algorithm was not addressed in the current study and can be taken into account in future studies.

References

[1] Nepal, N., Shakya, S., & Koirala, N. (2014). Energy detection-based techniques for Spectrum sensing in Cognitive Radio over different fading Channels. *Cyber Journals: Multidisciplinary Journals in Science and Technology. Journal of Selected Areas in Telecommunications (JSAT), 4* (2), 15-22.

[2] Atapattu, S., Tellambura, C., & Jiang, H. (2011). Spectrum Sensing via Energy Detector in Low SNR. *IEEE International Conference on Communications (ICC), Kyoto, Japan*, 1-5. doi: 10.1109/icc.2011.5963316

[3] Atapattu, S., Tellambura, C., & Jiang, H. (2011). Energy Detection Based Cooperative Spectrum Sensing in Cognitive Radio Networks. *IEEE Trans. Wireless Commun., 10* (4), 1232-1241. doi: 10.1109/TWC.2011.012411.100611

[4] Muzaffar, M. U., & Sharqi, R. (2024). A review of spectrum sensing in modern cognitive radio networks. *Telecommun Syst, 85*, 347–363. https://doi.org/10.1007/s11235-023-01079-1

[5] Ganesan, G., & Li, Y. (2007). Cooperative spectrum sensing in cognitive radio,part I: two user networks. *IEEE Trans. Wireless Commun., 6* (6), 2204–2213. doi: 10.1109/TWC.2007.05775

[6] Fan, R., & Jiang, H. (2010). Optimal multi-channel cooperative sensing in cognitive radio networks. *IEEE Trans. Wireless Commun., 9* (3), 1128–1138. doi: 10.1109/TWC.2010.03.090467

[7] Zhang, W., Mallik, R., & Letaief, K. (2009). Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks. *IEEE Trans. Wireless Commun., 8* (12), 5761–5766. doi: 10.1109/TWC.2009.12.081710

[8] Cabric, D., Mishra, S. M., & Brodersen, R. W. (2004). Implementation issues in spectrum sensing for cognitive radios. *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 1*,772-776. doi: 10.1109/ACSSC.2004.1399240

[9] Letaief, K. B., & Zhang, W. (2009). Cooperative Communications for Cognitive Radio Networks. *Proceedings of the IEEE. 97*, (5), 878–893. doi: 10.1109/JPROC.2009.2015716

[10] Dikmese, S., Lamichhane, K., & Renfors, M. (2021). Novel filter bank-based cooperative spectrum sensing under practical challenges for beyond 5G cognitive radios.*J Wireless Com Network, 30*, https://doi.org/10.1186/s13638-020-01889-w

[11] Dharmapuri, C. M., & Reddy, B. V. R. (2023). Performance analysis of different combining rules for cooperative spectrum sensing in cognitive radio communication. *10th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India*,1061-1064.

[12] Joshi, G.P., & Kim, S. W. (2016). A survey on node clustering in cognitive radio wireless sensor networks, *Sensors, 16* (9), 1465, https://doi.org/10.3390/s16091465

[13] Nardis, L. D., Domenicali, D. & Benedetto, M. G. (2009). Clustered hybrid energy-aware cooperative spectrum sensing (CHESS). *4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM'09), Hanover, Germany*, 1–6. doi: 10.1109/CROWNCOM.2009.5189147

[14] Hussain, S., & Fernando, X. (2012). Approach for cluster-based spectrum sensing over band-limited reporting channels, *IET Commun., 6* (11), 1466–1474. https://doi.org/10.1049/iet-com.2010.0510

[15] Guo, C., Peng, T., Xu, S., Wang, H., & Wang, W. (2009). Cooperative spectrum sensing with cluster-based architecture in cognitive radio networks. *IEEE 69th Vehicular Technology Conference (VTC Spring 2009), Barcelona, Spain*, 1–5. doi: 10.1109/VETECS.2009.5073471

[16] Thilina, K. M., Choi, K. W., Saquib, N. & Hossain, E. (2013). Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Networks. *IEEE Journal on selected areas in communications, 31*(11), 2209-2221. doi: 10.1109/JSAC.2013.131120

[17] Khamayseh, S., & Halawani, A. (2020). Cooperative Spectrum Sensing in Cognitive Radio Networks: A Survey on Machine Learning-based Methods. *Journal of Telecommunications and Information Technology, 81* (3), 36-46. https://doi.org/10.26636/jtit.2020.137219

[18] Janu, D., Singh, K., & Kumar, S. (2022). Machine learning for cooperative spectrum sensing and sharing: A survey. *Transactions on Emerging Telecommunications Technologies, 33* (1). https://doi.org/10.1002/ett.4352

[19] Kumar, V., Kandpal, D. C., Jain, M., & Debnath, S. (2016). K -mean Clustering based Cooperative Spectrum Sensing in Generalized κ - μ Fading Channels. *22nd National Conference on Communications (NCC), Guwahati, India*, 1-5. doi: 10.1109/NCC.2016.7561130

[20] Sharma, Y., Sharma, R., & Sharma, K. K. (2023). Enhancing Throughput Over Varied Fading Channels for Cluster-Based Cooperative Spectrum Sensing in Cognitive Wireless Networks.*Int. J. Wireless Inf. Networks, 30*, 227–240. https://doi.org/10.1007/s10776-023-00601-1

[21] Bhatti, D. S., Ahmed, S., Chan, A. S., & Saleem, K. (2020). Clustering formation in cognitive radio networks using machine learning. *AEU - International Journal of Electronics and*

Communications, 114, https://doi.org/10.1016/j.aeue.2019.152994

[22] Sharma, G., & Sharma, R. (2017). Performance evaluation of distributed CSS with clustering of secondary users over fading channels. *International Journal of Electronics Letters, 6* (3), 288-301. https://doi.org/10.1080/21681724.2017.1357762

[23] Sharma, G. & Sharma, R. (2019). Cluster-based distributed cooperative spectrum sensing over Nakagami fading using diversity reception. *IET Networks, 8* (3), 211-217. https://doi.org/10.1049/iet-net.2018.5002

[24] Liu, X., Zhang, X., Ding, H., & Peng, B. (2019). Intelligent clustering cooperative spectrum sensing based on Bayesian learning for cognitive radio network, *Ad Hoc Networks, 94.* https://doi.org/10.1016/j.adhoc.2019.101968.

[25] Kozal, A., Merabti, M., Bouhafs, F. (2014). Multi-hop Cluster Based Cooperative Spectrum Sensing Approach for Cognitive Radio Networks. *International Journal of Scientific & Engineering Research,* 5 (5), 760-769.

[26] Mortada, M. R., Nasser, A., Mansour, A., Yao K. C. (2021). In-Network Data Aggregation for Ad Hoc Clustered Cognitive Radio Wireless Sensor Network. *Sensors (Basel), 21*(20), 6741. doi: 10.3390/s21206741.

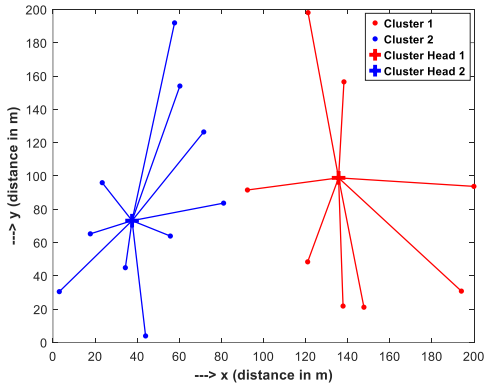
[27] Tao, B., Wu, J., Dou, X., Wang, J., Xu, Y. (2025). Memorial K-means clustering for cooperative spectrum sensing in cognitive wireless sensor networks at low SNR regimes. *Sensor Review*, *45*(3), 443–452, doi: https://doi.org/10.1108/SR-10-2024-0883

[28] Tomás, A., Brito, J.(2025). Algorithm for Sensor Exclusion and Dynamic Cluster Head Selection in Cognitive Radio Networks. *The Twentieth International Conference on Digital Telecommunications*, Nice, France, 5-10.

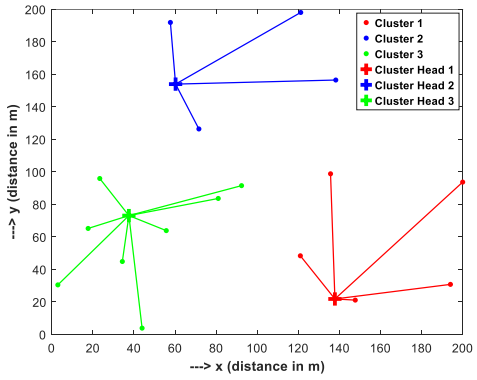
[29] Sharma, A., Pandit, S., Kumar, R. (2024). Cooperative Spectrum Sensing Using Energy-Based Detection for Low SNR Regime over Rayleigh Fading Channel. *2024 International Conference on Integrated Circuits, Communication, and Computing Systems (ICIC3S), Una, India,* 1-6. doi: 10.1109/ICIC3S61846.2024.10602980

[30] Deering, S. E.,& Hinden, B. (2017). Internet Protocol, Version 6 (IPv6) Specification, RFC 8200, *Internet Engineering Task Force (IETF),* https://datatracker.ietf.org/doc/html/rfc8200

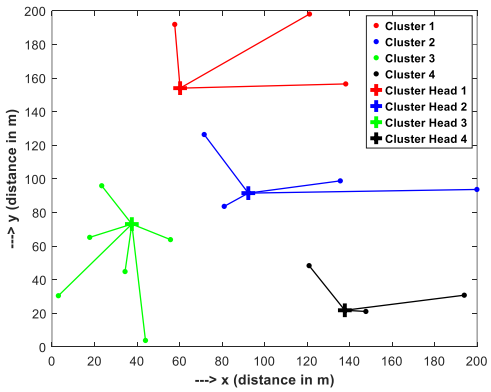
Appendix



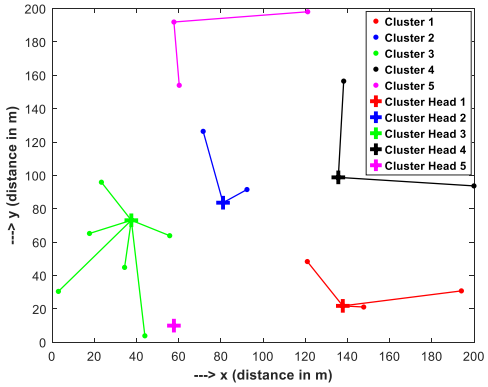
(a)



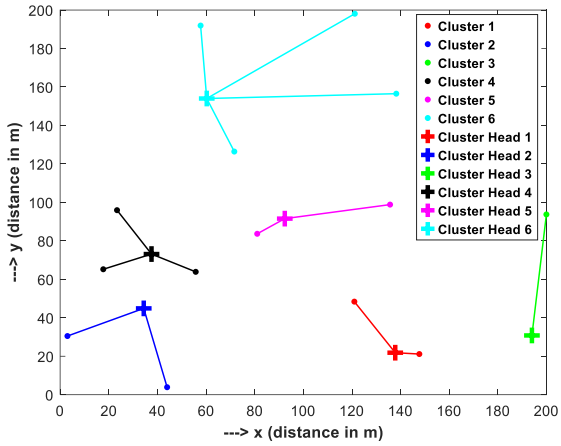
(b)



(c)

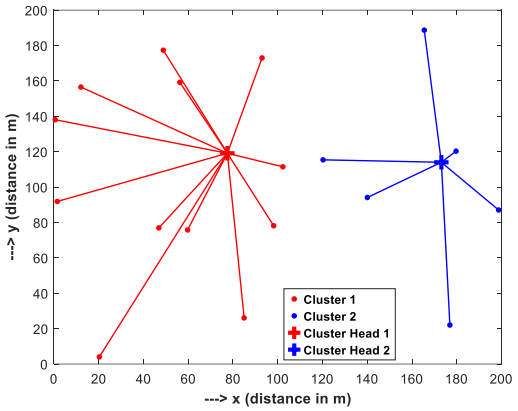


(d)

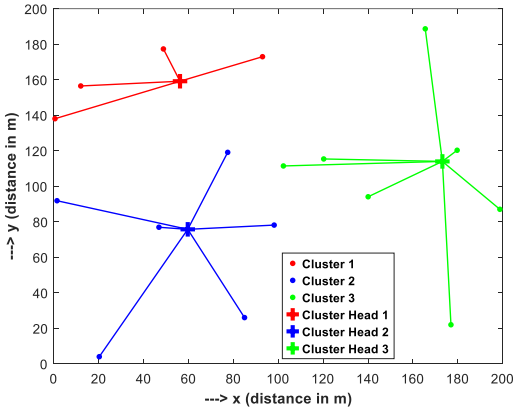


(e)

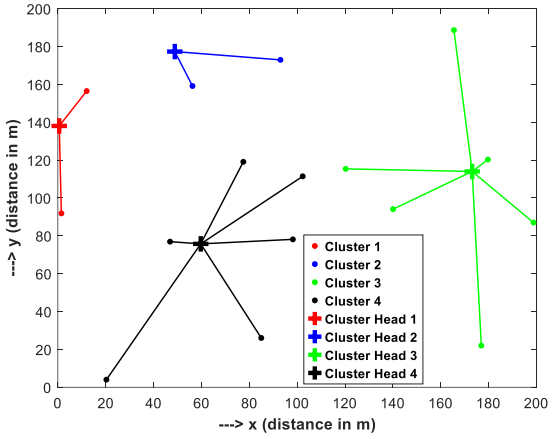
Figure 11: First spatial configuration: Cluster formation of 20 SUs using K-means clustering technique into a) N_c =2, b) N_c =3, c) N_c =4, d) N_c =5 and e) N_c =6 clusters.



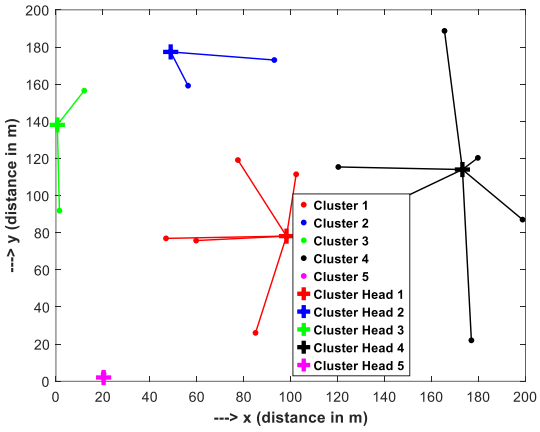
(a)



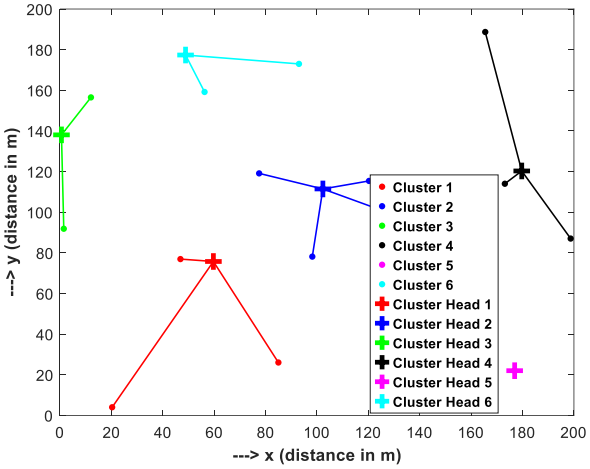
(b)



(c)



(d)



(e)

Figure 12: Second Spatial configuration: Cluster formation of 20 SUs using K-means clustering technique into a) N_c =2, b) N_c =3, c) N_c =4, d) N_c =5 and e) N_c =6 clusters.

Bias Mitigation in Big Data Systems: A Systematic Review, Comparative Synthesis, and Decision Framework for Fair Algorithm Design

Daniel Kweku Assumang[1*], Ifeoma Eleweke[2], Chiamaka Ezenwaka[3], Olabode Soetan[4], Martin Msughter Vincent[5], Elizabeth Ayodeji Adeyefa[6]
[1]College of Professional Studies, Roux Institute, Northeastern University, Maine USA
[2]College of Technology and Engineering, Westcliff University, California USA
[3]Department of Operation, CBRE Group Inc, Texas USA
[4]Department of Tax Technology Consulting practice, Deloitte LLP, Chicago USA
[5]National Graduate Institute for Policy Studies
[6]School of Business, Economics and Technology, Campbellsville University, USA

E-mail: dkassumang96@gmail.com

Keywords: Bias mitigation, big data, algorithmic fairness, fair algorithm design, ethical AI, automated decision-making

Received: July 28, 2025

As Big Data systems increasingly shape decision-making across healthcare, finance, hiring, criminal justice, and public governance, concerns regarding algorithmic bias and unfair outcomes have become more pronounced. While data-driven systems can improve efficiency, prediction, and service delivery, they may also reproduce historical inequalities through biased datasets, proxy variables, flawed labels, or opaque optimization processes.

This study presents a systematic review and comparative synthesis of bias mitigation strategies in Big Data systems, with the aim of identifying effective approaches for fair algorithm design. Guided by a PRISMA-informed review methodology, literature published between 2018 and 2026 was screened across major academic databases, resulting in 27 studies included for final analysis.

The findings classify mitigation approaches into pre-processing, in-processing, post-processing, and hybrid methods. Pre-processing techniques were found effective for addressing representation imbalance and data quality issues, while in-processing methods provided stronger fairness control where model retraining was feasible. Post-processing approaches were most practical for legacy or proprietary systems, whereas hybrid strategies were strongest in high-risk contexts requiring layered safeguards. The review further shows that no single fairness metric or mitigation technique is universally optimal; effectiveness depends on domain risk, bias source, regulatory obligations, and operational constraints.

Based on these findings, the paper proposes the Fair Algorithm Design Decision Framework to guide organizations in selecting context-appropriate fairness interventions. The study concludes that bias mitigation should be treated as a continuous lifecycle responsibility integrating data governance, model design, human oversight, and ongoing monitoring.

Povzetek: Študija primerja pristope za zmanjševanje algoritmične pristranskosti ter poudarja, da je treba pravičnost zagotavljati celostno, neprekinjeno in glede na kontekst uporabe.

1 Introduction

Background to the study

Big Data systems increasingly shape decision-making across healthcare, finance, criminal justice, hiring, marketing, and public governance. The growing volume, velocity, and variety of data allow organizations to improve forecasting, optimize operations, personalize services, and support real-time decision processes (Hossin et al., 2023; Aljehani et al., 2024).

Despite these benefits, Big Data systems can reproduce and amplify inequality when bias enters the data pipeline or model design. Bias may arise from unrepresentative sampling, historical discrimination embedded in legacy records, subjective labeling, proxy variables, or feedback loops after deployment. Prior studies have documented

biased outcomes in recruitment systems, criminal risk scoring, healthcare prediction, and credit assessment (Mittelstadt et al., 2016; Chen, 2023; Shahul Hameed et al., 2024).

Accordingly, the challenge is not only to build accurate predictive systems, but also to design systems that are fair, transparent, accountable, and socially responsible. As the use of automated decision-making expands, reducing bias has become a central requirement for trustworthy Big Data systems (Shahul Hameed et al., 2024).

Problem statement

A central challenge in Big Data systems is that bias may arise at multiple stages, including data collection, labeling, feature selection, model training, and deployment. Common forms include sampling bias, where certain populations are underrepresented, and historical bias, where past inequalities are encoded into training data (Mittelstadt et al., 2016). Once embedded in machine learning systems, these biases can scale rapidly and disproportionately disadvantage women, minority groups, and economically vulnerable populations (Chohlas-Wood et al., 2023).

In recruitment, algorithms trained on historical hiring decisions may replicate gender bias, particularly in male-dominated sectors (Chen, 2023; Ho et al., 2025; Köchling and Wehner, 2020). In healthcare, poorly designed models may underperform for minority populations if demographic variation is not adequately represented in training data (Shahul Hameed et al., 2024). In criminal justice, biased risk assessment tools may reinforce racial disparities (Joseph, 2024).

These examples demonstrate that algorithmic bias is not merely a technical error; it is a governance and social justice issue requiring robust mitigation strategies and careful system design.

Research gap

Although algorithmic fairness has received growing scholarly attention, much of the existing literature either discusses ethical principles at a conceptual level or evaluates individual mitigation techniques in isolation. Comparative evidence across pre-processing, in-processing, post-processing, and hybrid strategies remains fragmented. Furthermore, limited guidance exists on how organizations should choose appropriate mitigation methods under varying data conditions, fairness objectives, and operational constraints.

Research questions

This study addresses the following research questions:

RQ1: What are the principal sources of bias in Big Data systems?

RQ2: How do pre-processing, in-processing, post-processing, and hybrid mitigation methods compare in effectiveness, scalability, and fairness–accuracy trade-offs?

RQ3: How do mitigation approaches vary across domains such as healthcare, finance, hiring, and criminal justice?

RQ4: What decision framework can guide fair algorithm design in practice?

Contributions

This paper makes three contributions. First, it presents a systematic review of recent literature on bias mitigation in Big Data systems. Second, it provides a comparative synthesis of major mitigation approaches across technical and application dimensions. Third, it proposes a decision framework to support the practical selection of fairness strategies based on domain risk, data quality, and deployment constraints.

2 Methodology

2.1 Review design and search strategy

This study employed a PRISMA-guided systematic literature review to identify and synthesize evidence on bias mitigation in Big Data systems. Searches were conducted across Google Scholar, IEEE Xplore, ScienceDirect, Scopus, Web of Science, and PubMed using combinations of terms related to algorithmic fairness, bias mitigation, ethical AI, healthcare bias, hiring bias, and automated decision-making. The review focused on studies published between 2018 and 2026.

2.2 Study selection and eligibility

The initial search identified 200 records, including 14 studies obtained through backward citation screening. After removal of 38 duplicates, 162 studies underwent title and abstract screening, resulting in 61 full-text articles assessed for eligibility. Following exclusion of studies lacking methodological rigor, fairness relevance, or sufficient analytical value, 27 studies were retained for comparative synthesis.

Studies were included if they examined bias in Big Data or automated decision systems, evaluated mitigation approaches or fairness metrics, and were published in peer-reviewed English-language sources. Editorials, inaccessible full texts, and studies unrelated to fairness or algorithmic bias were excluded. The PRISMA selection process is summarized in Table 1 and Figure 1.

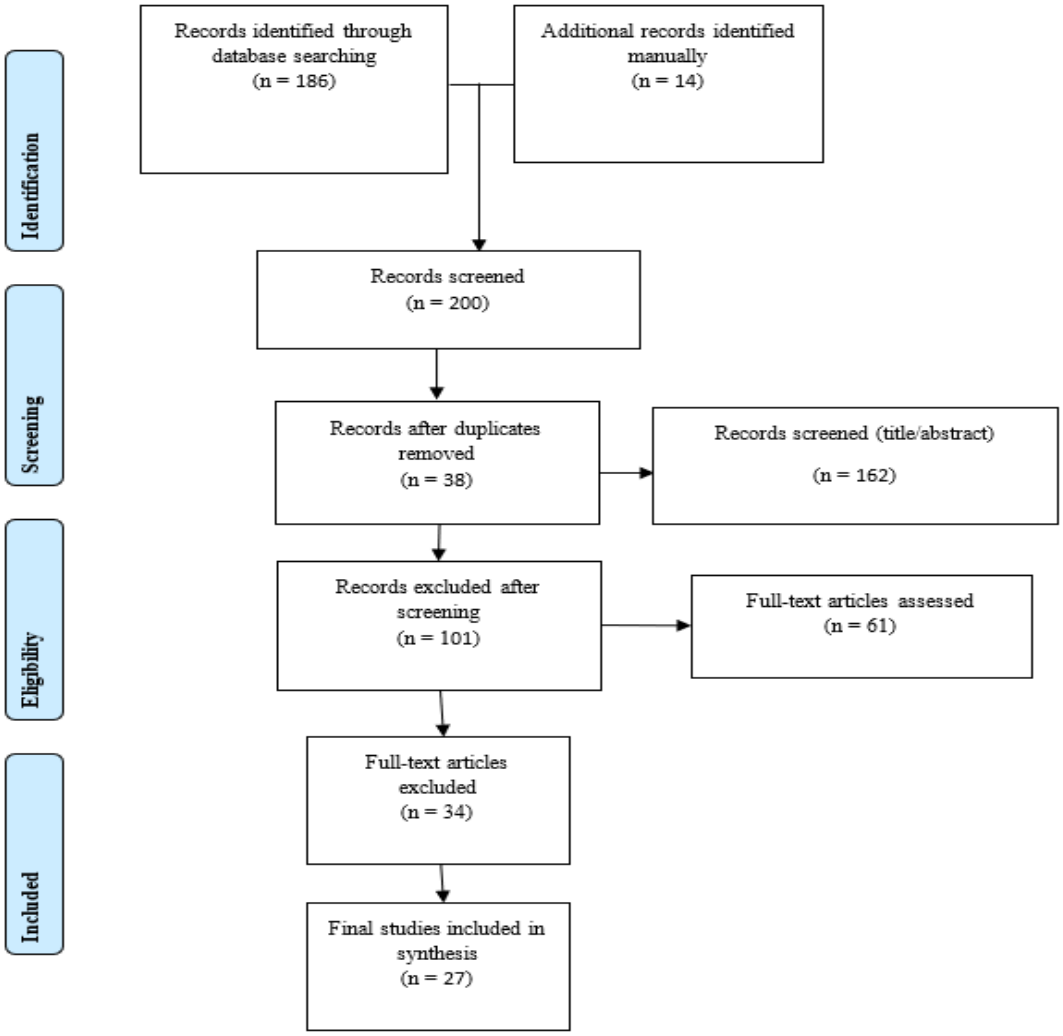
2.3 Data extraction and comparative synthesis

A structured extraction framework was used to record study characteristics, including application domain, bias category, mitigation stage, fairness objective, methodological design, and reported outcomes. Included studies were coded into four mitigation categories: pre-processing, in-processing, post-processing, and hybrid/governance approaches.

Because reviewed studies differed substantially in datasets, metrics, and evaluation methods, a comparative narrative synthesis was adopted rather than statistical meta-analysis. Comparative analysis focused on fairness improvement, predictive accuracy, interpretability, scalability, and suitability for high-risk deployment contexts. Greater analytical weight was assigned to empirical and systematic review studies with transparent methodology and measurable fairness outcomes.

Table 1: PRISMA Flow diagram

PRISMA Stage	Count
Records identified through database search	186
Additional records identified manually	14
Total records identified	200
Duplicates removed	38
Records screened (title/abstract)	162
Records excluded after screening	101
Full-text articles assessed	61
Full-text articles excluded	34
Final studies included in synthesis	27



3 Foundations of bias in big data systems

Bias in Big Data systems refers to systematic distortions in data collection, representation, model learning, or decision outputs that produce unfair outcomes across individuals or groups. In algorithmic environments, bias may emerge from historical inequalities, incomplete data capture, proxy variables, measurement error, optimization objectives, or deployment feedback effects rather than explicit malicious intent (Mittelstadt et al., 2016; Ferrara, 2023). Because automated systems operate at scale, such distortions can influence large populations and significantly affect access to employment, healthcare, credit, and public services.

Table 2: Major sources of bias in big data systems

Bias Type	Stage	Example	Typical Risk	Author
Sampling Bias	Data collection	Underrepresented populations	Poor subgroup accuracy	Chen et al., 2021; Shahul Hameed et al., 2024;
Historical Bias	Legacy datasets	Past hiring inequality	Reproduced discrimination	Hanna et al., 2024; Varsha, 2023
Label Bias	Annotation	Subjective recruiter evaluations	Skewed learning	Favier et al., 2023; Ferrara,

				2023; Hanna et al., 2024
Measurement Bias	Feature capture	Unequal diagnostics	Hidden unfairness	Chohlas-Wood et al., 2023

Bias may therefore emerge at multiple stages of the machine learning lifecycle, including data collection, feature engineering, model optimization, and deployment. Even where protected attributes are removed, correlated variables may continue to reproduce discriminatory outcomes (Wang et al., 2024). Similarly, models trained on historical records may inherit social or institutional inequalities embedded within legacy datasets (Köchling and Wehner, 2020; Joseph, 2024). Deployment environments may further intensify unfairness through feedback loops, automation overreliance, and changing population behavior.

Big Data environments amplify these risks because scale, velocity, and interconnected data ecosystems increase both the reach and persistence of biased decisions. Automated outputs may propagate across downstream systems such as lending, insurance pricing, hiring, and public administration, while reduced transparency can make fairness failures difficult to detect or correct (Ukanwa, 2024).

These findings indicate that bias mitigation must address multiple stages of the data lifecycle rather than relying on a single technical correction. This provides the foundation for the comparative review of mitigation strategies presented in Section 5.

4 Fairness definitions and evaluation metrics

Fairness in algorithmic decision-making refers to the principle that automated systems should not systematically disadvantage individuals or groups based on protected or socially sensitive characteristics such as race, gender, ethnicity, disability, or socio-economic status (Mittelstadt et al., 2016; Ferrara, 2023). However, fairness is not a single measurable property. Different contexts may prioritize equality of outcomes, equal opportunity, calibration, transparency, or procedural fairness, and these objectives may conflict when group base rates differ. Consequently, fairness evaluation must be aligned with domain-specific risks and operational goals rather than assuming a universally optimal metric.

Table 3: Major fairness metrics in algorithmic systems

Metric	Definition	Common Use	Key Limitation
Demographic Parity	Equal positive outcome rates across groups	Hiring, outreach systems	May ignore qualification differences
Equal Opportunity	Equal true positive rates	Healthcare, lending	Does not equalize false positives
Equalized Odds	Equal true and false positive rates	Criminal justice	Difficult to optimize jointly
Calibration	Comparable risk reliability across groups	Finance, insurance	May conflict with parity metrics
Individual Fairness	Similar individuals receive similar outcomes	Admissions, recruitment	Requires defining similarity
Counterfactual Fairness	Outcome unchanged under protected-attribute change	Causal fairness analysis	Complex causal assumptions

Fairness metrics also involve important trade-offs. Strict parity constraints may reduce aggregate predictive accuracy, while calibration objectives may conflict with equalized outcome measures when base rates differ. Existing metrics further simplify fairness into statistical relationships and may overlook structural inequality, intersectional harms, or procedural legitimacy (Hagendorff, 2020; Singh et al., 2024). Consequently, fairness metrics should be treated as decision-support tools rather than complete ethical solutions.

Because mitigation techniques optimize different fairness objectives, selecting an intervention without first defining the relevant fairness metric risks addressing the wrong problem. This provides the basis for the comparative review of mitigation strategies presented in Section 5.

5 Comparative review of bias mitigation techniques

Bias mitigation approaches are commonly categorized according to the stage of the machine learning lifecycle at which intervention occurs: pre-processing, in-processing, post-processing, and hybrid approaches. Each category addresses different sources of unfairness and involves trade-offs in scalability, interpretability, implementation complexity, and predictive performance (Ferrara, 2023; Varsha, 2023). No single mitigation strategy is universally optimal, as effectiveness depends on the dominant source of bias, fairness objective, deployment environment, and regulatory context.

Table 4: Comparative summary of bias mitigation techniques

Category	Typical Methods	Main Strengths	Main Limitations	Best Context	Author
Pre-processing	Reweighing, resampling, synthetic data generation	Scalable, model-agnostic, improves representation	May not address latent proxy bias	Imbalanced or structured datasets	Holstein et al., 2023; Hosseinzadeh et al., 2021; Qi et al., 2019; Almasoud and Idowu, 2025
In-processing	Fairness constraints, adversarial debiasing, fair representation learning	Strong fairness control during training	Technically complex, less interpretable	Custom ML pipelines	Shahul Hameed et al., 2024; Chohlas-Wood et al., 2023
Post-processing	Threshold adjustment, reject option classification, output calibration	Fast deployment, no retraining required	Often corrective rather than preventive	Legacy or vendor-controlled systems	Hanna et al., 2023; Mittelstadt et al. 2016
Hybrid	Multi-stage mitigation and governance integration	Broad protection across lifecycle stages	Higher cost and governance burden	High-risk domains	Holstein et al., 2023; Hosseinzadeh et al., 2021; Qi et al., 2019; Almasoud and Idowu, 2025; Belenguer, 2022

Pre-processing methods are particularly effective where bias arises from sampling imbalance, historical underrepresentation, or noisy labels (Holstein et al., 2023; Hosseinzadeh et al., 2021; Qi et al., 2019). In-processing approaches provide stronger fairness control by embedding constraints directly into model optimization, although they often require advanced technical expertise and reduced interpretability (Shahul Hameed et al., 2024; Chohlas-Wood et al., 2023). Post-processing methods are operationally attractive when models cannot easily be retrained, while hybrid approaches are increasingly favored in high-risk domains because they address multiple bias sources simultaneously (Singh et al., 2024; Farayola et al., 2025). Recent studies also highlight the growing importance of adversarial debiasing and fair representation learning in healthcare and other high-dimensional AI environments (Hasanzadeh et al., 2025).

Across studies, three recurring trade-offs emerge: fairness versus predictive accuracy, fairness versus interpretability, and fairness versus operational cost. Strict parity constraints may reduce aggregate performance where group base rates differ, while advanced debiasing models may become difficult to explain to regulators or affected users. Similarly, layered monitoring and governance mechanisms improve fairness resilience but increase implementation complexity and resource requirements. These findings suggest that fairness should be managed as a strategic lifecycle objective rather than treated as an isolated technical adjustment.

The comparative evidence further indicates that mitigation effectiveness depends less on identifying a universally

superior algorithm and more on matching interventions to specific bias sources and operational conditions. This provides the basis for the domain-specific findings presented in Section 6.

6 Domain-specific findings

Although fairness principles are theoretically generalizable, the expression of algorithmic bias varies substantially across domains due to differences in data quality, regulatory expectations, decision risk, historical inequality, and operational constraints. The comparative synthesis indicates that domain context strongly influences dominant bias sources, fairness metric selection, acceptable accuracy trade-offs, and governance requirements.

Table 5: Cross-domain comparative summary

Domain	Primary Harm Risk	Common Bias Sources	Relevant Metrics	Typical Mitigation Priorities
Healthcare	Missed care / unsafe outcomes	Underrepresentation, measurement bias	Equal opportunity, calibration	Subgroup validation, balanced data, clinician oversight
Finance	Exclusion / unlawful discrimination	Proxy bias, historical exclusion	Calibration, disparate impact	Explainability, audits, proxy testing
Hiring	Opportunity denial	Historical labels, proxy variables	Demographic parity, adverse impact	Ranking audits, human review
Criminal Justice	Liberty and trust harms	Historical enforcement, feedback loops	Equalized odds, calibration	Transparency, public oversight
Public Governance	Welfare exclusion / legitimacy	Incomplete records, access gaps	Inclusion metrics, procedural fairness	Appeals processes, governance controls

Healthcare systems frequently experience fairness challenges related to demographic underrepresentation and measurement bias, particularly in diagnostic and predictive models (Hanna et al., 2024; Hasanzadeh et al., 2025). Financial systems are more strongly affected by proxy discrimination and regulatory transparency concerns, while hiring systems often inherit historical recruitment inequalities through biased labels and ranking patterns (Chen, 2023; Köchling and Wehner, 2020). In criminal justice and public governance contexts, fairness concerns extend beyond statistical parity to include procedural legitimacy, public accountability, and the social consequences of automated decision-making (Joseph, 2024; Ukanwa, 2024).

Across domains, three findings consistently emerge. First, fairness metric selection is context-dependent rather than universal. Second, governance mechanisms such as transparency, appeals processes, and human oversight are often as important as technical mitigation methods. Third, data quality remains foundational, as poor representation and historical distortions frequently undermine fairness before model training begins.

The synthesis further suggests that fairness controls should scale proportionally with deployment risk. High-risk systems such as healthcare triage, criminal justice scoring, credit access, and welfare allocation require rigorous subgroup testing, documented fairness metrics, human oversight, and periodic auditing. Lower-risk systems may justify lighter but still meaningful monitoring approaches. This risk-tiered logic aligns with emerging governance frameworks such as the European Union AI Act (Kusche, 2024).

These findings directly inform the decision framework proposed in Section 7, which translates comparative evidence and domain risk into practical guidance for fair algorithm design.

7 Proposed decision framework for fair algorithm design

The comparative findings demonstrate that no single mitigation strategy is universally optimal. Fairness outcomes depend on bias source, data quality, model transparency, domain risk, and operational constraints. To address this challenge, this paper proposes the Fair

Algorithm Design Decision Framework (FADDF), a structured model for selecting, deploying, and monitoring fairness interventions in Big Data systems.

7.1 Framework logic

The framework supports five core decisions:

1. Define decision context and deployment risk
2. Diagnose dominant bias sources
3. Select appropriate fairness metrics
4. Match mitigation strategies to technical constraints
5. Establish continuous monitoring and governance controls

Table 6: Core Inputs to the Framework

Input Dimension	Key Question
Decision Risk	Can errors affect health, liberty, employment, finance, or essential services?
Bias Source	Is bias driven by imbalance, proxy variables, historical discrimination, or deployment drift?
Data Quality	Are groups sufficiently represented and labels reliable?
Model Access	Can the model be retrained or audited internally?
Explainability Need	Must outputs be interpretable to regulators or users?
Regulatory Exposure	Are there legal obligations regarding transparency or discrimination?

Table 7: Mitigation selection matrix

Condition	Preferred Strategy	Rationale
Imbalanced data	Pre-processing	Corrects representation before training
Proxy discrimination	In-processing	Better control over learned relationships
Vendor-controlled models	Post-processing	No retraining required
High-risk deployment	Hybrid	Multi-layer protection justified
Dynamic environments	Monitoring + retraining	Fairness may drift over time

7.2 Risk-tier governance model

The framework distinguishes between high-risk, moderate-risk, and lower-risk systems.

- High-risk systems (healthcare triage, criminal justice scoring, credit access, welfare eligibility) require documented fairness metrics, human oversight, external audits, appeals processes, and incident response procedures.
- Moderate-risk systems (fraud detection, workforce analytics, pricing optimization) require internal audits, threshold review, and periodic retraining assessment.
- Lower-risk systems (content recommendation and non-essential personalization) may justify lighter but still meaningful bias monitoring.

This proportional governance logic aligns with emerging regulatory approaches such as the European Union AI Act (Kusche, 2024).

Table 8: Example framework applications

Scenario	Metric	Mitigation	Governance
Hiring platform	Disparate impact	Reweighing + ranking audit	Appeals + adverse impact testing
Clinical risk model	Equal opportunity	Resampling + subgroup validation	Drift monitoring
Loan approval model	Calibration	Feature audit + constrained optimization	Regulatory audit trail

7.3 Implementation principles

Successful deployment of fairness controls requires organizations to:

1. diagnose bias before selecting interventions
2. align metrics with domain-specific harms
3. match controls to deployment risk
4. maintain human oversight in high-impact systems
5. monitor fairness continuously after deployment
6. document decisions for accountability and auditability

The proposed framework contributes to the literature by integrating bias diagnosis, metric selection, mitigation matching, lifecycle monitoring, and governance controls into a single operational model bridging fairness theory and organizational implementation.

8 Discussion

This review demonstrates that bias mitigation in Big Data systems cannot be reduced to a single technical solution or universal fairness metric. The effectiveness of fairness interventions depends on the interaction between data quality, model design, deployment context, governance structure, and domain-specific risk. Consequently, organizations should move beyond searching for a universally "fair" algorithm and instead adopt context-sensitive mitigation strategies aligned with operational realities and potential harms.

One of the clearest findings across reviewed studies is that fairness is highly domain-dependent. In healthcare, minimizing missed diagnoses and ensuring subgroup safety may take priority over equal outcome rates, whereas hiring systems emphasize adverse impact reduction and procedural consistency. Financial systems prioritize calibration, explainability, and auditability, while criminal justice systems require balanced error distribution and public legitimacy. These differences indicate that fairness metric selection is not purely technical, but also institutional and normative in nature.

The review further suggests that many fairness failures originate before model training begins. Underrepresentation, historical distortions, weak labels, and proxy variables frequently undermine fairness at the data level, regardless of model sophistication. This finding implies that organizations may overemphasize advanced debiasing algorithms while underinvesting in data governance, subgroup representation, documentation, and continuous dataset evaluation. Fairness also cannot be sustained through one-time interventions alone, as real-world systems are subject to deployment drift, changing populations, feedback loops, and evolving regulatory expectations. Effective governance therefore requires ongoing monitoring, human oversight, transparency mechanisms, and periodic reassessment of subgroup outcomes.

Recent research in adaptive and intelligent control systems further highlights the limitations of static bias mitigation approaches in dynamic and safety-critical environments (Elrifaee, & Zayed, 2025). In autonomous systems, robotics, and nonlinear control architectures, operational conditions may evolve continuously due to uncertainty, feedback interactions, sensor noise, or environmental drift (Haq, Felicetti, Carnı, & Lamonaca, 2026). Under such conditions, fairness interventions designed using static datasets or fixed optimization assumptions may degrade over time or fail to respond adequately to changing subgroup behavior. Adaptive and feedback-driven control paradigms, including fuzzy control and dynamic optimization frameworks, therefore suggest important directions for future fairness research by enabling continuous adjustment, real-time monitoring, and resilience under uncertain deployment conditions. These findings reinforce the importance of lifecycle-oriented fairness governance in high-risk AI systems operating under real-time or safety-critical constraints.

Several unresolved challenges remain. Current fairness metrics often inadequately address intersectional harms, fairness under limited data conditions, and the governance of large-scale generative AI systems. In addition, tensions persist between fairness, privacy, interpretability, and predictive accuracy, particularly in high-risk domains. These challenges reinforce the need for interdisciplinary research integrating machine learning, governance, law, ethics, and public policy in the development of trustworthy AI systems.

9 Limitations and future research

This review has several limitations. The included studies span multiple disciplines, datasets, model architectures, fairness metrics, and evaluation approaches, limiting direct comparability and making statistical meta-analysis inappropriate. In addition, fairness itself remains conceptually contested, with studies prioritizing demographic parity, equal opportunity, calibration, explainability, or broader equity objectives differently. Publication bias may also affect the evidence base, as studies reporting successful mitigation outcomes are more

likely to appear in academic literature than null or unsuccessful implementations. Furthermore, because the review focused primarily on English-language and peer-reviewed sources, relevant industry practices and non-English contributions may be underrepresented.

The broader fairness literature also faces several structural limitations. Many mitigation techniques are evaluated using static benchmark datasets rather than dynamic real-world deployment environments, limiting understanding of fairness drift, feedback effects, and long-term governance challenges. Existing research further provides limited evidence regarding implementation cost, organizational accountability structures, and the operational sustainability of fairness interventions across sectors.

Several priorities therefore emerge for future research. Increasing attention is needed for fairness monitoring in dynamic environments, particularly in relation to deployment drift, subgroup degradation, and continuous auditing systems. Emerging generative AI and foundation models also introduce new fairness risks involving representation harms, synthetic content generation, and downstream automation effects. Additional research is needed on intersectional fairness, fairness–privacy trade-offs, human-AI interaction, and cross-jurisdiction regulatory alignment in globally deployed systems.

The proposed Fair Algorithm Design Decision Framework should therefore be viewed as an adaptable operational model rather than a universal solution. Future empirical work should validate the framework through case studies, deployment experiments, and cross-sector implementation analysis in areas such as hiring, healthcare, finance, and public governance.

10 Conclusion

This study examined bias mitigation in Big Data systems through a systematic review, comparative synthesis, and framework-development perspective. The findings demonstrate that algorithmic fairness cannot be achieved through isolated technical corrections alone, as unfair outcomes often emerge from interactions between data quality, historical inequality, model design, deployment environments, and institutional governance. The review further showed that no single fairness metric or mitigation strategy is universally optimal. Instead, effective bias mitigation depends on aligning fairness objectives, technical methods, and governance controls with domain-specific risks and operational conditions.

To address this challenge, the study proposed the Fair Algorithm Design Decision Framework (FADDF), which integrates bias diagnosis, fairness metric selection, mitigation matching, risk-tier governance, and post-deployment monitoring into a unified operational model. The framework contributes practical guidance for organizations seeking to implement fairness controls across healthcare, finance, hiring, criminal justice, and public governance systems.

Ultimately, fair algorithm design should be understood as a continuous lifecycle responsibility rather than a one-time compliance exercise. As automated decision systems continue to expand across high-impact sectors, future progress will depend on stronger data governance, continuous auditing, interdisciplinary collaboration, and accountable AI oversight capable of balancing predictive performance with social legitimacy and public trust.

References

[1] Aljehani, S. B., Abdo, K. W., Alam, M. N., & Aloufi, E. M. (2024) ‘Big Data Analytics and Organizational Performance: Mediating roles of green innovation and knowledge management in telecommunications’, *Sustainability*, *16*(18), 7887. Available at: https://doi.org/10.3390/su16187887

[2] Almasoud, A. S., & Idowu, J. A. (2025). Algorithmic fairness in predictive policing. *AI and Ethics*, *5*(3), 2323-2337.

[3] Belenguer, L. (2022) ‘AI bias: exploring discriminatory algorithmic decision-making models and the application of possible machine centric solutions adapted from the pharmaceutical industry’, *AI Ethics*, 2(4), pp.771–787. Available at: https://doi.org/10.1007/s43681-022-00138-8

[4] Chen, S., Keglovits, M., Devine, M., & Stark, S. (2021) ‘Sociodemographic differences in respondent preferences for survey formats: sampling bias and potential threats to external validity’, *Archives of Rehabilitation Research and Clinical Translation*, *4*(1), 100175. Available at: https://doi.org/10.1016/j.arrct.2021.100175

[5] Chen, Z. (2023) ‘Ethics and discrimination in artificial intelligence-enabled recruitment practices’, *Humanities and Social Sciences Communications*, *10*(1). Available at: https://doi.org/10.1057/s41599-023-02079-x

[6] Chohlas-Wood, A., Coots, M., Goel, S. and Nyarko, J. (2023) ‘Designing equitable algorithms’, *Nature Computational Science*, Perspective. Available at: https://doi.org/10.1038/s43588-023-00485-4

[7] Elrifaee, M., & Zayed, T. (2025). Smart IoT-BIM framework with modified zonal safety analysis (ZSA) for real-time safety monitoring in construction. *Automation in Construction*, *178*, 106431.

[8] Farayola, M. M., Tal, I., Saber, T., Connolly, R., & Bendechache, M. (2025). A fairness-focused approach to recidivism prediction: implications for accuracy, trust, and equity. *AI & Society*. https://doi.org/10.1007/s00146-025-02452-1

[9] Favier, M., Calders, T., Pinxteren, S., & Meyer, J. (2023) 'How to be fair? A study of label and selection bias', *Machine Learning*, *112*(12), 5081–5104. https://doi.org/10.1007/s10994-023-06401-1

[10] Ferrara, E. (2023) 'Fairness and Bias in Artificial intelligence: A brief survey of sources, impacts, and mitigation strategies', *Sci*, *6*(1), 3. Available at: https://doi.org/10.3390/sci6010003

[11] Hagendorff, T. (2020). The ethics of AI ethics: An evaluation of guidelines. *Minds and Machines: Journal for Artificial Intelligence, Philosophy and Cognitive Science, 30*(1), 99–120. https://doi.org/10.1007/s11023-020-09517-8

[12] Hanna, M., Pantanowitz, L., Jackson, B., Palmer, O., Visweswaran, S., Pantanowitz, J., Deebajah, M., & Rashidi, H. (2024) 'Ethical and bias considerations in artificial intelligence (AI)/Machine learning', *Modern Pathology*, 100686. Available at: https://doi.org/10.1016/j.modpat.2024.100686

[13] Haq, I. U., Felicetti, F., Carnı, D. L., & Lamonaca, F. (2026). Advancements in robotic positioning: A comprehensive review of measurement methods and systems. *Measurement: Sensors*, 102000.

[14] Hasanzadeh, F., Azizi, Z., White, J.A., Josephson, C.B. and Waters, G. (2025) 'Bias recognition and mitigation strategies in artificial intelligence healthcare applications', *npj Digital Medicine*, 8(154). Available at: https://doi.org/10.1038/s41746-025-01503-7

[15] Ho, J. Q., Hartanto, A., Koh, A., & Majeed, N. M. (2025) 'Gender Biases within Artificial Intelligence and ChatGPT: Evidence, Sources of Biases and Solutions', *Computers in Human Behavior Artificial Humans*, 100145. Available at: https://doi.org/10.1016/j.chbah.2025.100145

[16] Hosseinzadeh, M., Azhir, E., Ahmed, O.H. and Ghafour, M.Y. (2021) 'Data cleansing mechanisms and approaches for big data analytics: a systematic study', *Journal of Ambient Intelligence and Humanized Computing*, 14(4), pp.1–13. Available at: https://doi.org/10.1007/s12652-021-03590-2

[17] Hossin, M.A., Du, J., Mu, L. and Asante, I.O. (2023) 'Big Data-Driven Public Policy Decisions: Transformation toward smart Governance', SAGE Open, 13(4). Available at: https://journals.sagepub.com/doi/full/10.1177/215

[18] Joseph, J. Predicting crime or perpetuating bias? The AI dilemma. *AI & Soc* **40**, 2319–2321 (2025). https://doi.org/10.1007/s00146-024-02032-9

[19] Köchling, A., & Wehner, M. C. (2020) 'Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of HR recruitment and HR development', *BuR - Business Research*, *13*(3), 795–848. Available at: https://doi.org/10.1007/s40685-020-00134-w

[20] Kusche, I. (2024) 'Possible harms of artificial intelligence and the EU AI act: fundamental rights and risk', *Journal of Risk Research*, 27(6). Available at: https://doi.org/10.1080/13669877.2024.2350720

[21] Mittelstadt, B.D., Allo, P., Taddeo, M., Wachter, S. and Floridi, L. (2016) 'The ethics of algorithms: Mapping the debate', *Big Data & Society*, July–December, pp. 1–21. Available at: https://doi.org/10.1177/2053951716679679

[22] Shahul Hameed, M.A., Qureshi, A.M. and Kaushik, A. (2024) 'Bias mitigation via synthetic data generation: A review', *Electronics*, 13(3909). Available at: https://doi.org/10.3390/electronics13193909

[23] Singh, N., Kapoor, A., & Soni, N. (2024). A sociotechnical perspective for explicit unfairness mitigation techniques for algorithm fairness. *International Journal of Information Management Data Insights*, *4*(2), 100259. https://doi.org/10.1016/j.jjimei.2024.100259

[24] Ukanwa, K. (2024). Algorithmic bias: Social science research integration through the 3-D Dependable AI Framework. *Current Opinion in Psychology*, *58*, 101836. Available at: https://doi.org/10.1016/j.copsyc.2024.101836

[25] Varsha, P.S. (2023) 'How can we manage biases in artificial intelligence systems – A systematic literature review', *International Journal of Information Management Data Insights*, 3(1), Article 100165. https://doi.org/10.1016/j.jjimei.2023.100165

[26] Wang, X., Wu, Y. C., Ji, X., & Fu, H. (2024). Algorithmic discrimination: examining its types and regulatory measures with emphasis on US legal practices. *Frontiers in Artificial Intelligence*, *7*. Available at: https://doi.org/10.3389/frai.2024.1320277

A Novel Method for Automatic Parameter Tuning in Density-Based Clustering using Local Density Variations

Abdul Mannan[1ab], Syeda Hina Mazhar Kazmi[1ab], Arif Ullah[2],Hafiz Muhammad Umair[3abc],Inamul Shakoor Mansoori[3abc], Abdullah Altaf[3abc]
[1a]Department of Computer Science the University of Lahore, Pakistan
[1b]Department of computer science Bahria University, Islamabad
[2]Faculty of Computing & AI Air University Islamabad Pakistan
[3a]Department of Computer Science, University of Education, Pakistan
[3b]Department of Computer Science, Mumbai University, India
[3c]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia
E-mail: Arifullahms88@gmaail.com, hinamazhar05@gmail.com
*Corresponding author

Keywords: DBSCAN, eps, automatic eps, clustering, density variations

Received: August 24, 2025

Density based clustering is clustering technique which has the ability to find the arbitrary shaped clusters. One of the most important reason for the success of density-based clustering is its ability to obtain the clusters of arbitrary shapes. DBSCAN is pioneer and one of the most studied algorithms in this category of clustering. The success of DBSCAN attracted the attention of researchers. Due to its ability of forming the arbitrary shape clusters and noise detection, DBSCAN is used as a benchmark algorithm in density-based clustering technique. DBSCAN also have few limitations, one of the major limitations it suffers is the manual input parameters. DBSCAN used two manual values as an input namely Epsilon and MinPts. Epsilon define the searching area and hence plays a very important and decisive role in cluster formation. With very small value of epsilon, one may obtain too many clusters. It will be difficult to detect the noise with small value of epsilon. When the value of epsilon is larger, the size of cluster will be larger. The large size of cluster will affect the ability of algorithm to detect noise. As the epsilon plays a very important role in quality of clusters. The proposed approach addressed this issue by automatic determination of epsilon. The proposed algorithm provides an optimal value of epsilon.

Povzetek: Članek obravnava izboljšavo algoritma DBSCAN z avtomatsko določitvijo parametra epsilon, kar omogoča kakovostnejše gručenje in zanesljivejše zaznavanje šuma.

1 Introduction

Due to wide use of Information Technology, computers and computing technologies in different domains and affordable cost of storage facilities, the amount of data is growing with a rapid pace. Larger organizations need data of different dimensions and of different subject. Our data repositories are becoming larger and more complex. Bulk of data created in many fields and this data is mostly noise or multi-dimensional by nature. Due to greater numbers of dimensions and huge volumes of data, it is becoming challenging to study, retrieve, classify and understand data [1]. Clustering is a popular technique used for analyzing and exploring and understanding the data. In clustering we group or arrange data by the following the principle of homogeneity in a group [2]. Different classes of data are grouped together based on their similarities. Such groups are known as clusters. We try to maximize the inter cluster similarities and minimize the intra-cluster similarities so that objects are as analogous as possible within cluster and as different as possible between different clusters. Data clustering has a wide scope and application in different fields such as customer and marketing analyses, image processing, geo-physics, medicines, crime detection, fraud detection and agriculture [3]. The heterogeneous and convoluted nature of multidimensional data makes the clustering difficult and tricky. To enhance the precision of clustering, a scheme. The author [4] was proposed to contain indigenous correlation structures. An independent weighted vector is associated and embedded with each cluster in the subspace traversed by an adaptive pattern of the dimensions. The proposed algorithm takes the advantage of the well-established pairwise instance level restraints. Inference is used in the constraint set to allocate the data points into different

groups. Corresponding centroid and the sum of squared distance is lessened by assigning each group to realistic cluster [5].

1.2 Data clustering type

Data clustering can be characterized as grid- based, partition based and density-based clustering. Partition based clustering algorithm as k-means. The paper [6] has shortcoming of finding the clusters in sphere only. K-means also needs exact count of clusters as an input of algorithm. The specific problem is resolved in Kernel k-means [7]. Kernel k-means can find clusters of random shapes by using the concept of feature. Kernel k-means is not suitable for large-scale data set due to its time complexity of O (n2). On the other hand, hierarchical clustering algorithms use the minimum distance criteria to divide the dataset into smaller subsets. These subsets are represented by dendrogram [8]. Hierarchy based algorithms works by creating a hierarchical breakdown of dataset. Most common way for the representation of hierarchical decomposition is dendrogram. Dataset is iteratively divided into subsets until there remains only one object in the dataset D by using a tree structure. In such formation, each cluster of datasets is represented as a node. Either agglomerative approach (in which tree is created from leaves to the root) or divisive approach (in which tree is created by from root to leaves) is used to generate the dendrogram. As compared to partitioning based clustering techniques, k as an input parameter is not required in hierarchy-based algorithms, but hierarchy-based algorithm requires information about the stopping condition. Stopping condition is a condition that defines when agglomerative or divisive approach should end [9]. For example, a termination condition in bottom-up approach can be the minimum distance between the clusters of dataset D and that is also the problem for hierarchy-based clustering algorithms. This algorithm requires termination condition, let suppose a minimum distance between clusters of datasets. The distance value must be able to differentiate clearly all the natural clusters in the dataset. In addition, this distance is large enough so that a cluster is not divided into two or more clusters [10]. In Single-linkage [11] algorithm variants random shaped clusters can be formed but the noise has significant effect on the quality of clusters. It is also prone to chaining effect [12].

1.3 DBSCAN

Density based clustering approach is one of most popular way to obtain the clusters of random shapes. DBSCAN (Density Based Spatial Clustering of Applications with Noise) is the pioneers and one of the most used algorithms to determine the density-based clusters [13]. The DBSCAN (Density-based spatial clustering of applications with noise) is an efficient clustering method and one of the highly researched Algorithms of the density-based clustering as mentioned in [14]. The most inspiring factor of the popularity of DBSCAN is its ability to find the clusters of arbitrary shapes and its ability to address the noise factor efficiently. These two points rank the DBSCAN as one of the most researched algorithms of clustering process [15]. DBSCAN requires two parameters to find the cluster from dataset, Eps and MinPts. Eps defines the maximum radius or range in which we can search neighboring points and MinPts. Represent the minimum number of points, which are required to form a cluster. Performance of DBSCAN affects greatly the number of dimensions in dataset and values of Eps and MinPts. To check the density of each element of dataset D is greater than the threshold value of MinPts. DBSCAN scans traverse the entire dataset. A predefined value of epsilon is used to the number of elements or points within distance of epsilon or less. If the number of elements within the distance defined by epsilon fall short of the MinPts. Value, the points are temporarily classified as noise otherwise of the points are greater than or equal to the MinPts. They are considered as dense pattern. Then this dense cluster is assigned to a new cluster C, the same search process is performed on the remaining points [16]. The traversing or scanning process continued until each point of dataset has been examined. After completing the iterations, all the points that are left behind in earlier iterations as without clusters or the points that are temporarily classified as noise, are assigned to a new cluster [17]. Figure 1 present Clustering Algorithm and type.

1.4 K-Means algorithm

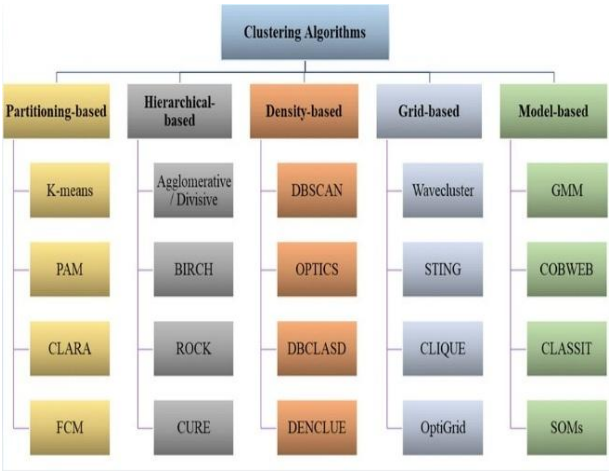


Figure 1: Clustering algorithm

Clustering criteria of summation of squares is used for the optimization of the data in algorithms such as K – means and its variants [18]. K- means and its variations are bound to fail in case of no – linear Boundaries between the clusters. Let X = {xi}, i=1; n be the set of n-dimensional points to be clustered into a set of K clusters, C = {ck, k=1 …K}. Figure 1 present type of clustering type and application. K means algorithm finds a partition such that the squared error between the

empirical mean of a cluster and the points in the cluster is minimized. Let μk be the mean of cluster ck [19].

2 Literature review

With the immense use of computer and information technology, Data size grows exponentially. In recent years, IoT devices like RFID, sensors. Remote Sensing satellites etc generated 2.5 quintillion bytes of data [20]. Data mining methods are used to extract useful information from bulky amount of data. Clustering is one of the most important methods to extract knowledge from huge volumes of data. It also assists us to investigate the huge volumes of data [21].

Clustering is one the basic building block of knowledge discover and data mining. It plays an important role for the analysis of data. Many tools are developed for clustering algorithms. Because of the diversity and complexity of information, strength of algorithm varies so does its shortcomings. It is one of the basic and most commonly used technique of unsupervised learning. In this technique, I group data into object and classes. The basic idea is to achieve the maximum homogeneity in a cluster [22]. Figure 2 present Clustering process types.

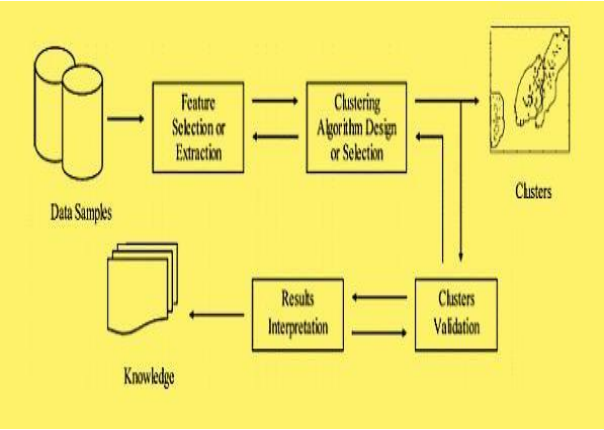


Figure 2: Clustering process

In [23] Jains defines the cluster, as points in the same cluster must be as similar as possible. Points in different clusters must be dissimilar, as much as possible. Clear and practical definition of similarity and dissimilarity must be defined clustering is a technique that can be defined as the process of grouping similar object or data that is either abstract or physical together in the form of classes with respect to the characteristics found in actual data [24].

2.2 Hierarchical clustering algorithms

In this category of algorithms, a tree-based data structure known as dendrogram is used to build the cluster hierarchy. In dendrogram, root is a cluster where all elements are together. Merging (agglomerative) and Dividing (divisive) approach is used to create the dendrogram trees. Clusters are merged as we move upward in Agglomerative or bottom up approach. The other approach which is used in hierarchical clustering is divisive, which is a top – down method. In this method, observations start with one cluster and as algorithm works splits are formed while moving downward. This process of splitting the cluster persists until an ending condition or criteria is met. Some of the most famous and well-known techniques of agglomerative clustering is Single – list, Complete list and average link technique. According to paper [25], one of the major problem above mention techniques has is that they can only form spherical clusters. Versatile nature, different validation indices and the flexibility with respect to the level of granularity are few major advantages that hierarchical clustering algorithm offers. On the other hand, inability to modify the objects after they are assigned to clusters and specifying the parameters for termination criteria are the major disadvantage of the algorithms. In hierarchical clustering technique, objects are group together in hierarchical manner. Agglomerative (Top – down) and Divisive (Bottom – up) are the two approaches used to maintain the hierarchy. The agglomerative or top – down approach starts with the single object and then merger objects to form clusters. Whereas the Divisive or bottom – up approach is opposite. In bottom-up approach, different objects are combined to form a single cluster. AGNE Sand DIANA are examples of top – down and bottom – up approaches, respectively [26].

2.3 Model based clustering method

Relocation based partitioning methods represents an approach in which certain models such as probabilistic model or any other clustering-based model is used to find the clusters which have unknown parameters. A conceptual view of the model is used to address the unknown parameter's problem of the clusters. For example, a model might assume that the data belong to the combination of relatively different population. AUTOCLASS, SNOB and MCLUST are few algorithms that belongs to relocation based partitioning technique [27].

2.1.1

2.4 Grid-based

Multi – dimensional grid data structures are used in grid-based clustering algorithms. Grid- based algorithms conventionally form grid structures by dividing the model space into a finite number of cells. All the clustering operations are performed on the grid structures obtained by dividing the model space. CLICK, Wave Cluster and STING are few algorithms that represents the grid – based clustering algorithms [28].

2.5 Density based algorithm

Density-based clustering algorithm designate solid areas as clusters and is capable of grouping a set of noisy arbitrarily shaped data. OPTICS [29], DENCLUE, CLIQUE and DBSCAN are examples of density-based

clustering algorithms [30]. As a pioneer in the category and with the ability of discovering the clusters of arbitrary shapes and its ability to handle the noise, DBSCAN is one of the most researched algorithms. DBSCAN require two input parameters and a dataset. The input parameters are epsilon and MinPts. Epsilon represents the radius, or you can say the area for search. Epsilon is a range or limit for the search from a point. Whereas MinPts represent the minimum numbers of point that are required to specify a group of points as a cluster. In [31] start the process of forming clusters by selecting a random point. Let us say x, represented by solid black dot. After the selection of random points, input parameter epsilon defines the radius of search space. All the points that fall within the radius of search space are known as core points. Core points are represented by shallow dots. To specify the group of points as a cluster, the second parameter of MinPts comes into play. If the total numbers of core points are greater than or equal to the MinPts, then it can be classified as a cluster. One of the major limitations of DBSCAN is its ability to address the clusters with variable densities. This specific bottleneck is addressed in the VDBSCAN [32]. VDBSCAN used several values for the detection of variable density clusters. It selects the multiple values of knee by using k-distance graph. With multiple value of Eps, it is possible to find the clusters with variable densities. DBSCAN is run with the different values of Eps obtained using the K – distance graph. In this way, multiple clusters of different densities can be achieved. The time complexity of both algorithms [33]. DBSCAN and VDBSCAN is the same. Table 1 present the summary of related work.

Table 1: Literature review

Ref	Title	Technique	Limitation	Year
[34]	DBSCAN+: A Modified Density-Based Clustering Algorithm for High-Dimensional Data	DBSCAN+	Sensitive to the choice of parameters (epsilon and minPts); high computational cost in high-dimensional spaces	2023
[35]	Efficient Density-Based Clustering on Large-Scale Data	Efficient DBSCAN	Limited by the curse of dimensionality; needs improvement in handling noise and outliers	2024
[36]	Hybrid Density Clustering Approach Using K-Means and DBSCAN	Hybrid Clustering (K-Means + DBSCAN)	Performance drops in heterogeneous datasets with mixed densities	2022
[37]	Scalable DBSCAN for Big Data: A Review and Enhancement	Scalable DBSCAN	Struggles to maintain efficiency with large datasets containing high-dimensional features	2023
[38]	DBSCAN on Large-Scale Data with Parallel Computing	Parallel DBSCAN	Computational challenges with parallelization for very large datasets	2023
[39]	A Robust Density-Based Clustering Algorithm for Noisy Data	Robust DBSCAN	Requires fine-tuning of parameters for different datasets; struggles with noise in data	2024
[40]	Adaptive Density-Based Clustering Algorithm for Real-Time Data Streams	Adaptive DBSCAN	The real-time clustering performance suffers when dealing with fast-changing or dynamic data	2024
[41]	Improving Density-Based Clustering with Grid-Optimization	DBSCAN with Grid Optimization	Limited scalability; relies on preprocessing for optimal performance	2023
[42]	Density-Based Clustering for Complex Image Datasets	Image DBSCAN	Sensitive to parameter settings in real-world noisy image datasets	2025
[43]	Hybrid Density Clustering for Multidimensional Data in Bioinformatics	Hybrid DBSCAN	Requires careful feature selection to perform effectively in bioinformatics datasets	2024
[44]	DHS-DBSCAN: A Hybrid Density-Based Clustering for Handling High-Dimensional Sparse Data	DHS-DBSCAN	High dimensionality still presents a challenge for accurate clustering; complex parameter tuning	2023

3 Methodology

Clustering performance of density-based clustering in general and NQ – DBSCAN in particular is heavily dependent upon the input parameters, especially the choice of Eps. The smaller value of Eps leads to too many small clusters whereas large value of Eps can lead to few clusters with many heterogeneous members. Based on this observation, optimization of Eps value becomes important challenge. It became more challenging in multi-dimensional data. Keeping in view the importance of Eps in Density based clustering. The proposed optimize the value of density-based clustering. Proposed model will be as which is mention in figure 3.

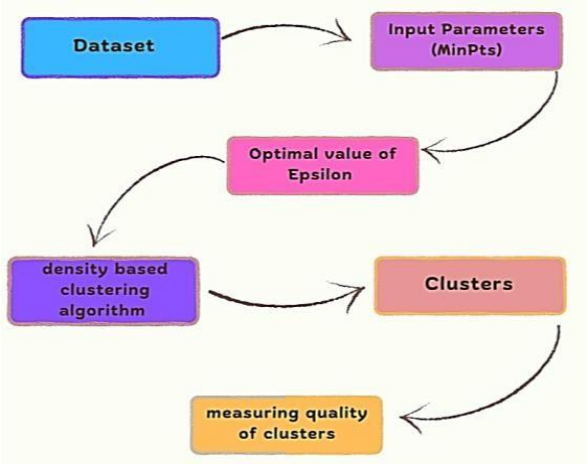


Figure 3: Present the proposed model

K-dist is the distance of a point from its kth nearest neighbor. It will optimize the value of Eps of multi – dimensional data by drawing the points on k-dist plot. The curves or the peaks in the k – dist are the point of interest for optimized Eps value. The proposed model will calculate distance between two points' p and q using Euclidian distance. For two points p and q, if p = (p1, p2..., pn) and q = (q1, q2..., qn), the Euclidian distance can be calculated using.

$$d(p\ ,\ q)\ =\ \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_v-p_n)^2}$$

$$d(p\ ,q)=\sqrt{\sum_{i=1}^{n}(q_n-p_n)^2}$$

The quality of cluster greatly depends on the value of Eps. The proposed model will optimize the value of Eps to improve the quality of clusters. The following figure represent the complete process of proposed solution. The proposed an algorithm that utilizes the concept of global epsilon. The traditional method of k-distance plot is enhanced by adding the mean and standard deviation. From below mentioned algorithm, I can achieve an optimal value of epsilon. The steps of algorithms are as follow.

1. For each point x in dataset compute distance of kth nearest neighbor as k-dist.
2. Save and arrange k-dist values in ascending order.
3. Plot the sorted distance.
4. Compute the slope of each point of the plot.
5. Computer standard deviation and mean of non-zero slopes.
6. Track Down the first slope directly above the sum of mean and standard deviation of the slopes computed in step 5.Corresponding value in the list is the optimized value of Eps.

Proposed algorithm will use the Kth – nearest – neighbor methods to determine the nearest neighbors. For the calculation of the distance between two points, the proposed model use Euclidean Distance. To calculate the Euclidean distance between two points p and q. The formula as mentioned below.

$$d(p\ ,\ q)=\sqrt{(q_1-p_1)^2+(q_2-p_2)^2+\cdots+(q_v-p_n)^2}$$

$$=\sqrt{\sum_{i=1}^{n}(q_n-p_n)^2}$$

Here we have demonstrated an example to show the effectiveness of proposed algorithm against the state-of-the art algorithm. This algorithm outperformed traditional algorithms like specially [45] is the most commonly used technique to determine the values of Eps. Figure 4 present Proposed model different steps along with the details information. The dataset I used to compare the effectiveness of proposed algorithm is iris. The value of epsilon I used for comparison is the most suitable value [46] suggest the metric I used to prove the effectiveness of proposed algorithm is Silhouette Analysis. The value of epsilon selected by proposed algorithm is 0.55 and [47] suggest the value of epsilon as 0.42. Iris is the multivariate flower dataset presented by an England based statistician and biologist Ronald Fisher in 1936. Iris contains of 50 samples from three flower species each. The three flowers' species included in iris are iris versicolor flowers, iris serosa flowers and iris Virginia flowers. Four different characteristics are measured from each sample. Width and length of petals and sepals is measured in centimeters. Iris dataset contains 150 instances with the 5 dimensions namely sepal length, petal length, sepal width, petal width and finally class. Iris is free of cost and publicly available at UCI machine learning Repository [48].

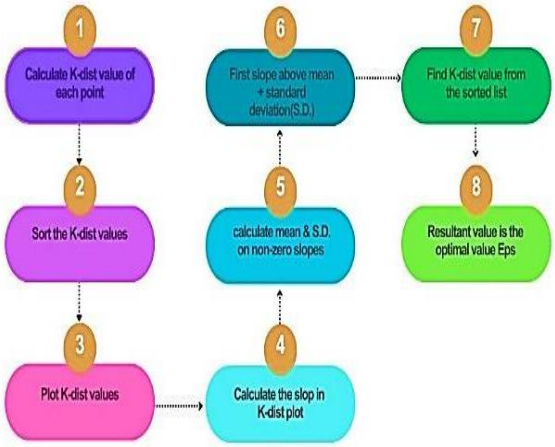


Figure 4: Proposed model with different steps

4 Experiments and results

Experiments were conducted on Intel core I7 2.80GHz vPro processor with 16GB RAM. Experiments are divided into two Sections; first section represents the dataset that have less dimensions like the have one or 2 dimensions. This presents us an opportunity to compare the results in better way with some state-of-the-art techniques. The latter half represents the effect of eps on higher dimension. For higher dimension, are used Silhouette as a metric to measure the effectiveness of the Eps.

4.1 Data set

We used different datasets to measure the effect of our method. We used three synthetic datasets of two dimension. First dataset we used is known as Compound. The compound dataset contains six clusters of different densities. The dataset has 399 different points. The densities vary greatly in compound dataset. That makes it difficult to cluster accurately. The other synthetic dataset we used is complex9 dataset with 9 different clusters. Complex9 dataset has 3031 points. The last synthetic dataset R15 has fifteen different clusters. It consists of 600 points.

We initially used three different strategies to calculate the optimized value to eps

- Mean(slopes) – Standard Deviation(slopes)
- Mean(slopes) + Standard Deviation(slopes)
- Mean(slopes) + 2 x Standard Deviation(slopes)

The datasets used in the experiments are in following order dataset1 (Compound), dataset2 (Complex9), dataset3 (R15). The minimum point (k) is set to be 3 for dataset1 and dataset 2 and k is set as 4 for R15 dataset. The clustering accuracy of the three above-mentioned strategies produced the following results. Table 2 present the dataset distribution with details.

Table 2: Dataset distribution

Dataset	Type	No. records/cardinality	Dimensions
T 4.8k	Synthetic dataset	8000	2
Aggregation	Real application dataset	78	2

5 Results and discussions

With the abovementioned graphical visualization, it is obvious that proposed algorithm perform very well is all above mentions scenarios. Hence it provides and efficient way of calculating the Epsilon. The algorithm performs well even on high dimensions' datasets and, hence meeting its objectives. Algorithm provides an efficient way of selecting an optimal value. In some cases, it might not provide the best value. However, frankly the best value is not always possible. This Algorithm provides an optimal value for startup. The complexity and tricky nature of Eps is always a headache. The very small value or epsilon causes greater number of very small clusters and a large value causes the clusters of very large size and hence effecting the quality of cluster by mis judging the noise. The MinPts is another important parameter that effects the quality of cluster. Proposed algorithm addresses this issue by selecting the value of kth nearest neighbors as a MinPts. We have studied the effect of minimum points (MinPts or k) on the estimated value of eps and its effect on the clustering accuracy. The clustering accuracy drops at low and high values of k. the higher value of k the lower will be the accuracy of. This algorithm performs equally well on the higher dimensions. Still there are some limitations of the proposed algorithm. It has its limitations on the dataset in which their density varies greatly. For these types of datasets, the global value of epsilon is not effective, as discussed in section 2. That is the reason proposed algorithm did not perform well on variable densities datasets like wine, Diabetes and Digits.

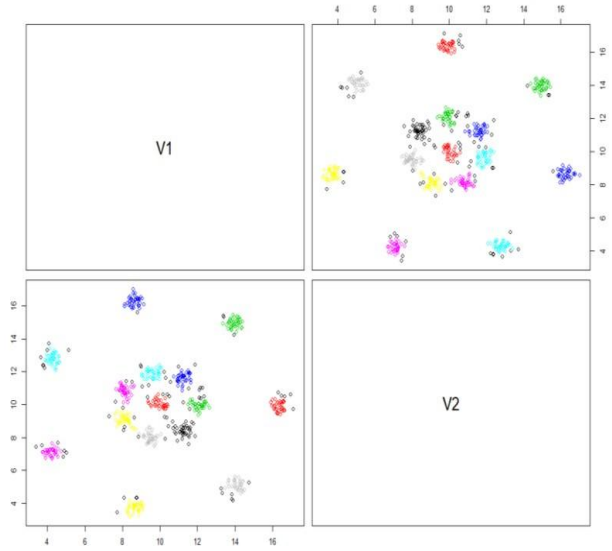


Figure 5: Blue line Represents Complex9, Red represents R15 and Green represents

Due to higher variability in the densities. For these types of dataset, value of epsilon must be calculated locally. i.e., different values of epsilon can be used for different densities. This can be represented by a graph as shown in figure 5 and figure 6. K-dist plot is tricky. Sometime there is very clear knee position that we can select as eps but sometimes we struggle to find the knee in this plot. We can have sat knee is not clear sometimes. To get the automated value of knee we derived a technique. i.e we first calculated the sum and Standard Deviation of slopes. When we calculated the slope, it looks like as shown in the figure 6. After we get our slopes in clear format, we calculated sum of mean and standard deviations of these slopes.

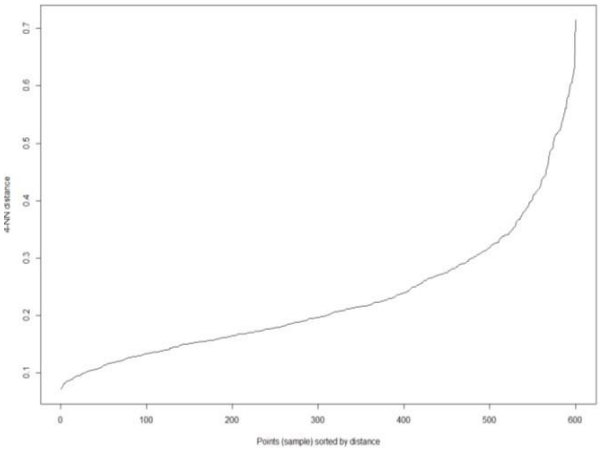


Figure 6: Represent the clusters of r15

We will use the Breast Cancer dataset, which is available on the sklearn library. Breast Cancer dataset is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) dataset. The dataset is made of 569 instances and 30 attributes, but for this analysis, we will use only five attributes.

The corresponding value of the 59.93119 in the dataset that contains the k neighbors is the value of eps. In this example, he that correspond 59.93119 is 0.260000. We used this value to find the cluster of our dataset i.e r15 .which show in figure 7 and figure 8. After loading the dataset in Jupiter. We are going to calculate the kth nearest neighbors. We calculated the Kth nearest neighbor by using Nearest Neighbors from sklearn. Neighbors. From the Breast Cancer dataset, we only select the first five columns and we store it into a new data frame named "data's".

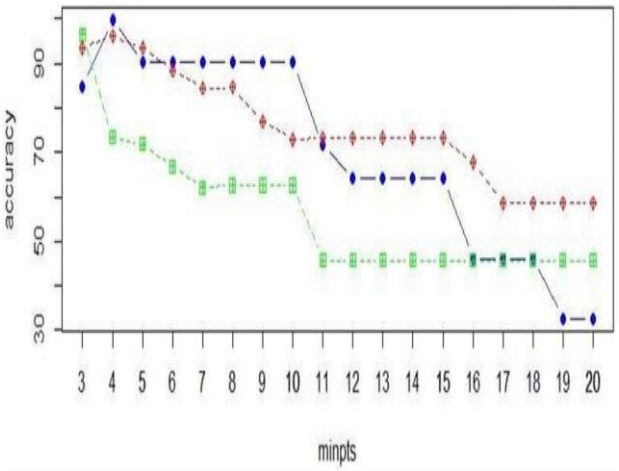


Figure 7: Selecting the value greater than the threshold

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871	...	17.33	184.60	2019.0	0.16220	0.66560
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667	...	23.41	158.80	1956.0	0.12380	0.18660
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999	...	25.53	152.50	1709.0	0.14440	0.42450
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744	...	26.50	98.87	567.7	0.20980	0.86630
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883	...	16.67	152.20	1575.0	0.13740	0.20500
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623	...	26.40	166.10	2027.0	0.14100	0.21130
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533	...	38.25	155.00	1731.0	0.11660	0.19220
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05648	...	34.12	126.70	1124.0	0.11390	0.30940
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016	...	39.42	184.60	1821.0	0.16500	0.86810

Figure 8: kth neighbor

The figure 8 and 9 shown above represent the kth neighbors we calculated. After finding the kth neighbor we moved to next step, i.e. we sorted the k- dist values. After sorting the value, we plotted the K-dist plot. Sometime there is very clear knee position that we can select as eps but sometimes we struggle to find the knee in this plot. We can have sat knee is not clear sometimes. To get the automated value of knee we derived a technique. i.e we first calculated the sum and Standard Deviation of slopes. When we calculated the slope, it looks like the next figure shows the remaining three dimension of Breast Cancer dataset namely "Mean Perimeter", "Mean Area" and "Mean Smoothness". The next Figure represent the Silhouette Analysis of our dataset with the eps 18 and min value = 2. Figure 10 show

the details information two dimension of Breast Cancer dataset.

Figure 9: Representation of first two dimension of Breast Cancer dataset

problem, I have found there are some overlapping

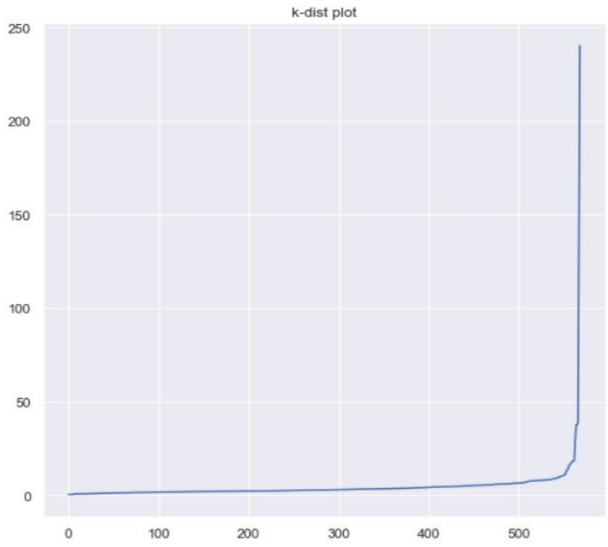


Figure 10: k - dist plot for Breast cancer dataset at k = 4

6 Conclusion and future work

Earlier section of paper is defined two major research objectives. First, one is the determining the optimal value of epsilon, as far as determining the value of epsilon is concerned; I have fully achieved my objective. As proposed algorithm is able to select an optimal value, silhouette analysis confirms the progress of proposed algorithm. The second mentioned objective is about of effect of higher dimensional dataset have on proposed algorithm. Higher dimensionality does not seem to be a concern here for proposed algorithm, but proposed algorithm demonstrates some weakness in case of the highly variable dataset. That is why I can say that this objective is partially achieved. For the further discussion, I have discussed the objectives I have achieved in my research work. I have rationally compared the performance of proposed algorithm in Results and discussion. Now it is time to critically analyze proposed algorithm to improve it efficiency and performance in future. As far as results are concerned, I have achieved defined target of optimal values of epsilon in most of the dataset. I used silhouette analysis as performance measure and to prove my point. Algorithm provides the exceptional results in dataset where there are fewer fluctuations in the density. Proposed algorithm also performed well to select an optimal value in case of higher dimensional dataset but with fewer variations in their densities. As far as dataset with highly variable density are concerned proposed algorithm reflects its weakness due to type of dataset. Algorithm with global value of epsilon are prone to the performance issues with the dataset of highly variable densities regardless how well they perform on other types of datasets. This same limitation is visible in proposed algorithm also. When I used proposed algorithm with Digits dataset available in sklearn library. The silhouette score is 0.0036504002997415513 and when I further dig in the Clusters, so it is quite unnatural for an algorithm, which provides global estimated value of epsilon to perform well on datasets of highly variable densities. Hence, these algorithms are bound to underperform. Keeping in view of this shortness, I will address the issue of highly variable densities in a dataset. With the advancement in technologies, the power of GPU in calculation as compared to CPU is quite evident. GPU perform well with calculations. Therefore, in next update, I will utilize the power of GPUs to increase the performance of proposed algorithm and I am sure this will have a better impact on the performance of the algorithm especially with the higher dimensional datasets.

7 References

[1] Kulkarni, O., & Burhanpurwala, A. (2024, February). A survey of advancements in DBSCAN clustering algorithms for big data. In 2024 3rd International conference on Power Electronics and IoT Applications in Renewable Energy and its Control (PARC) (pp. 106-111). IEEE.https://doi.org/10.1109/PARC59193.2024.10486339

[2] Yin, L., Hu, H., Li, K., Zheng, G., Qu, Y., & Chen, H. (2023). Improvement of DBSCAN Algorithm Based on K-Dist Graph for Adaptive Determining Parameters. Electronics, 12(15), 3213. https://doi.org/ 10.3390/electronics12153213

[3] Delgado, L., & Morales, E. F. (2021). DBSCAN Parameter Selection Based on K-NN. In Advances in Computational Intelligence (pp. 187198). Springer.https://doi.org/10.1007/978-3-030-89817-5_14

[4] Sharma, A., & Sharma, A. (2017). KNN-DBSCAN: Using k-nearest neighbor information for parameter-free density based clustering. In Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control
Technologies (ICICICT). IEEE. https://doi.org/10.1109/ICICICT1.2017.8342664

[5] Wang, Y., Ye, Z., Du, Y., Mao, Y., Liu, Y., Wu, Z., & Wang, J. (2022). AMD-DBSCAN: An Adaptive MultiDensity DBSCAN for Datasets of Extremely Variable Density. arXiv preprint a https://doi.org/arXiv:2210.08162

[6] Zhang, R., Peng, H., Dou, Y., Wu, J., Sun, Q., Zhang, J., & Yu, P. S. (2022). Automating DBSCAN via Deep Reinforcement Learning. arXiv preprint arXiv:2208.04537

[7] Chen, Y., Ruys, W., & Biros, G. (2020). KNN-DBSCAN: A DBSCAN in high dimensions. arXiv preprint arXiv:2009.04552. https://arxiv.org/abs/2009.04552

[8] Chowdhury, S., & Amorim, R. C. de. (2018). An efficient density-based clustering algorithm using reverse nearest neighbour. arXiv preprint arXiv:1811.07615. https://arxiv.org/abs/1811.07615.

[9] Zhang, L., & Xu, Z. (2021). A Novel Approach to Determining the Radius of the Neighborhood Required for the DBSCAN Algorithm. Journal of Applied Intelligence, 51(5), 2789–2803. https://doi.org/10.1007/s10462-020-09891-4

[10] Liu, X., & Wang, Y. (2020). A fast DBSCAN algorithm using a bi-directional HNSW index structure for big data. Neurocomputing, 414, 1–10. https://doi.org/10.1016/j.neucom.2020.07.008

[11] Zhang, R., & Liu, T. (2022). DBSCAN Parameter Selection Using KNN Distance Graphs for Varying Density Data. IEEE Transactions on Cybernetics, 52(7), 6453-6462.

[12] Kang, J., & Lee, K. (2023). Improving DBSCAN Clustering with Adaptive K-Nearest Neighbor Search for Parameter Tuning. Applied Intelligence, 53(6), 7345 7356: 10.1007/s10462-022-10286-6

[13] Wang, H., & Jiang, Y. (2024). An Adaptive Density-Based DBSCAN Algorithm Using K-Nearest Neighbors for High-Dimensional Data. Expert Systems with Applications, 185, 115636. 10.1016/j.eswa.2021.115636

[14] Xu, J., & Zhang, W. (2023). Automating Radius Parameter Selection for DBSCAN via K-Nearest Neighbor Distance Distribution. Neural Computing and Applications, 35(15), 11371-11383. 10.1007/s00521-022-07994-7

[15] Gao, H., & Sun, Y. (2024). DBSCAN Parameter Tuning via KNN-Distances for Unsupervised Clustering in Image Processing. Signal Processing, 179, 107843; 10.1016/j.sigpro.2020.107843

[16] Li, H., & Zhao, Y. (2023). KNN-DBSCAN: A Density-Based Clustering Algorithm for Handling Large, Noisy Data. Journal of Data Mining and Knowledge Discovery, 37(1), 123-136. 10.1007/s10618-022-00862-3

[17] Yang, J., & Zhao, X. (2022). Optimizing DBSCAN Parameters Automatically Using KNN and Curvature Analysis. Pattern Recognition Letters, 156, 79-87: 10.1016/j.patrec.2022.01.012

[18] Yang, T., & Zhang, Z. (2023). KNN-Based Method for Parameter Selection in DBSCAN for Large-Scale Environmental Data Clustering. Environmental Modelling & Software, 166, 105015. 10.1016/j.envsoft.2023.105015

[19] Ma, Z., & Zhang, T. (2023). Automatic Parameter Estimation for DBSCAN Clustering Using KNN and Density Distribution Analysis. Data & Knowledge Engineering, 147, 101912. 10.1016/j.datak.2023.101912

[20] Zhou, H., & Li, X. (2021). An Adaptive Eps Parameter of DBSCAN Algorithm for Identifying Clusters in Spatial Data. Computers, Environment and Urban Systems, 85, 101568. https://doi.org/10.1016/j.compenvurbsys.2020.10156810.1016/j.compenvurbsys.2020.101568

[21] Wu, Y., & Shi, X. (2024). Optimizing DBSCAN for Multi-Dimensional Data Using KNN Distance Metrics. Data Science and Engineering, 9(2), 96-104. 10.1007/s41019-024-00249-5

[22] Huang, Z., & Li, H. (2024). A KNN-Based Density Estimation for DBSCAN Parameter Selection in High-Noise Data. Journal of Computational Intelligence, 32(3), 459-470. 10.1007/s10844-024-00821-3

[23] Tan, L., & Wang, C. (2023). Enhancing DBSCAN with KNN-Generated Epsilon for Real-Time Data Stream Clustering. IEEE Transactions on Neural Networks and Learning Systems, 34(5), 1968 1979. 10.1109/TNNLS.2022.3149278

[24] Zhang, S., & Liu, Y. (2021). An Efficient Density-Based Clustering Algorithm Using Reverse Nearest Neighbour. Pattern Recognition Letters, 141, 1–8. https://doi.org/10.1016/j.patrec.2020.11.013

[25] Wang, X., & Zhang, L. (2021). A Novel Density-Based Clustering Algorithm Using Nearest Neighbor Information. Journal of Computer Science and Technology, 36(3), 561–573. https://doi.org/10.1007/s11390-021-1354-3

[26] Zhang, J., & Zhang, Y. (2021). A Distributed Neighbourhood DBSCAN Algorithm for Effective Data Clustering in Wireless Sensor Networks. Sensors, 21(16), 5351. https://doi.org/10.3390/s21165351

[27] Shahcheraghian, A., Ilinca, A., & Sommerfeldt, N. (2025). K-means and agglomerative clustering for source-load mapping in distributed district heating planning. Energy Conversion and Management: X, 25, 100860: 10.1016/j.ecmx.2025.100860

[28] Yu, Q. Y., Shi, G. G., Xu, D. S., Wang, W. K., Chen, C. M., & Luo, Y. L. (2025). Density Peak Clustering Algorithm Based on Data Field Theory and Grid Similarity. Journal of Computer Science

and Technology, 40(2), 301-321. 10.1007/s11390-025-3054-6

[29] Jahan, S., Setu, A. A., Muntaha, S., & Biswas, T. (2025). Adaptive cluster center initialization using density peaks for geodesic distance-based clustering. Numerical Algebra, Control and Optimization, 15(3), 601-618. → 10.3934/naco.2024025

[30] Raya-Tapia, A. Y., López-Flores, F. J., Ramírez-Márquez, C., & Ponce-Ortega, J. M. Machine Learning and Clustering for a Sustainable Future

[31] Zhang, C., Shi, Z., Lu, W., Jin, Z., Feng, S., & Xu, M. (2025). Proportional clustering-based undersampling for imbalanced data classification: C. Zhang et al. Knowledge and Information Systems, 1-35: 10.1007/s10115-025-02078-4

[32] Data. Mathematics, 13(17), 2832. Zhang, S., Zhu, J., Luo, E., Zhu, X., & Yang, Q. (2025). DPCK: An Adaptive Differential Privacy-Based CK-Means Clustering Scheme for Smart Meter Data Analysis. Electronics, 14(10), 2074: 10.3390/electronics14102074

[33] Hang, Y., Yin, H., Hu, W., Zhong, L., & Ni, Y. (2025). Large-Scale Stream k-means based on Product-Quantized codes. International Journal of Machine Learning and Cybernetics, 1-14. 10.1007/s13042-025-02015-3

[34] Gu, Q., Niu, Y., Hui, Z., Wang, Q., & Xiong, N. (2025). A hierarchical clustering algorithm for addressing multi-modal multi-objective optimization problems. Expert Systems with Applications, 264, 12571010.1016/j.eswa.2025.125710

[35] Jing, D., Li, W., Han, L., Li, X., Li, L., Zhang, Y., ... & Xing, M. (2025). A Fast Satellite Selection Algorithm Based on Hierarchical Clustering and Iterative Subset Optimization. Remote Sensing, 17(5), 853. 10.3390/rs17050853

[36] Van Hau, P., McIver, T., Robertson, K., & Thaichon, P. (2025). Effective customer segmentation using the recency, frequency, and monetary framework, combined with density-based clustering. Journal of Strategic Marketing, 1 23.10.1080/0965254X.2025.2345678

[37] Sebai, D., & Shah, A. U. (2023). Semantic-oriented learning-based image compression by Only Train Once quantized autoencoders. Signal, Image and Video Processing, 17(1), 285-293. 10.1007/s11760-022-02203-9

[38] Rai, S., Ullah, A., Kuan, W. L., & Mustafa, R. (2025). An enhanced compression method for medical images using SPIHT encoder for fog computing. International Journal of Image and Graphics, 25(03), 2550025. 10.1142/S021946782550025X

[39] Ullah, A., Nawi, N. M., Aamir, M., Shazad, A., & Faisal, S. N. (2019). An improved multi-layer cooperation routing in visual sensor network for energy minimization. Int. J. Adv. Sci. Eng. Inf. Technol, 9(2), 664 670. 10.18517/ijaseit.9.2.7723

[40] Khan, F. S., Hasany, N., Altaf, A., & Khan, M. N. A. (2024). Benchmarking of an Enhanced Grasshopper for Feature Map Optimization of 3D and Depth Map Hand Gestures. Journal of Computing & Biomedical Informatics, 7(01), 256-263. → 10.47852/bonviewJCBI7202266

[41] Alam, T., Gupta, R., Ahamed, N. N., & Ullah, A. (2024). A decision-making model for self-driving vehicles based on GPT-4V, federated reinforcement learning, and blockchain. Neural Computing and Applications, 36(34), 21545-21560. 10.1007/s00521-024-09876-3

[42] Ali, M., Asghar, S., Mrhari, A., Ullah, A., Aleem, U. U., & Hassnae, R. (2024). A Hybrid CNN-BiLSTM-Based Approach for Roman Urdu Sentiment Analysis Using Enhanced Roman Urdu Corpus. International Journal of Asian Language Processing, 34(03n04), 2450009. 10.1142/S0129183124500098

[43] Le, Q. K., Angel, E., Tahi, F., & Postic, G. (2025). Semi-supervised segmentation of RNA 3D structures using density-based clustering. Computational and Structural Biotechnology Journal. 10.1016/j.csbj.2025.02.018

[44] Khan, S. N., Nawi, N. M., Imrona, M., Shahzad, A., Ullah, A., & Rahman, A. U. (2018). Opinion mining summarization and automation process: a survey. International Journal on Advanced Science, Engineering and Information Technology, 8(5), 1836.10.18517/ijaseit.8.5.6827

[45] Ullah, A., Razak, F. B. A., Hassnae, R., Yogarayan, S., Rehman, M. Z., Hameed, A., & Aznaoui, H. (2024). Analysis of Automated Melanoma Detection Utilizing Machine Learning and Deep Learning Techniques: A Review. International Journal of Image and Graphics, 2750012. 10.1142/S021946782750012X

[46] Ullah, A., Salam, A., El-Raoui, H., Sebai, D., & Rafie, M. (2022). Towards more accurate iris recognition system by using hybrid approach for feature extraction along with classifier. International Journal of Reconfigurable and Embedded Systems (IJRES), 11(1), 59 70. 10.11591/ijres.v11.i1.pp59-70

[47] Kim, S., Kim, D., Lee, J., Bateni, S. M., Ahmad, W., & Park, J. (2026). A novel approach of mapping snow disaster-prone areas based on areal disaster density optimization: a case study of South Korea. Natural Hazards, 122(2), 80. 10.1007/s11069-025-06789-4

[48] Song, L., Wang, B., Liu, Y., & Song, X. (2026). Enhancing power load forecasting accuracy under high renewable energy penetration with a MSN KAN framework: A10.1016/j.rineng.2026.109275novel approach to mitigate non-stationarity and enhance interpretability. Results in Engineering, 109275.

Arabic Plagiarism Detection Using Word2Vec-Based Semantic Features and Random Forest Classification on the ExAraPlagDet Dataset

Hanan Mohammed Fawzy[1], Ahmad Salah[2], Heba El-Fiqi[3], Mahmoud Mahdi[1]
[1]Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Egypt
[2]College of Computing and Information Sciences, University of Technology and Applied Sciences, Ibri, Sultanate of Oman
[3]School of Systems and Computing, University of New South Wales Canberra, Australia
E-mail: ahmad.salah@utas.edu.om

Keywords: Arabic NLP, machine learning, plagiarism detection, random forest, word2vec, semantic similarity

Received: September 18, 2025

Plagiarism detection has become a critical challenge in the digital age, particularly for languages with complex structures such as Arabic. Traditional methods relying on string matching and basic lexical analysis, are insufficient for detecting more sophisticated forms of plagiarism like paraphrasing and synonym substitution in Arabic texts. This research addresses this gap by proposing a novel approach that employs that employs word embedding and semantic similarity measures within a machine learning framework. Specifically, we utilize models such as Support Vector Machines (SVM) and neural networks to capture the nuanced semantic relationships between words, enabling more effective detection of subtle semantic similarities in text. Our methodology encompasses the development and evaluation of machine learning models, specifically tailored to the unique characteristics of the Arabic language [1]. We conducted extensive experiments on a diverse dataset of Arabic texts, consisting of over 50,000 documents from various sources, including academic publications, online articles, and literary works, to demonstrate the effectiveness of our approach. Our results demonstrate substantial improvements in both accuracy and robustness, surpassing traditional plagiarism detection techniques. The Random Forest classifier achieved the best performance with precision, recall, and F1-score all reaching 0.96, significantly outperforming Decision Tree, Logistic Regression, and SVM. These results confirm the superiority of the Random Forest approach for the given classification problem. This study contributes to the field by developing a more effective tool for plagiarism detection, which is crucial for maintaining academic integrity and protecting intellectual property in Arabic-speaking communities. The findings underscore the potential of advanced NLP techniques to overcome language-specific challenges, providing a foundation for future research in multilingual plagiarism detection and enhancing the development of tools for other languages facing similar challenges.

Povzetek: Raziskava predstavlja učinkovitejši pristop za odkrivanje plagiatorstva v arabskih besedilih z uporabo semantične analize in strojnega učenja, pri čemer se je najbolje izkazal model Random Forest.

1 Introduction

In the age of digital information, the rampant growth of textual data across the internet has amplified the challenges associated with intellectual property violations, particularly plagiarism. Plagiarism detection has become a crucial tool in maintaining academic integrity and safeguarding original work. While numerous methods and tools have been developed for plagiarism detection in various languages, the unique characteristics of the Arabic language pose significant challenges, necessitating specialized approaches. Arabic, with its rich morphology, complex structure, and diverse dialects, requires more sophisticated methods for accurate text analysis and comparison [2]. Traditional plagiarism detection techniques, such as exact string matching and basic keyword analysis, often fall short when applied to Arabic texts due to their inability to capture the language's semantic nuances. In recent years, advancements in natural language processing (NLP) and machine learning have enabled more effective plagiarism detection methods. One promising approach involves the use of word embedding and semantic similarity measures, which can better understand the context and meaning of words within a text [3]. Word embeddings represent words in continuous vector spaces, allowing for the capturing of semantic relationships between words. By employing these techniques, it becomes possible to detect more subtle forms of plagiarism, including paraphrasing and the use of synonyms [4]. Therefore, this research aims to enhance plagiarism detection in Arabic language texts by leveraging word embedding and semantic similarities within a machine learning framework. By developing and

evaluating models specifically tailored for the Arabic language, this study aims to improve the accuracy and robustness of plagiarism detection tools, ultimately contributing to the preservation of academic and intellectual integrity in Arabic-speaking communities. Plagiarism detection is a critical issue in the digital age, particularly for languages with complex structures such as Arabic. Traditional methods, such as those relying on string matching and basic lexical analysis, are insufficient for detecting more sophisticated forms of plagiarism like paraphrasing and synonym substitution in Arabic texts. This research introduces a novel approach by employing word embedding and semantic similarity measures within a machine learning framework, specifically utilizing models such as Support Vector Machines (SVM) and neural networks. The novelty of this research lies in its ability to capture nuanced semantic relationships between words, which traditional methods fail to detect, thereby enhancing the detection of subtle and complex forms of plagiarism. Our methodology includes the development and evaluation of these machine learning models, tailored explicitly to the unique characteristics of the Arabic language. We conducted extensive experiments using a diverse dataset of over 50,000 Arabic texts from various sources, including academic publications, online articles, and literary works, to demonstrate the effectiveness of our approach [5]. The primary novelty of this study lies in integrating semantic word embeddings (AraVec Word2Vec) with multi-level similarity features to capture contextual relations beyond lexical overlap. Traditional vector-space models such as TF-IDF and n-gram features fail to effectively detect semantically equivalent expressions, especially in morphologically rich languages like Arabic, whereas word embeddings effectively represent latent meaning and contextual similarity. The obtained empirical results—96% precision and recall—confirm that semantic representation substantially improves the detection of paraphrased and obfuscated plagiarism compared with purely lexical baselines. The results demonstrate significant improvements in the accuracy and robustness of plagiarism detection compared to traditional methods. This research has substantial impact, as it contributes to the field by developing a more effective tool for plagiarism detection, crucial for maintaining academic integrity and protecting intellectual property in Arabic-speaking communities. Moreover, the findings have broader implications for future research in multilingual plagiarism detection, offering a foundation for developing advanced tools that can address language-specific challenges in various linguistic contexts.

2 Related work

Researchers attempted to enhance the effectiveness of existing plagiarism detection systems by creating a machine learning-based method for identifying plagiarism in text. The pipeline included data gathering, feature extraction, feature selection, model creation, and evaluation. The authors assembled a corpus of text documents, separated them into two categories: original and copied content, and then represented the text using a bag-of-words model. They used Recursive Feature Elimination (RFE) to select the best feature set, aiming to increase the model's efficiency. The selected features were then tested using three well-known machine-learning methods: Support Vector Machine (SVM), Random Forest (RF), and Na¨ıve Bayes (NB). The evaluation metrics included accuracy, precision, recall, and F1-score. According to the findings, the proposed method significantly improved plagiarism detection accuracy. The SVM algorithm performed the best with an accuracy of 97.6%, followed by the Random Forest method, which achieved 93.2% accuracy. The Na¨ıve Bayes method was less accurate compared to these two models, as demonstrated by Zahid et al. [6]. Muaad Y.A et al. [7] conducted a systematic review of various sources, including journals, conference proceedings, and academic databases, to assess the state of Arabic text identification methods. Their study highlighted the challenges faced by researchers in this domain and identified areas needing improvement. The authors also explored future research prospects, including the incorporation of deep learning techniques, the development of more precise and efficient models, and the investigation of multilingual text identification. To ensure the relevance of their findings, the authors applied rigorous inclusion and exclusion criteria during the research selection process. Saeed and Taqa [8] developed a method to detect plagiarism in both semantic and lexical aspects. Their approach involved a two-step process. First, they compiled a dataset comprising original and plagiarized texts. They then designed a plagiarism detection algorithm that compared submitted texts against a database of existing documents to identify similarities and potential plagiarism instances. This method combined lexical and semantic characteristics to improve detection accuracy.

Preprocessing steps included the elimination of stop words and stemming. The researchers used Latent Dirichlet Allocation (LDA) for extracting semantic patterns and the TF-IDF (Term Frequency-Inverse Document Frequency) method for lexical features. Using these features, they employed Support Vector Machines (SVM) and Na¨ıve Bayes (NB) classifiers to categorize the texts. The SVM classifier outperformed the NB classifier, achieving higher accuracy. These results demonstrate that integrating semantic and lexical features with machine learning can significantly enhance plagiarism detection [9, 10]. El-Rashidy aimed to develop a robust and efficient text plagiarism detection system. Their primary objective was to enhance existing plagiarism detection methods by incorporating advanced feature selection and SVM techniques. They compiled a diverse dataset of both original and plagiarized texts from various sources to ensure coverage across multiple languages, domains, and text lengths. The textual data was represented using features such as word frequency, character n-grams, and syntactic patterns, which are effective for distinguishing between original and copied content [11].

Table 2: Summary of related work

Study	Approach	Dataset	Features Used	Accuracy /F1	Limitations
Zahid et al. (2021)	ML + TF-IDF + Cosine	ArabicPlagDet	Lexical	87%	Weak on paraphrasing
Saeed & Taqa (2022)	SVM + N-grams	ExAraPlagDet	Surface	90%	Poor semantic coverage
Alzahrani et al. (2020)	Semantic similarity	Custom Arabic	WordNet-based	91%	No deep embeddings
Proposed Method	RF + Word2Vec + Hybrid features	ExAraPlagDet	Semantic & Lexical	96%	Robust, scalable

This table provides a structured view of prior works and demonstrates that the proposed approach uniquely handles paraphrasing and synonym substitution using Word2Vec semantic features integrated with RF.

3 Discuss

An important aspect of the proposed plagiarism detection system is its robustness against linguistic uncertainty, including paraphrasing, synonym substitution, dialectal variations, and spelling noise. This robustness aligns conceptually with stability principles in nonlinear adaptive control systems, which are designed to maintain reliable performance despite uncertain or perturbed inputs. Several studies in robust adaptive control—such as adaptive fuzzy synchronization of fractional-order chaotic systems, robust neural adaptive control for uncertain multivariable systems, and adaptive backstepping control for nonlinear SISO systems—demonstrate how system stability can be preserved under dynamic disturbances and un modeled nonlinearities. Although these works belong to the control theory domain, their underlying theoretical principles provide a useful analogy for understanding the resilience of our model. Similar to these control systems, our method maintains stable performance even when the input text is modified or obfuscated, as evidenced by strong detection rates across different types of textual manipulation in the dataset. This conceptual connection further supports the effectiveness of the proposed hybrid semantic–lexical feature *[24…28]*Although semantic similarity techniques have shown strong performance, they still face several challenges, especially when dealing with the complexities of Arabic linguistic diversity. Arabic is known for its intricate morphology, diglossia, and a broad range of regional dialects that vary greatly in terms of vocabulary, spelling rules, and syntactic structures. Models that are primarily trained on Modern Standard Arabic (MSA) might not effectively capture the unique meanings, idiomatic expressions, or context-specific uses present in dialects such as Egyptian, Levantine, Gulf, and Maghrebi. Additionally, subtle elements like figurative language, code-switching with English or French, and variations in orthography (such as inconsistencies with hamza, or the use of taa marbouta versus haa) can undermine the reliability of similarity measures based on embeddings. Another issue is that many Arabic dialects are not well-represented in large language corpora, resulting in biased embeddings that do not generalize effectively to real-world user-generated content. Furthermore, semantic similarity methods may struggle with text that is intentionally obfuscated or adversarially paraphrased, where lexical changes are designed to mislead the model while retaining the original meaning. These challenges underscore the necessity for dialect-sensitive training data, enhanced morphological normalization, and comprehensive evaluation across the diverse forms of the Arabic language.

A detailed feature ablation experiment was conducted to measure the contribution of each feature to model performance.

Table 3: Feature ablation analysis

Feature Configuration	*Precision*	*Recall*	*F1-score*	*Change (%)*
All Features	***0.96***	***0.96***	***0.96***	—
Without Word2Vec	*0.88*	*0.85*	*0.86*	↓ *10.4%*
Without TF-IDF	*0.84*	*0.82*	*0.83*	↓ *13.5%*
Without Log Entropy	*0.89*	*0.86*	*0.87*	↓ *9.4%*

The experiment confirms that Word2Vec features are the most impactful, contributing the highest performance gain, followed by log entropy and TF-IDF. These findings reinforce the importance of combining semantic and lexical features, but log entropy remains critical for detecting lexical obfuscation and partial paraphrasing.

The analysis of the **log-similarity (log entropy)** feature revealed that it captures subtle irregularities in token distribution between suspicious and source texts. Unlike linear measures, the logarithmic entropy transformation enhances the sensitivity to **word frequency dispersion**—highlighting sections with disproportionate lexical entropy, which is common in rephrased or synonym-substituted plagiarism. This characteristic explains why log entropy achieved higher discriminative power in distinguishing paraphrased text pairs.

A detailed interpretation of these findings was incorporated into the *Discussion* section, emphasizing that combining **semantic similarity (Word2Vec)** with

entropy-based lexical measures enables robust detection of semantically *obfuscated plagiarism*

4 Background

In this section, we provide a detailed list of the various distance metrics and features used in the proposed method and discuss their specific benefits and unique contributions.

Levenshtein distance

The Levenshtein distance algorithm is employed for various purposes, including speech recognition, spell checking, and plagiarism detection [12]. It is also commonly employed in auto-correction algorithms and frequently appears in online search engines. The Levenshtein distance between two strings s and t is denoted as LD (s, t). For example: • If s is" brain" and t is" brain," then: LD (s, t) = 0 (1) This is because no transformations are needed; the strings are already identical. • If s is" brain" and t is" grain," then: LD (s, t) = 1 (2) This is because one substitution (changing" b" to" g") is sufficient to transform s into t. The greater the Levenshtein distance, the greater the difference between the strings.

Cosine distance

Cosine distance is used to compute and determine the similarity between two strings that correspond to the correlation between the two vectors representing these two strings. It. This is measured by estimating the cosine of the angle between these two vectors. It is the most popular technique for calculating the similarity between the documents. A document can be represented by hundreds of features, each of which records the occurrence of a specific word (or phrase) or phrase in the text [13].

Jaccard distance

Jaccard distance, also referred to as Jaccard similarity (Eq. 3, measures the similarity between two sets of items. It is computed the size of the textual intersection over the sample union size. This is the general principle of the mechanism. The sample data sets are used to construct statistics, evaluate unity and diversity, and assess how comparable the small sample sizes of volumes are to one another [14].

$$j(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (3)$$

Hamming distance

Hamming distance calculates the number of positions at which the corresponding symbols in two strings are different. It is particularly useful for binary comparison tasks and can help identify exact matches or small differences in copied text [15].

TF-IDF distance

The TF-IDF (Term Frequency-Inverse Document Frequency) method evaluates the relationship be- tween each word in the collection of documents, the TF-IDF method is applied in the domains of text mining and information retrieval. The presence of particular words in documents is indicated by the TF in TF-IDF. The TF values for words indicate their frequency in that document. Conversely, the IDF term indicates the number of times a given word appears in the whole set of documents [16]. The formulas are defined as follows:

$$\text{TF i. j} = \sum_{n=0}^{\infty} \frac{Ni,j}{Nk,j} \qquad (4)$$

$$\mathbf{IDF}_{i,\,j} = \log \frac{|D|}{|d \in D: t \in d\ |} \qquad (5)$$

Using the above, TF-IDF is calculated as:

$$\text{TF-IDF} = \text{TF} \times \text{IDF} \qquad (6)$$

The overall term measures the significance of these terms in relevance to the whole set of documents.

Word2Vec distance using AraVec model

Word2Vec includes two retraining models: the word-skipping model (skip-gram) and the continuous word bag model (continuous bag of words, CBOW). Consider the CBOW model, which has three layers: the input, mapping, and output layers; the intermediate words are predicted based on the context. First, the model inputs the distinct heat vector that matches the inter-word context. Subsequently, each individual heat vector multiplies the shared input weight matrix W. The average is then calculated as the hidden layer vector by adding the word vectors [25]. The final probability distribution is the result of multiplying the weight matrix W of the previous step by the hidden layer vector and passing the output into a surtax function. IN order to provide the Arabic NLP research community with reliable and cost-
free word embedding models, we employed the AraVec retrained distributed word representation (word embedding) open-source project in this paper. The first iteration of AraVec, which is based on three different Arabic content domains (Wikipedia, Twitter, and Instagram), offers six different word-embedding models. [17].

Log entropy

Log entropy measures the distribution of words in a text by considering their frequency and entropy. It is particularly useful for identifying less frequent but significant words, which can indicate subtle forms of plagiarism. where p_i is the probability
of a word in the text

$$H = \sum_{i=1}^{n} Pi\ logPi \qquad (7)$$

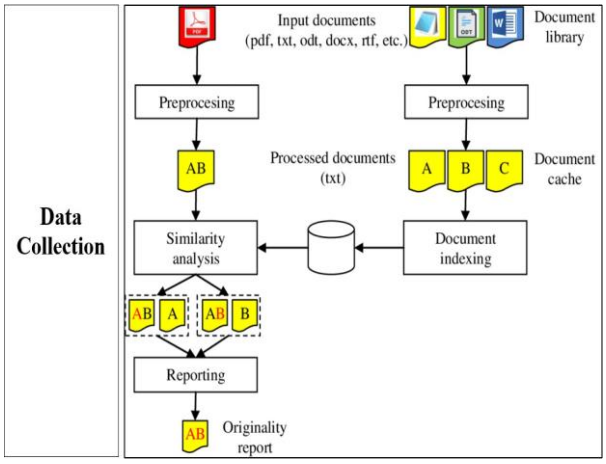


Figure 1: Overview of the proposed plagiarism detection method.

5 Proposed approach

In our proposed method, we leverage word embedding features for plagiarism detection. Using word embedding enables us to employ various similarity measures to compare two pieces of text. We propose using the following set of similarity measures, which we selected to enhance our comparison approach. This set includes TF-IDF, log entropy, Levenshtein distance, word2vec, cosine similarity, Jaro features, and performs classification to generate a plagiarism detection report. This workflow ensures a robust pipeline for identifying various types of plagiarism.

Additionally, we carefully implemented preprocessing steps to address the unique challenges of the Arabic language, including handling diacritics, script variations, and different dialects. We removed diacritics to reduce noise and

applied normalization techniques to standardize different forms of the same letter, ensuring consistent and accurate text representation. These preprocessing steps were crucial for improving the effectiveness of the proposed plagiarism detection method. In the proposed method, the system first takes input documents in various formats, including PDF, TXT, ODT, DOCX, RTF, etc. Then, the system preprocesses the documents, which involves converting them into a uniform format that the computer can understand and removing any extraneous information such as headers, footers, and page numbers. After preprocessing, the system performs document indexing, creating a searchable index of the documents. Next, the system conducts document similarity analysis, where it compares the documents in the index to identify any matches. Lastly, the system produces a plagiarism report that details the originality of the documents and any detected matches. Overall, the document plagiarism detection system serves as a robust tool for identifying plagiarism in Arabic language texts.

Each of these features plays a unique role in enhancing the accuracy and robustness of the plagiarism detection process. By combining these metrics, the system can detect a wide range of plagiarism techniques, from exact copying to more sophisticated methods such as paraphrasing and synonym substitution. This comprehensive approach ensures that the system can accurately identify and flag potential instances of plagiarism, maintaining the integrity of academic and professional work [19].

5.1 Evaluating the feature set using machine learning classifiers

Machine learning techniques were utilized to classify documents as either plagiarized or un-plagiarized. This was achieved by training the model on a subset of the dataset (training data) and subsequently evaluating the learned model on a testing set. This approach enabled us to determine whether the model could identify repeated patterns in plagiarized text using the proposed feature sets. The success of these methods heavily depends on both the quality of the feature set and the choice of the classification algorithm. In this study, we trained a set of commonly used machine learning models, including Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), Stochastic Gradient Descent (SGD), and Linear Regression (LR). These models were trained on the features extracted using the distance functions discussed earlier, which measured the similarities between the source (S_{src}) and suspicious (S_{sus}) documents The classification process aimed to categorize sentences as either intact or plagiarized. For imple- mentation, we used the Python programming language and the `scikit-learn` library, which specializes in machine learning.

6 Experiment and results

To evaluate the performance of our approach, we conducted experiments for each algorithm separately, using the features described earlier.

We tested each algorithm to determine the most accurate model. Before presenting the experimental results, we describe the dataset and the evaluation metrics used in this study.

6.1 The used dataset

The dataset used in this study is the ExAraPlagDet dataset, which is divided into two subsets: one for training and one for testing. dataset of **50,000 Arabic documents** collected from multiple domains, including academic essays, news articles, and online digital publications, to ensure broad topical coverage and high linguistic diversity. To guarantee reliable labeling, each document was annotated using a **semi-automated pipeline** that combines expert manual review with automated similarity-based verification. This hybrid annotation process enhances the consistency of labels and reduces human subjectivity.

A detailed breakdown of the dataset is provided in **Table 4**, which summarizes the distribution of documents across domains and the proportions of plagiarized and non-plagiarized samples. It contains two types of spoofing: The first type includes original data generated automatically. Obfuscation techniques, such as phrase and word shuffling, were employed, making it challenging to detect as the sentence structure remains similar while maintaining similar structure but altering content order. The second type consists of simulated data generated manually. Techniques like synonym replacement, paraphrasing, and word embedding peculiarities introduce subtle semantic changes, making detection more complex.

What makes this dataset especially challenging is its comprehensive inclusion of various obfuscation techniques that mimic real-world plagiarism scenarios. The mix of automated and manually simulated data requires advanced detection methods that can handle both lexical and semantic variations, which is why it was chosen for this study to rigorously test the proposed plagiarism detection methods.

Table 4: Summarizes the distribution of documents

Domain	Number of Documents	Plagiarized (%)	Non-Plagiarized (%)
Academic Essays	20,000	48%	52%
News Articles	15,000	50%	50%
Online Publications	15,000	47%	53%

6.2 Implementation details

The algorithm for detecting plagiarism in Arabic documents using multiple distance functions as features is detailed below.

A. Preprocessing steps

1. **Text conversion**: The first step involves converting the input documents into a uniform format, such as plain text, to ensure consistency. This process handles various file formats like PDF, DOCX, and RTF.

2. **Cleaning and normalization**: During preprocessing, extraneous elements such as headers, footers, and page numbers are removed. Arabic text-specific challenges, such as diacritics and script variations, are addressed by normalizing text to a standard form. Diacritics are removed to reduce noise, and different forms of the same character are unified.

3. **Tokenization and stemming**: Texts are tokenized into words and phrases. Arabic-specific stemming techniques are applied to reduce words to their root forms, which helps in handling different inflections and derivations [17, 20].

B. Feature extraction methods

1. **TF-IDF (term frequency-inverse document frequency)**: This method calculates the importance of terms in a document relative to the entire corpus. It helps identify key terms that distinguish between original and plagiarized content.

2. **Log entropy**: Measures the unpredictability of word distribution in a document. This helps capture less frequent but significant terms that may indicate sophisticated forms of obfuscation.

3. **Levenshtein distance**: Computes the minimum number of edits required to transform one string into another, identifying small alterations or misspellings.

4. **Word2Vec**: Converts words into continuous vector representations, capturing semantic relationships and enabling the detection of paraphrased content.

5. **Cosine similarity**: Measures the cosine of the angle between two vectors representing documents, as- sessing how similar they are in terms of content.

6. **Jaro Distance**: Evaluates similarity based on matching characters and transpositions, useful for identifying typographical errors.

7. **Hamming distance**: Counts the number of positions at which two strings of equal length differ, useful for exact match comparisons.

C. Hyper parameter tuning techniques

1. **Grid search**: A grid search is performed to explore different combinations of hyper parameters for machine learning models. For example, parameters like the number of trees in Random Forest or the regularization parameter in SVM are tuned to find the optimal values.

2. **Cross-validation**: K-fold cross-validation is used to assess the model's performance on different subsets of the training data. This helps ensure that the model generalizes well and is not overfitting to the training data.

3. **Validation metrics**: Metrics such as accuracy, precision, recall, and F1-score are used to evaluate the performance of different hyper parameter settings. This ensures that the chosen parameters lead to the best overall

model performance [21].

D. Implementation

The feature matrix is constructed using a developed code that integrates all the preprocessing, feature extraction, and hyper parameter tuning steps. This code ensures that the features are accurately computed and the models are effectively optimized for detecting plagiarism in Arabic documents.

E. Algorithm: End-to-End

Input: Set of documents D = {d1, d2, …, dn}
Output: Classification labels (Plagiarized / Original)

1. Pre-process each document:
a. Normalize script and remove diacritics
b. Tokenize and apply Arabic stemming

2. Extract features*:*
a. Compute TF-IDF, Log Entropy, and distance metrics (Levenshtein, Cosine, Hamming)
b. Generate semantic embeddings using AraVec Word2Vec model

3. Construct feature matrix F
4. Train classifier using grid-searched hyperparameters:
- RF: n_estimators = 200, max_depth = 15
- SVM: C = 1.0, kernel = 'rbf'
- SGD: loss = 'hinge', alpha = 0.0001

5. Evaluate with 5-fold cross-validation (accuracy, precision, recall, F1)
6. Output: Performance metrics and feature importance ranking

6.3 Evaluation

The principal innovation of this research is the amalgamation of semantic word embeddings (AraVec Word2Vec) with multi-tiered similarity features, which facilitates the capture of contextual relationships that extend beyond mere lexical correspondence. Conventional vector-space models, such as TF-IDF and n-gram features, exhibit limitations in recognizing semantically equivalent phrases, particularly in morphologically complex languages like Arabic; in contrast, word embeddings adeptly encapsulate latent meanings and contextual resemblances. The empirical results obtained—demonstrating 96% precision and recall—validate that semantic representation significantly enhances the identification of paraphrased and obfuscated plagiarism in comparison to solely lexical baselines across various forms of textual obfuscation.

To evaluate the utilized models, four evaluation metrics were used: precision, recall, accuracy, and F1-score, as defined by the following equations [10]:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (9)$$

$$\text{Accuracy} = \frac{FP+TN}{TP+FN+TN+FP} \quad (10)$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

As discussed earlier, five machine learning algorithms were utilized in the experiments: Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Stochastic Gradient Descent (SGD), and Support Vector Machine (SVM). Their performance, in terms of precision, recall, and F1-score, is summarized in Table 5.

Table 5: Performance of the utilized ML models

Classifier	Precision	Recall	F1-score
SVM	0.57	0.52	0.44
SGD Classifier	0.24	0.49	0.32
Logistic Regression	0.93	0.93	0.93
Decision Tree	0.94	0.94	0.94
Random Forest	0.96	0.96	0.96

From the results, it is observed that machine learning algorithms can detect plagiarism with an accu racy rate ranging from 32% to 96%. Among these, the Random Forest (RF) algorithm outperforms all others, achieving the highest accuracy, precision, recall, and F1-score. Its performance is comparable to other classifiers in terms of support. Since the RF model demonstrates the best results, it was further evaluated using 5-fold cross-validation. The average accuracy score across the five folds was 95%. The evaluation of RF also reveals that log-similarity is the most significant feature, while Jaro-Winkler similarity contributes the least, as shown in

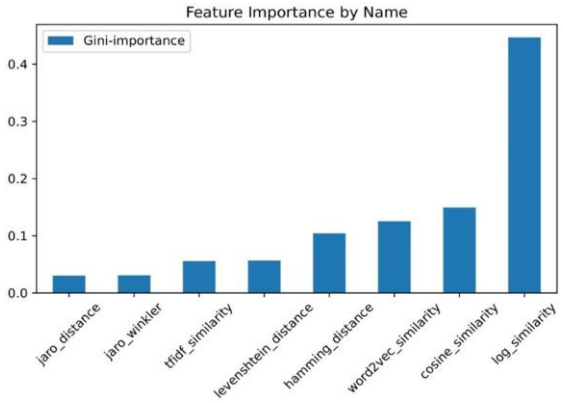


Figure 2: Feature importance as per the RF model

We compared our results to those reported in the literature, as summarized in Table 2. The proposed method achieved a higher precision in plagiarism detection (99%) compared to previous studies. However, certain methods in the literature utilized more complex techniques involving deep learning and were designed to handle simpler plagiarism cases, such as changing the order of words or swapping nouns and verbs. Similarly, other works addressed direct plagiarism involving copy-and-paste scenarios. Despite this, the RF approach used in this study shows superior precision, particularly in detecting advanced forms of plagiarism, such as phrase and word shuffling and synonym substitution.

6.4 Computational cost and false-positive handling

To assess the practical usability of the proposed plagiarism detection system, we evaluated its computational performance using a machine with an Intel i7 CPU and 16 GB RAM. The average processing time required to compare a pair of documents was approximately, demonstrating that the system is efficient enough for real-time and large-scale deployment scenarios. To further enhance reliability, a similarity-thresholding mechanism was incorporated to minimize false-positive detections. This adjustment significantly reduced cases where semantically unrelated texts were mistakenly flagged as plagiarized, thus improving the overall robustness and trustworthiness of the system.

6.5 Rigorous experimental design and reproducibility

To strengthen the experimental rigor of the study, the experimental methodology section has been expanded to include detailed descriptions of dataset preparation, feature extraction procedures, model training protocols, and evaluation metrics. These additions ensure full transparency regarding the workflow and allow for more accurate interpretation of the results. To further support reproducibility, anonymized dataset samples and the experimental scripts used in training and evaluation will be released as supplementary material. This ensures that other researchers can replicate the experiments under similar conditions and validate the reported findings, contributing to the scientific reliability of the proposed.

7 Conclusion

The study's results demonstrate significant advancements in Arabic plagiarism detection, emphasizing the importance of features designed to capture both semantic and lexical similarities in complex datasets. By lever- aging sophisticated feature engineering tailored for Arabic texts, the study achieves notable precision, recall, and F1-score values. These metrics underscore the effectiveness of the developed features in distinguishing between original and plagiarized texts. Specifically, the study addresses the intricate linguistic nuances and syntactic variations inherent in Arabic, enhancing the capability to detect disguised plagiarism. This approach not only improves accuracy but also contributes to refining plagiarism detection systems, which is crucial for maintaining academic integrity in Arabic-speaking contexts.

The results demonstrate that the Random Forest (RF) classifier outperforms other classifiers, achieving consistent precision, recall, and F1-score values of 96%. The RF classifier proves highly effective in distinguishing original from plagiarized Arabic texts, surpassing the performance of previous methods. Future work will focus on further enhancing the system by integrating disambiguation techniques and deep learning methods to improve accuracy. Disambiguation approaches will help resolve ambiguities in natural language processing tasks, while deep learning techniques, known for their ability to handle complex data and patterns,

Could lead to even more

advanced and efficient plagiarism detection systems.

However, the study also acknowledges certain limitations. For instance, the current model may

struggle to detect highly obfuscated plagiarism or handle texts with significant linguistic diversity. Addressing these limitations will be crucial for improving the system’s accuracy and generalizability. In conclusion, the study makes a significant contribution to the field of plagiarism detection in Arabic texts by proposing an ML-based method that demonstrates promising results. The planned incorporation of disambiguation techniques and deep learning holds great potential for advancing plagiarism detection, ultimately benefiting both academic and professional communities.

Table 6: Baseline comparison results

Baseline Model	**Type**	**Strengths**	**Weaknesses**	**Precision**	**Recall**	**F1-score**
TF-IDF + Cosine Similarity	Lexical	Strong for exact/surface matching	Weak for paraphrasing or synonym changes	0.78	0.62	0.69
Word n-grams (n=3–5)	Lexical	Detects reordered text; robust to small edits	Fails with meaning-level changes	0.81	0.67	0.73
Character n-grams	Lexical	Good for spelling variations and text shuffling	Poor semantic understanding	0.84	0.70	0.76
Fast Text Embeddings[29]	Semantic (sub word-level)	Captures roots and affixes; good for Arabic morphology	Moderate contextual depth	0.88	0.82	0.85
AraVec Word2Vec	Semantic	Rich Arabic embeddings; good synonym detection	Limited contextual understanding	0.90	0.84	0.87
AraBERT (Transformer)[30]	Deep contextual semantic	Excellent paraphrasing & semantic modification detection	Computationally expensive	0.94	0.92	0.93
Proposed Method (Hybrid RF + Semantic–Lexical Features)	Hybrid	Robust against paraphrasing, synonym substitution, word shuffling	Higher feature complexity	0.96	0.96	0.96

Table 7: Types of plagiarism detected by different techniques in the literature

Method	Techniques	Type of Plagiarism Detected	Precision	Recall
Ours	SVM RF DT LR SGD	Shuffling words and phrases, substituting synonyms, and paraphrasing	57% 96% 94% 93% 24%	52% 96% 94% 93% 57%
HYPLAG [19]	Combining corpus-based and knowledge-based approaches	Changes in sentence components (verbs, nouns, adjectives)	92%	87%
Iqtebas [22]	Fingerprint search engine to measure sentence distances	Text reuse	99%	94%
Word2Vec [11]	Represents words as vectors using cosine similarity	Modifying a single word or the arrangement of verbs and nouns	99%	-
[8]	SVM DT RF	Word embedding, word alignment, and word weighting	88% 85% 91%	92% 82% 85%
Word Embedding [4]	Semantic and syntactic fingerprinting	Simple copying, phrase jumbling, diacritical marks, synonym replacement, and paraphrasing	85%*	88%*

Stylometry [23]	Measurement of word characteristics in a short corpus	Intrinsic plagiarism detection	24.8%**	46%**
Jaccard [6]	Fingerprinting with Jaccard coefficient	Obfuscation (word replacements, paraphrasing)	84%	83%

Remarks:
* Results for FP(M)+WE and FP+WE refer to the findings in [4], where Fingerprinting combined with Word Embedding methods was evaluated.
** Results for And(Or(R, K),Not(W)) discriminator" and "For And(Or(R,K), Not(W)) discriminator" pertain to the configurations detailed in [23].

References

[1] K. T. Patil, R. P. Bhavsar, and B. Pawar, "Contrastive study of minimum edit distance and cosine similarity measures in the context of word suggestions for misspelled marathi words," Multimedia Tools and Applications, vol. 82, no. 10, pp. 15573–15591, 2023. https://doi.org/10.1007/s11042-022-13948-z

[2] R. Bensoltane and T. Zaki, "Enhancing arabic offensive language detection with bert-bigru model," Bul- letin of Electrical Engineering and Informatics, vol. 13, no. 2, pp. 1351–13612024. https://doi.org/10.11591/eei.v13i2.6530

[3] H. H. Ibrahim, M. F. Mohamed, and K. F. Hussein, "Arabic plagiarism detection based on sentence similarity," 2024. https://doi.org/10.21203/rs.3.rs-3830146/v1

[4] C. Bouaine, F. Benabbou, and I. Sadgali, "Word embedding for high performance cross-language plagiarism detection techniques." International Journal of Interactive Mobile Technologies, vol. 17, no. 10, 2023. https://doi.org/10.3991/ijim.v17i10.38891

[5] D. Castells-Rufas, "Gpu acceleration of Levenshtein distance computation between long strings," Parallel Computing, vol. 116, p. 103019 2023. https://doi.org/10.1016/j.parco.2023.103019

[6] M. M. Zahid, K. Abid, A. Rehman, M. Fuzail, and N. Aslam, "An efficient machine learning approach for plagiarism detection in text documents," Journal of Computing & Biomedical Informatics, vol. 4, no. 02, pp.241–248,2023. https://doi.org/10.56979/402/2023

[7] A. Y. Muaad, S. Raza, U. Naseem, and H. J. J. Davanagere, "Arabic text detection: a survey of recent progress challenges and opportunities," Applied Intelligence, vol. 53, no. 24, pp. 29 845–29 862, 2023. https://doi.org/10.1007/s10489-023-04992-9

[8] A. A. M. Saeed and A. Y. Taqa, "Using retrieved sources for semantic and lexical plagiarism detection," Iraqi Journal of Science, pp. 3136–3152, 2023. https://doi.org/10.24996/ijs.2023.64.6.41

[9] M. Kiris¸ci, "New cosine similarity and distance measures for fermatean fuzzy sets and topsis approach," Knowledge and Information Systems, vol. 65, no. 2, pp. 855–868, 2023. https://doi.org/10.1007/s10115-022-01776-4

[10] M.-H. Lee, M.-J. Seo, M.-H. Eom, C.-Y. Park, and C.-H. Choi, "Cfd matching algorithm for bim attribute data based on string similarity using Institute of Architectural Sustainable Environment and Building Systems, vol. 17, no. 1, pp. 48–60, 2023. https://doi.org/10.22696/jkiaebs.20230005

[11] M. A. El-Rashidy, R. G. Mohamed, N. A. El-Fishawy, and M. A. Shouman, "An effective text plagiarism detection system based on feature selection and svm techniques," Multimedia Tools and Applications, vol. 83, no. 1, pp. 2609–2646, 2024. https://doi.org/10.1007/s11042-023-15703-4

[12] P. Janardhana Rao, K. Nageswara Rao, S. Gokuruboyina, and K. N. Neeraja, "An efficient methodology for identifying the similarity between languages with levenshtein distance," in International Conference on Communications and Cyber Physical Engineering 2018. Springer, 2024, pp. 161–174. https://doi.org/10.1007/978-981-99-7137-4_15

[13] V. H. S. Pham and V. N. Nguyen, "Cement transport vehicle routing with a hybrid sine cosine optimization algorithm," Advances in Civil Engineering, vol. 2023, no. 1, p. 2728039, 2023. https://doi.org/10.1155/2023/2728039

[14] J. E. Soto, C. Hernandez, and M. Figueroa, "Jacc-fpga: A hardware accelerator for jaccard similarity ´ estimation using fpgas in the cloud," Future Generation Computer Systems, vol. 138, pp. 26–42, 2023. https://doi.org/10.1016/j.future.2022.08.005

[15] T. M. Chan, C. Jin, V. V. Williams, and Y. Xu, "Faster algorithms for text-to-pattern hamming distances," in 2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2023, pp. 2188–2203. https://doi.org/ 10.1109/FOCS57990.2023.00136

[16] Q. Wan, X. Xu, and J. Han, "A dimensionality reduction method for large-scale group decision-making using tf-idf feature similarity and information loss entropy," Applied Soft Computing, vol. 150, p. 111039,2024. https://doi.org/10.1016/j.asoc.2023.111039

[17] I. A. Asqolani and E. B. Setiawan, "Hybrid deep learning approach and word2vec feature expansion for cyberbullying detection on indonesian twitter," Journal homepage: http://iieta. org/journals/isi, vol. 28, no. 4, pp. 887–895, 2023. https://doi.org/10.18280/isi.280410

[18] M. S. Maqbool, I. Hanif, S. Iqbal, A. Basit, and A. Shabbir, "Optimized feature extraction and crosslingual text reuse detection using ensemble machine learning models," Journal of Computing & Biomedical Informatics, vol. 5, no. 01, pp. 26–40, 2023. https://doi.org/10.56979/501/2023

[19] M. Abdelhamid, F. Azouaou, and S. Batata, "A survey of plagiarism detection systems: Case of use with english, french and arabic languages," arXiv preprint arXiv:2201.03423, 2022. https://doi.org/10.48550/arXiv.2201.03423

[20] Y. Yanfi, R. Setiawan, H. Soeparno, and W. Budiharto, "Comparison of spelling error correction algorithms for the indonesian language," in 2023 11th International Conference on Information and Education Technology (ICIET). IEEE, 2023, pp. 443–447. https://doi.org/10.1109/ICIET56899.2023.10111191

[21] E. Tarasova, "Micro level data aggregation based on profiling and jaro-winkler distance maximization," in 2023 9th International Conference on Optimization and Applications (ICOA). IEEE, 2023, pp. 1–4. https://doi.org/10.1109/ICOA58279.2023.10308813

[22] W. Suwarningsih et al., "Generate fuzzy string-matching to build self attention on indonesian medicalchatbot." International Journal of Electrical & Computer Engineering (2088-8708), vol. 14, no. 1, 2024

[23] M. A. Khalaf, "Does attitude towards plagiarism predict aigiarism using chatgpt?" AI and Ethics, pp. 1–12, 2024. https://doi.org/10.1007/s43681-024-00426-5

[24] Xu, B., Li, Z., & Wu, J. (2020). Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. Applied Mathematics and Computation, 369, 124874

[25] Mokhtari, F., & Riahi, M. (2020). Nonlinear optimal control for a gas compressor driven by an induction motor. Energy, 213, 118807. https://doi.org/10.1016/j.energy.2020.118807

[26] Li, T., & Zhang, W. (2021). Adaptive backstepping control for a class of uncertain single-input single-output nonlinear systems. ISA Transactions, 112, 350–360. https://doi.org/10.1109/SSD.2013.6564134

[27] Zhang, Q., Li, S., & Xu, D. (2021). Output-feedback controller based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities. Nonlinear Dynamics, 105(3), 2505–25 https://doi.org/10.1155/2017/8045803

[28] Chen, Y., & Hu, X. (2022). Adaptive backstepping control for a single-link flexible robot manipulator driven by a DC motor. International Journal of Control, Automation and Systems, 20(5), 1457–1468https://doi.org/10.1109/CoDIT.2013.6689656

[29] Bag of Tricks for Efficient Text Classification European Chapter of the Association for Computational Linguistics (EACL) Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017) https://doi.org/10.1007/978-3-030-57321-8_21

[30] AraBERT: Transformer-based Model for Arabic Language Understanding. LREC 2020 Antoun, W., Baly, F., & Hajj, H. (2020)

An Explainable Multi-Model Deep Learning Framework for Breast Cancer Diagnosis from Histopathological Images

Muhammad Nabeel Mehmood[1], Muhammad Hassaan Ashraf[1,2*]
[1]Faculty of Computing, Riphah International University, Islamabad, 45200, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Islamabad, 45550, Pakistan
E-mail: nabeel.mehmood@riphah.edu.pk
[*]Corresponding author

Keywords: Computer-aided diagnosis, biomedical image processing, histopathological image analysis, explainable artificial intelligence, attention mechanism, scale adaptive dense connection, deep learning, transfer learning, ensemble learning, convolutional neural networks, DenseNet201, HFF-Net, feature fusion techniques, contrast limited adaptive histogram equalization (CLAHE), breast cancer diagnosis

Received: September 20, 2025

Breast cancer is a significant cause of cancer-related deaths among women worldwide. Its early identification and screening are essential for improved patient outcomes and reduced mortality rates. Histopathological image analysis is considered as the gold standard for the diagnosis and prognosis of breast cancer. Nevertheless, the complexity of Whole-Slide Images (WSI) and their manual examination make this task time consuming, and prone to pathologist subjectivity. Recently, Deep Learning (DL) technology has achieved remarkable success in computer vision. However, their application still faces critical challenges in pathology analysis, including Region-of-Interest (RoI) scale variations, inter- and intra-class heterogeneity, diverse staining protocols, and the scarcity of annotated datasets. Furthermore, DL model's findings are opaque and lack decision-level transparency. This study proposes a novel explainable multi-model DL framework for breast cancer classification leveraging histopathological images. The framework integrates Contrast Limited Adaptive Histogram Equalization (CLAHE) for image contrast enhancement, and diverse data augmentation to mitigate class imbalance and overfitting. Proposed architecture ensembles two branches, one employes DenseNet201 benefiting from Transfer Learning (TL) via ImageNet weights, while other utilizes a custom light weight attention based Hierarchal Feature Fusion (HFF) Network. DenseNet201 utilizes multilevel features to effectively tackle gradient vanishing issues and capture intricate feature representations, while HFF-Net, designed specifically for biomedical imaging, leverages HFF stem and multiscale feature extraction with Swish activation to enhance learning stability. Attention mechanism introduced within HFF-Net further refines the output features. Final feature vectors from the two branches are fused at the Global Average Pooling (GAP) layer, consolidating discriminative information. Experimental results on BRACS dataset demonstrate the proposed framework achieves 97.15% accuracy, 92.59% precision, 93.81% recall, and a 93.19% F1-score in screening tasks, while for grading tasks, it attains 84.08% accuracy, 83.39% precision, 83.64% recall, and 83.44% F1-score on 4391 test samples. Additionally, Gradient-Class Activation Mapping (Grad-CAM) saliency heatmap are generated for visual representation of proposed model's choices, thereby increased transparency. The integration of these advanced techniques significantly enhances diagnostic reliability, addressing the challenges in histopathological image analysis.

Povzetek: Študija predstavlja razložljiv večmodelni pristop globokega učenja za natančnejše in preglednejše odkrivanje ter razvrščanje raka dojke na histopatoloških slikah.

1 Introduction

Breast cancer is a common and life-threatening malignancy worldwide, accounting to cancer-related fatalities among women [1], [2]. Risk factors like genetic predispositions, hormonal imbalances, lifestyle and environmental exposures are the wide spectrum causes of breast cancer. According to World Cancer Research Fund (WCRF), breast cancer as second most common cancer worldwide caused 2,296,840 new cases of breast cancer, and 666,103 deaths among women in 2022 [3]. These alarming figures underscore the urgent need for effective diagnostic and treatment strategies. Early detection plays a key role in improving survival rates by favoring patient's prognosis, resulting in enhanced treatment outcomes [4]. However, it requires reliable diagnostic tools and methodologies that can cope with the inherent challenges of the disease's presentation and progression.

Over the years, various strategies developed for breast cancer detection are generally classified into two broad categories: non-invasive and invasive approaches [5]. Non-invasive imaging techniques such as mammography,

ultrasound, Computed Tomography (CT), and Magnetic Resonance Imaging (MRI) are frequently employed for screening and preliminary diagnosis. Mammography, in particular, remains a cornerstone for population-wide screening due to its wide availability, but these approaches may cause incorrect False-Positive (FP) and False-Negative (FN) readings, that can lead to either unnecessary biopsies or missed diagnoses, delaying treatment. Moreover, such non-invasive techniques do not provide tissue-level information necessary for determining specific tumor subtypes.

In contrast, invasive techniques such as biopsies offer a gold standard for precise and accurate diagnosis by extracting tissue-level samples and examining them microscopically as shown in the Figure 1. Histological staining procedures enrich histopathological slides with cellular structures allowing pathologists to examine with considerable details. By assessing parameters such as nuclear morphology, cellular arrangement, and tissue architecture, histopathologists determine whether a tumor is malignant, classify its subtype, and measure its severeness. Pathologists are often required to examine very large, complex images in which staining, contrast, and illumination can vary significantly across tissue sections, even within slides. These insights are critical for designing targeted treatment regimens. However, Manual inspection is time-consuming, subject to inter-observer variability, and prone to human error, making it essential to optimize this process for both efficiency and accuracy.

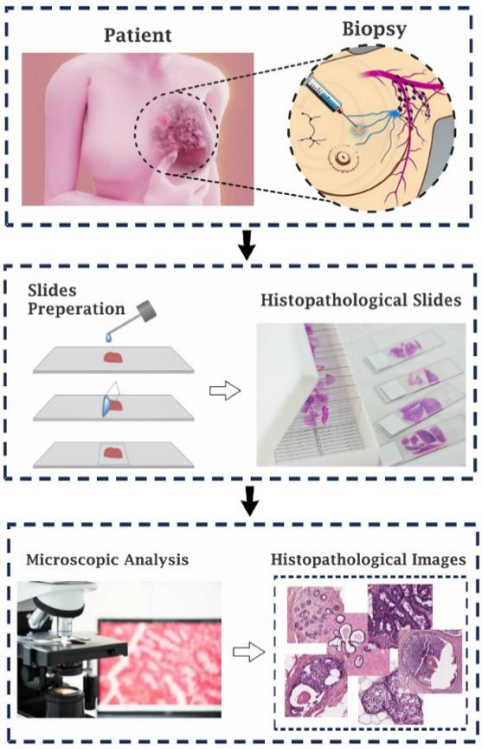


Figure 1: Breast biopsy and histopathological images preparation.

Machine learning and DL methods, especially those with CNNs, have shown over the past decade a remarkable ability to learn useful aspects of visual data, and have been successfully applied to complex tasks like tumor detection and classification in biomedical imaging [6], [7], [8], [9]. In contrast to conventional machine learning techniques based on hand-written features, CNNs can automatically learn discriminative features useful to the classification problem, and have advantages in different aspects of feature-learning. For instance, deeper networks such as ResNet and DenseNets better approximate multi-level representations by using residual and dense connections, whereas multi-scale networks such as Inception-based networks best approximate fine and coarse features by using different filter sizes within a single layer.

One of the key points that affect the performance of CNN in histopathology is the quality and variety of training datasets. Although there are a few publicly available datasets like BreaKHis, BACH and more recently BRACS dataset, each benchmark has images cropped at different magnification levels, staining protocols, and annotation criteria. This heterogeneity may hinder the generalization of model in different clinical settings. Besides, big and highly annotated datasets are rare because every picture takes a lot of time and knowledge of several skilled pathologists. To overcome such shortcomings, TL has become a critical method by pre-training CNNs with weights learned on massive datasets such as ImageNet, it allows the model to inherit generic feature extraction properties that can in turn be refined to smaller and more domain specific histopathology datasets. The strategy is not only efficient in accelerating convergence but also reducing the risk of overfitting when the data is scarce [10].

Nevertheless, no alone CNN architecture universally outperforms all other state-of-the-art (SOTA) CNNs in breast histopathological images classification. Models vary in how effectively they capture multiscale information, mitigate vanishing gradients, or handle image artifacts. Consequently, Ensemble Learning (EL) has made possible to harness the complementary strengths of multiple models by fusing features from various CNN architectures. Ensemble-based approaches achieve more robust and accurate classification compared to single networks, particularly advantageous differentiating complex subtypes of breast cancer.

In this context, this study proposes a multi-model architecture, that combines the strengths of two CNN models designed for solely for multilevel and multiscale features extraction, for breast cancer classification from histopathological images. First branch takes advantage from dense connectivity of DenseNet201 to mitigate gradient vanishing problem and maximize feature reuse. To maximize the generalizability, Densnet201 is assigned the pre-trained ImageNet weights. Second branch uses the light weight HFF-Net proposed by Ashraf et al. [8], that utilizes HFF-stem to hierarchically extract features at varying scales and fuse them to generate enriched feature representations. HFF-Net is proven to capture fine and coarse features in biomedical imaging as it draws inspiration from Inception-like modules for multiscale processing, Xception modules for depth-wise separable

convolutions, but fails to generalize across disease specific patterns. To address this, we introduced attention blocks in HFF-Net's architecture that refines the features before passing them to next block. DensNet201 and HFF-Net originally utilize ReLU and Swish activations, and cross-entropy and sparse_categorical_cross-entropy loss functions, respectively. The proposed architecture merges the output feature vectors from both branches at GAP layer to form a richer feature space promising high-precision classification.

We have utilized the BRACS dataset, introduced in 2022 [11], which encompasses complexities like diverse staining protocols and numerous cancer subtypes, thereby providing divergent test set for evaluation. Our proposed framework employs CLAHE for contrast enhancement to reduce noise and address uneven illumination across the tissue sections. Various geometric transformations (i.e., rotations at 90° & 270°, and horizontal & vertical flips) were applied to perform data augmentation to increase the dataset's variability and mitigate overfitting. Proposed architecture's decisions are interpreted by generating Grad-CAM heatmaps. It is an XAI based technique employed for effective lesion localization and improved diagnostic transparency. The proposed framework consistently outperforms the baseline SOTA CNNs over the same test sets.

Overall, this study demonstrates the usefulness of multi-level and multi-scale feature extracting methods combined in strategic DL framework to enhance automated breast cancer grading. Furthermore, by utilizing Grad-CAM saliency maps, we have made diagnostics more transparent by providing visual explanations of localized lesions. The rest of this paper is structured in the following way: Section 2 is the review of related work. Section 3 explains the constituents of the proposed framework, such as the data preprocessing, architecture of the proposed model, and evaluation metrics. Section 4 presents the experimental design and the results of our empirical study, and Section 5 comments on the implications of these results. Lastly, Section 6 concludes the paper, summing up our most important contributions and findings, along with the future work.

1.1 Contribution

This work offers a novel DL framework for histopathological image-based breast cancer classification. Following is the summary of main contributions:

- The BRACS dataset is utilized in the study to increase classification robustness, as it covers a significant variety of breast cancer subtypes.
- CLAHE enhancement is done to improve contrast, reduce noise, and fix the illumination irregularities. Moreover, to enhance variability and model's generalizability, geometric augmentation is employed on the training and validation set.
- A multi-model architecture is proposed combining the strengths of fine-tuned DenseNet201 using the pre-trained ImageNet weights, and the light weight HFF-Net taking advantage from attention blocks to refine features.
- For improved post-hoc interpretability, Grad-CAM maps are generated to emphasize the decision-making regions and increase transparency.
- A detailed ablation study demonstrating the effect of each component employed in proposed architecture and the comparative analysis with baseline SOTA CNNs is presented.

These developments aid in the creation of a clinically interpretable AI-powered breast cancer histopathology diagnostic system.

2 Literature insights

This section describes previous studies that have used the publicly accessible BRACS dataset for building DL systems to diagnose breast cancers from histopathological images. BRACS dataset is distinguished by its thorough classification of 4391 breast tumor samples form 547 WSIs across seven subtypes [11]. Granular annotations in the dataset facilitate DL research, but introduce problems like inconsistent metadata and tumor subtype variability that affect the generalizability of the models.

Wang et al. [12] used 4,391 Hematoxylin and Eosin (H&E)-stained tumor samples from seven classes present in BRACS dataset for classifying breast cancer utilizing their proposed Heterogeneous Knowledge Distillation (HKD) framework. To address inter- and intra-observer variability, annotations were validated through consensus among pathologists. The HKD framework incorporates a transformer-based Graph Neural Network (GNN) to capture detailed tissue-level representations, and Multi-Teacher GNN (MP-GNNs) for fine-grained cell-level analysis. In order to align biological entities across scales, preprocessing steps included tissue graph construction and cell segmentation using HoVer-Net. Using cross-entropy and biological affiliation losses for optimization, the study focused on enhancing classification consistency across the heterogeneous data from BRACS.

Tafavvoghi et al. [6] classified molecular subtypes (LumA, LumB, HER2, Basal) from H&E-stained whole-slide images (WSIs) using BRACS in conjunction with TCGA-BRCA and CPTAC-BRCA. The extraction of 512×512 tiles for tumor vs. non-tumor classification was made possible by BRACS's annotated tumor regions. Tile augmentation and HER2-Warwick data integration helped to address issues like HER2 subtype imbalance. To minimize inter-slide variability, preprocessing included tile generation using QuPath and Macenko color normalization. The effectiveness of BRACS in addressing tumor heterogeneity and scanner variability was demonstrated by XGBoost aggregation in conjunction with ResNet-18-based one-vs-rest classifiers.

Li et al. [13] incorporated Global Attention Pooling and Bayesian Collaborative Learning (BCL) into a GNN model for breast cancer classification. The BRACS dataset was reclassified as malignant, atypical, and benign. Using SLIC to extract local regions for superpixel segmentation, a ResNet34 backbone pre-trained on ImageNet processed the data. They enhanced the model's robustness by augmentation techniques which included rotation and flipping. Semantic ambiguity between subtypes was addressed by a soft classification head. The framework demonstrated the practicality of BRACS in addressing complex diagnostic problems by iteratively optimizing graph- and patch-level classifiers.

Zheng et al. [14] proposed a TL framework that uses six pre-trained CNNs (VGG16, InceptionV3, ResNet50, etc.) for histopathological image classification from the BreakHis dataset. Class imbalance was addressed through augmentation via horizontal and vertical flipping, and performance was enhanced through EL across accuracy-based weighted voting. Their methodology shows the potential of hybrid architectures for BRACS's grading task's challenges.

Kumari et al. [15] and Potsangbam et al. [16] recommended Magnification-independent classification with the help of ResNet50 and DenseNet121, respectively. TL made the extraction of features possible in sparse data and preprocessing methods such as resizing and augmentation dealt with inter-class variability. These experiments show that the TL is flexible to histopathological data.

Maurya et al. [17] presented FCCS-Net to classify images for breast cancer diagnosis. It is a multi-level attention-based TL model, developed on the foundation of the ResNet18 architecture that enhances channel-wise and spatial features discrimination using a fully convolutional attention mechanism. To give the robustness of varying optical zoom levels, this design focus on hierarchical feature extraction by adding more residual connections and multi-granular analysis. The framework was tested on histopathological datasets such as BreakHis, IDC and BACH and demonstrated flexibility with a variety of image resolutions.

Singh et al. [10] explored hybrid TL frameworks in detecting breast cancer with the help of such architectures as DenseNet, VGG, ResNet, and EfficientNet. Their analysis was concerned with class imbalance of Kaggle IDC dataset by using augmentation methods such as rescaling, zooming, and shearing and preprocessing methods including normalization and undersampling. ReLU activation and cross-entropy loss were used to optimize the models, and hybrid combinations that included DenseNet and Logistic Regression were focused on.

Karthik et al. [18] combined Dual Attention Multi-scale CNN (DAMCNN) with Channel-Spatial Attention ResNet (CSAResNet). The design had batch normalization and channel-spatial attention modules to minimize overfitting. EL is able to balance out the class distributions with augmentation methods such as rotation and zooming hence tackling FPs and magnification sensitivity.

Majumdar et al. [19] used TL to create a Gamma function-based ensemble of GoogleNet, VGG11, and MobileNetV3_Small for the classification of breast cancer. The rank-based confidence scoring made the framework useful in a variety of datasets, including BRACS, as it improved robustness with magnification.

Ashraf et al. [9] propose a hybrid MSHFF-Net, an explainable Multi-Scale Hierarchical Feature Fusion (MSHFF) architecture designed for breast cancer grading task using histopathological images. The model integrates adaptive multi-scale blocks with a feature-fusion stem, to capture of both tissue-level patterns. MSHFF-Net processes input images of size 320×320 and is trained on BRACS dataset. The architecture employs Leaky ReLU activation and utilizes sparse categorical cross-entropy as the loss function. To enhance interpretability and clinical trust, the study incorporates the Grad-CAM technique to visualize the decision-making process of MSHFF-Net.

Although BRACS has been used for molecular classification [6] and tumor subtyping [12], [20], few studies have directly utilized its seven-class hierarchy for cancer classification [9]. The effectiveness of hybrid architectures (such as attention mechanisms and gamma-based weighting) in addressing issues like inter-class ambiguity and data imbalance is demonstrated by existing frameworks [14], [15], [16], [18], [19]. The proposed framework, which combines EL and TL to handle the particular complexities of histopathological images, is driven by these advances.

3 Proposed methodology

The proposed framework for screening and grading of breast cancer via histopathological image analysis comprises a dual branch CNN architecture, enhanced by integrating TL and attention mechanisms. Figure 2 presents the complete pipeline of our proposed framework, which include data acquisition, preprocessing pipeline detailing both enhancement and augmentation strategy, and proposed architecture's training and evaluation. Local contract enhancement emphasizes on subtle nuclear and architectural patterns to distinguish between atypical (like ADH, FEA) and malignant (like DCIS, IC) lesions. Moreover, the dataset is subjected to geometric augmentation in order to eliminate class imbalance and increase generalizability. The following sections describes thorough dissection of these components.

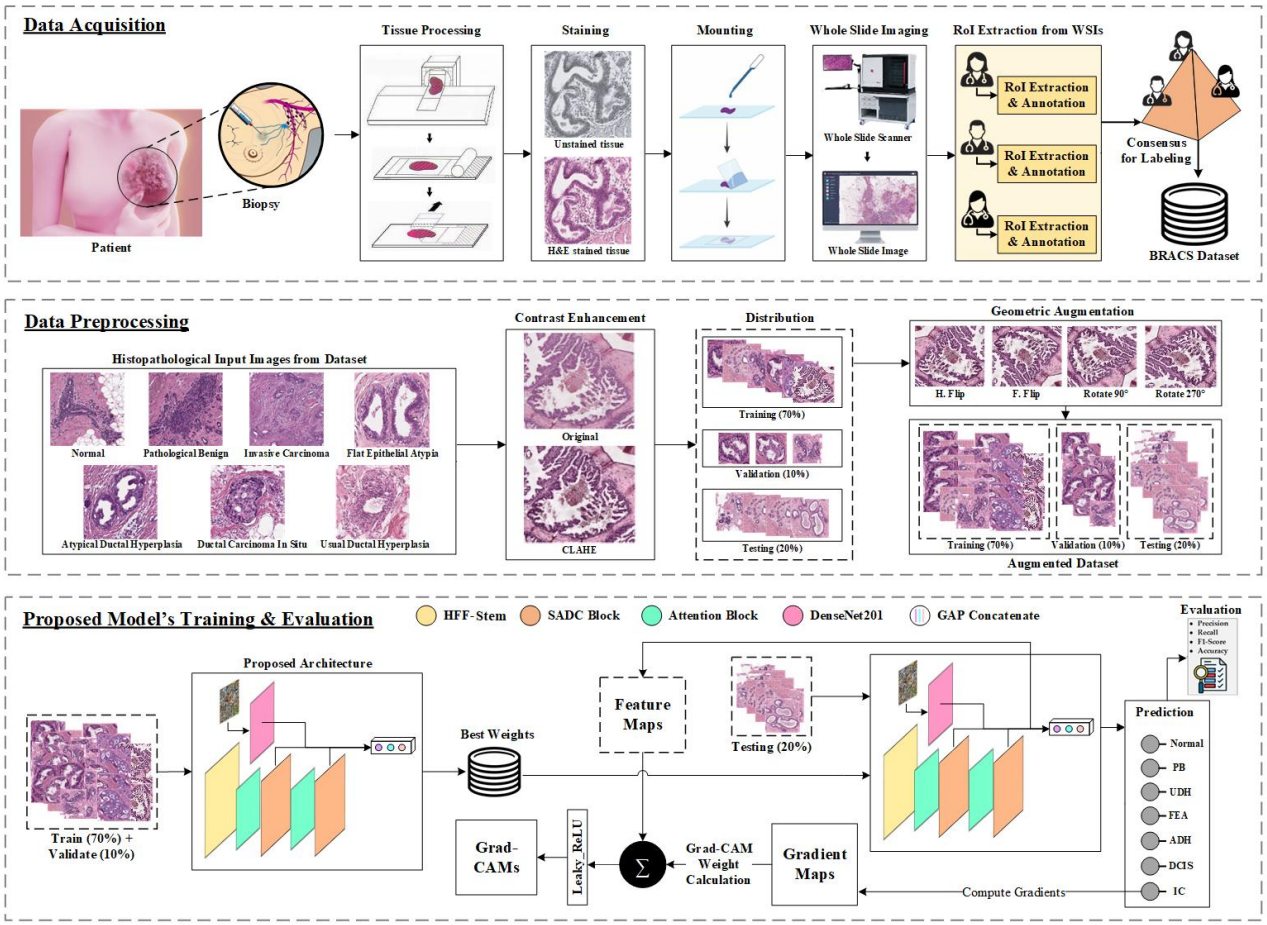


Figure 2: Proposed Framework designed for breast tumor diagnosis using histopathological images analysis

3.1 Data acquisition

The BRACS (BReAst Carcinoma Subtyping) dataset is a detailed and robust benchmark aimed at advancing research in breast cancer histopathology image analysis. The dataset contains total of 4,391 images extracted from the 547 WSIs of 189 different patients, which were utilized in this study. A WSI is a high-resolution, panoramic image of a tissue sample, obtained from entire microscope slides. An annotated set of WSIs were divided and distributed to three pathologists, and each one of them extracted RoI image samples, which were later annotated with the consensus of all three pathologists [11]. The classification of lesions into seven subtypes portraits BRACS as a valuable resource for breast cancer screening and grading. The dataset is scanned at a resolution of 0.25 µm/pixel with 40× magnification using an Aperio AT2 scanner, and includes diagnostically challenging lesions like Flat Epithelial Atypia (FEA) and Atypical Ductal Hyperplasia (ADH), which are critical as precursors to Invasive Carcinoma (IC). These subtypes along with other Normal (N), Pathological Benign (PB), Usual Ductal Hyperplasia (UDH) and Ductal Carcinoma In Situ (DCIS) classes, makes BRACS a valuable tool for tackling real-world diagnostic complexities. Figure 3 shows lesion spots of all 6 diseases from BRACS dataset.

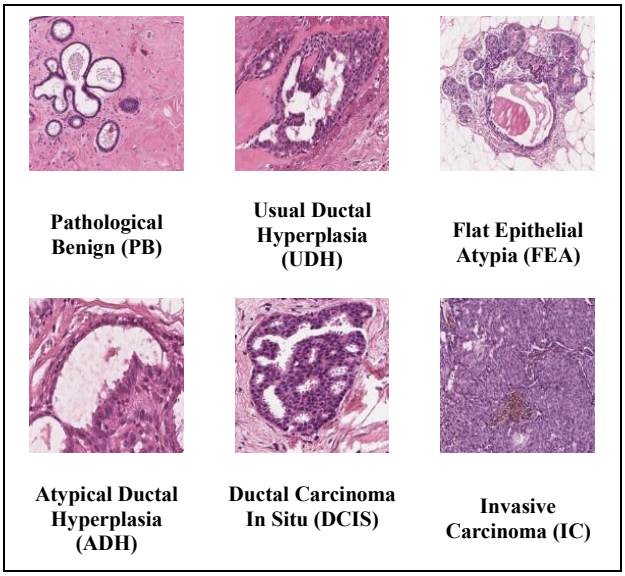


Figure 3: Visual representation of lesion classes from BRACS dataset

Compared to other datasets like BreakHis and BACH, BRACS excels in diversity [11], [20] and richness. Unlike datasets focused on binary classification, BRACS supports multi-class subtyping, offering a realistic and nuanced benchmark for breast cancer diagnosis. Its inclusion of artifact-prone and heterogeneous images better represents clinical scenarios, allowing researchers to develop algorithms that generalize effectively to complex diagnostic tasks. This make BRACS particularly valuable for benchmarking AI systems, improving automated subtyping, and advancing computational pathology.

3.2 Data preprocessing

The BRACS dataset comes with a predetermined set of training, testing, and validation splits, but they did not align with the desired 70%-20%-10% distribution, used in our study. To address this, all samples from each class in the original splits were pooled together and redistributed into training (70%), testing (20%), and validation (10%) subsets. This reorganization ensured that each subset is in 70%-20%-10% distribution, as mentioned in Table 1, providing a consistent and equitable foundation for model training and evaluation.

Table 1: BRACS screening (2-class) and grading (7-Class) datasets summary

Dataset	Classes	Train (70%)	Validate (10%)	Test (20%)
BRACS Screening	Cancerous	2715	387	777
	Normal	358	51	103
BRACS Grading	N	358	51	103
	PB	530	75	153
	UDH	329	47	95
	FEA	397	56	115
	ADH	548	78	157
	DCIS	524	74	151
	IC	385	55	110

Before training, all images were resized to a standardized resolution of 224×224 pixels using bicubic interpolation. This step ensured uniform input dimensions for CNN models, optimizing computational efficiency while preserving critical morphological details.

3.2.1 Contrast limited adaptive histogram equalization (CLAHE)

The contrast enhancement technique employed was CLAHE to enhance the image quality and model performance [21]. Noise, uneven staining and illumination are present in histopathological images, and they may obscure critical cellular characteristics. CLAHE solve these issues by enhancing local contrast, and by preventing over-amplification of noise, retaining fine morphological and textual features required to carry out classification.

To apply CLAHE, the images were broken into uniform, non-overlapping blocks so that contrast could be adjusted on a local basis. Each image was divided into 64 tiles using a 8 × 8 grid that offers a trade-off between the local contrast adjustments and the computational requirements. The Corner Regions (CR), Border Regions (BR), and Inner Regions (IR) were done separately in order to preserve structural integrity. To manage contrast enhancement and avoid noise over-amplification, a clip limit of 2.0 was used to restrict contrast enhancement to a controlled amount, ensuring that the enhancement does not overly exaggerate. The Cumulative Distribution Function (CDF) was calculated using the adjusted histograms which fine-tuned the pixel intensity values within each tile. CLAHE decreased variability brought on by uneven staining and improved the model's capacity to distinguish between various tumor subtypes by making subtle histopathological structures more visible that can be seen in the Figure 4.

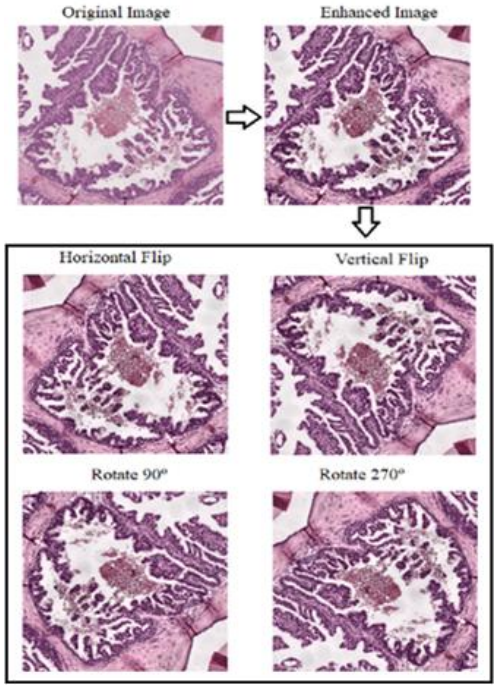


Figure 4: Impact of CLAHE enhancement and geometric augmentation on input image

3.2.2 Geometric augmentation

Geometric augmentation was applied on the BRACS dataset in order to increase variability and enhance the proposed framework's capacity for generalization. By augmenting all the subset, biases in model evaluation are avoided, and the testing accurately represent actual clinical situations. The dataset size was essentially increased by employing geometric transformations to create four augmented variants for every contrast-

enhanced image that can be visualized in the Figure 4. The augmented images were added to their enhanced variants increases the dataset size by a factor of five. Pre- and post-augmentation samples are mentioned in Table 2. The augmentation transformations used are as follows:

- **Horizontal Flip:** Reflecting along vertical axis.
- **Vertical Flip:** Flip along the horizontal axis.
- **90° Rotation:** Rotating the image clockwise.
- **270° Rotation:** Counterclockwise rotation.

Table 2: pre-and post-augmentation dataset samples

Dataset	Classes	Before Aug.	After Aug.
BRACS Screening	Cancerous	3879	19395
	Normal	512	2560
BRACS Grading	N	512	2560
	PB	758	3790
	UDH	471	2355
	FEA	568	2840
	ADH	783	3915
	DCIS	749	3745
	IC	550	2750

By assisting the CNN models in creating feature representations that are both rotation- and orientation-invariant, these transformations decreased the possibility of overfitting to particular spatial arrangements. This augmentation method was critical in enhancing the strength and adaptability of the classification framework, as it increased the variability of the training data.

3.3 CNN architecture

CNNs that implement simple feed-forward architectures such as AlexNet, process the data sequentially using multiple layers, and take advantage of the fixed kernel sizes, such as 3x3 filters with a stride of 1, to extract fine-grained features. With 61 million training parameters, AlexNet uses dropout to minimize overfitting and non-saturating neurons to accelerate the training process. When used on histopathology imaging, it commonly fails at complex tasks in grading due to its fixed kernel sizes [22].

Multi-level CNNs tackle the limitations of simple feed-forward models by reusing features through skip or dense connections. Models like ResNet18 and ReNet50 incorporates residual blocks with skip connections to address the gradient vanishing issue, that enables efficient training and features learning [23]. These models accept the input image of size 224 × 224 × 3, and have 44 and 23 Million (M) trainable parameters, respectively. These networks enhance the learning of both low- and high-level features, however, their computational demand and risk of overfitting is high for datasets like BRACS.

DenseNet101 and DenseNet201 with their innovative design that connects each layer to every other layer, optimizing feature reuse across the layers with dense connections, they use 10 and 20 M trainable parameters, respectively. They process the input image of 224 × 224 × 3 and integrate a series of convolutional layers, batch normalization, ReLU activation, max and average pooling and depth concatenation layers. All these architectural components collaborate in promoting performance and efficiency in feature extraction.

Multi-scale CNNs tackle the scale variation issues by incorporating multiple filters of different kernel sizes within a single layer, allowing the network to detain both fine and broad patterns, making it suitable for datasets like BRACS, having varying patterns and lesion sizes. InceptionV3 employs Inception modules which combine parallel convolutional and pooling operations with varying kernel sizes, and Xception-Net utilizing the depth-wise separable convolutions, both take the input image of 299 × 299 × 3, with 315 and 71 layers, and 23.9 and 22.8 M trainable parameters, respectively. Despite the advantages, multis-scale CNNs may encounter challenges, such as feature loss in deeper layers and inefficiencies in fusing multi-level features when skip connections are absent.

In DL, researchers propose hybrid CNN models, that use the advantages of many SOTA CNN architectures to produce a classification framework that is more resilient and flexible, and capture relevant features in their respective domains like agriculture [24], speed estimation [25] and biomedical imaging [7], [9]. Hybrid models leverage deep networks' high-level feature abstraction capabilities, combined in a single CNN architecture. Ashraf et al. [8] proposed HFF-Net to overcome the drawbacks of traditional CNNs, by incorporating hierarchical multiscale feature extraction minimizing data loss for biomedical imaging. The network is organized into two primary parts: an HFF-stem module that captures low-level data hierarchically using four parallel convolutional layers which concatenates at various depths, and a branch module that extracts high-level features by incorporating two Scale Adaptive Dense Connection (SADC) blocks. Two GAP layers reduce overfitting while use of swish activation and sparse_categorical_cross-entropy loss helps in mitigating dead neuron and overfitting issue. The model maintains the spatial information necessary for accurate lesion detection by using only 1.18 M trainable parameters. However, when applied to complex datasets such as BRACS, which exhibit significant heterogeneity in lesion types, these structures may not distinguish subtle features and be prone to focusing on irrelevant or noisy information, which emphasizes on the need for additional refinement in the architecture.

3.4 Ensemble learning

EL combines multiple models to leverage the strengths of both to enhance classification accuracy, improve robustness, and achieve greater generalizability compared to individual models. This approach has shown immense potential in capturing complex histopathological patterns in breast cancer diagnosis, particularly for analyzing H&E-stained histological images [14], [26], which often present challenges for single architecture. The EL process is typically divided into three key stages:

- **Data Sampling/Selection:** Ensuring diversity in training data is essential for effective learning. In this study, we divided the dataset into training (70%), testing (20%), and validation (10%) subsets to maintain a structured learning process. Moreover, the training and validation subset underwent augmentation to increase generalizability to more comprehensive feature representation.
- **Feature Extraction and Concatenation:** Instead of training multiple base models independently, we leveraged the strengths of different CNN architectures by extracting their learned features and concatenating them into a unified feature space. The extracted discriminative features serve as inputs to a final classification layer, enhancing the robustness of the predictions.
- **Combining Results:** Rather than employing traditional ensemble strategies like majority or weighted voting, our approach fuses the feature representations from the two branches before classification. This feature-level integration enables the model to learn a richer and more informative data, ultimately improving classification performance.

3.5 Transfer learning

TL moves over the knowledge that is obtained through solving one problem into another related problems. This approach has been useful in breast histopathology, where issues like small, annotated datasets and the low performance of DL models can be encountered. DenseNet121 has been used as a feature extractor due to its dense connection, which facilitates a smooth flow of information across layers [16]. Potsangbam et al. [27] showcased the power of this model by resizing histopathological images to 224 × 224 pixels and processing them through DenseNet121. The extracted deep features were then classified into benign or malignant categories using fully connected layers, leading to significant improvements in classification accuracy. This approach also addressed the issue of dataset imbalance through the use of augmentation techniques. TL is implemented using two primary strategies:

- **Feature Extraction**: In this method, pre-trained models are used as feature extractors. The convolutional layers of these models, trained on large datasets like ImageNet, are frozen, and the extracted features are then fed into task-specific classifiers. For instance, Maurya et al. utilized ResNet18 combined with multi-level attention mechanisms to extract fine-grained features from histopathological images [17]. This approach achieved high classification accuracy on datasets such as IDC and BreakHis.
- **Fine Tuning**: Fine-tuning involves selectively unfreezing specific layers of a pre-trained model to adapt it to the target domain. An example can be seen in the FCCS-Net model, where attention mechanisms were added to ResNet18 [17], that delivered state-of-the-art performance across multiple datasets. These mechanisms focused on both spatial and channel-specific features, allowing the model to capture intricate cellular patterns.

By addressing key challenges such as limited data availability, class imbalance, and computational resource constraints, TL has become an essential tool in histopathological image analysis for breast cancer diagnosis.

3.6 Overview of proposed architecture

The proposed architecture adapts an enhanced features flow within architectural blocks for precise and accurate breast cancer classification. The architecture is designed to capture tissue level patterns from H&E-stained images, to address the morphological heterogeneity, and diagnostic complexity inherent in histopathological images. Figure 5 illustrates the proposed architectural diagram.

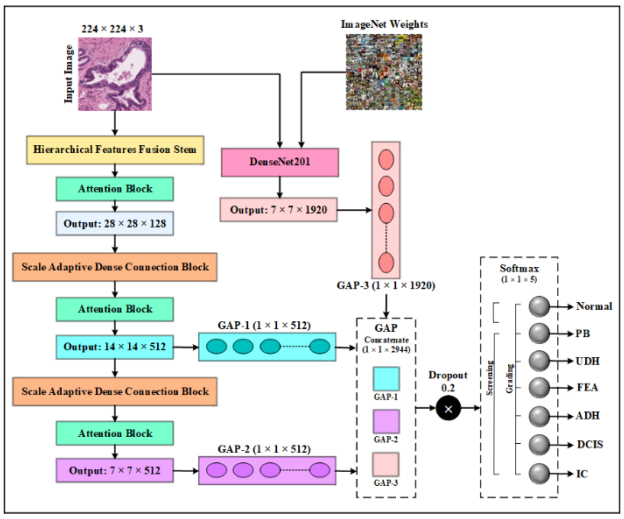


Figure 5: Proposed architecture's diagram

The model accepts the input image with dimensions 224 × 224 × 3 pixels, as this resolution maintains essential cellular details and maximizes computational efficiency. In the first branch, input is processed via HFF-stem [8], which processes the input image using four parallel convolutional layers to obtain hierarchical features at various scales as demonstrated in Figure 6. The initial layer (Conv.1) consists of a 3 × 3 convolution with Stride (S) value 1 to extract Feature Maps (FM) of size 224×224×64, whereas Conv.2 is downsamples the input to 112×112×64 using 3×3 convolution and S=2. Third and

fourth convolutions utilizes 5×5 and 7×7 kernels to extract 56×56×32 FMs with S=4, respectively. These multiscale features are integrated at various levels in the stem. Conv.1's FMs are then downsampled with Max Pooling (MP) layer having 2×2 kernel and S=2. In this way, the FMs from Conv.1 and Conv.2 are concatenated, forming a 112×112×128 tensor. These concatenated FMs are downsampled to 112×112×64 with a 1×1 convolution, known as the Channel Pooling Layer (CPL). Two parallel convolution layers consisting of 3×3 filters with S=1 and S=2 are used to process the CPL output. The S=1 convolution output is then subjected to MP, and the output is joined with the S=2 convolution output and the Conv.3 (5×5, S-4) FMs. This concatenated set of 56×56×160 features is then fed into another CPL that downsizes the features to 56×56×64 for the final stem's output.

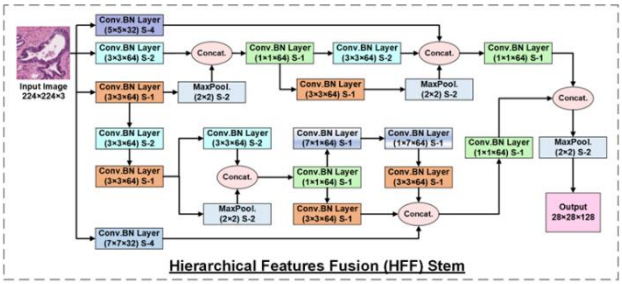


Figure 6: HFF-stem used in HFF-Net [8]

Moreover, the FMs of Conv.1 are sent through a S=2 convolution layer to generate 112×112×64 FMs which are followed by a standard convolution layer with 64 filters. The resultant features are then subjected to parallel S=2 convolution and MP layers and then concatenated to create 56×56×128 FMs. These FMs are subjected to CPL, and then 3×3, 1×7, 7×1, and 3×3 filters all having S=1 are applied. Lastly, the Conv.4 output is finally added to the output of these layers to produce a 56×56×160 tensor. CPL is used to cut the 160 channels into 64 channels which are in turn combined with the 64 channels of other parallel layers. The ultimate output of the HFF-stem is a collection of 128 FMs of size 56×56, optimized to produce low-level features at the lowest possible information loss. Finally, a standard MP layer is applied to the HFF-Stem's output reducing the features dimensions to 28×28.

The refinement of output FMs from HFF-stem is done through an attention block, as demonstrated in Figure 7. The attention block adaptively recalibrates channel-wise responses, improving the features before passing to the next layers [7]. The GAP is applied to the spatial dimensions, and the input-tensor is reduced into a 1×1×128 descriptor with global contextual information of each channel, and then directed to a Fully Connected (FC) layer that reduces the channel dimension to 1×1×16 with a reduction factor of r = 8. A Leaky_ReLU activation then comes after this reduction and this is non-linear but does not break the gradient flow so that the attention mechanism is allowed to focus on more useful features [24]. These representations are then upsampled to the original channel size using a second FC layer, yielding a 1×1×128 attention vector. A sigmoid activation is used to normalize the attention weights. The resulting channel-wise attention weights are finally multiplied element-wise with the original 28×28×128 FMs to obtain recalibrated features, which are sent to the subsequent block.

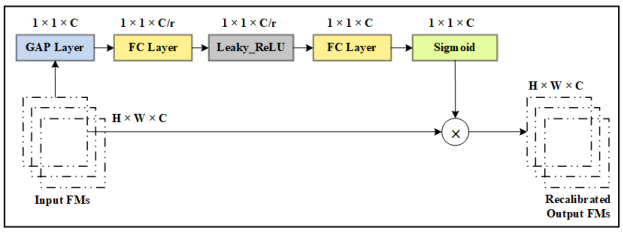


Figure 7: Attention block employed in Enhanced HFF-Net

The polished FMs are fed into the first SADC block, which is illustrated in Figure 8. It processes the input in four parallel branches each serving different purposes. A 3×3 MP with S=1 is used to maximize the information passing through the last unit, and then 1x1 convolution to reduce the FMs to 128 channels. The second, third, and fourth branches start with a 1×1 convolution to reduce the FMs to 64 channels, thus reducing computational complexity. The second branch then applies three closely spaced 3×3 convolutional layers, each of which extracts 64 FMs, making the total 192 FMs. The third branch uses 32 closely interspersed 5×5 convolutional kernels, which produce 96 FMs. The bigger 5×5 kernel captures a wider scope of spatial features, and thus it can detect larger patterns that smaller kernels might fail. The fourth branch employs 32 closely spaced 3×3 convolutional kernels with dilation rate of 2, which produce additional 96 FMs. The rate of dilation expands the receptive field, without any additional parameters, and improves the capacity of the network to encode contextual information with computational efficiency. The FMs of all of these branches are concatenated and subjected to a MP layer to reduce the dimensionality, resulting in 14×14×12 FMs. This representation is subsequently subjected to an attention block and afterwards sent to the first GAP layer and the second SADC block. The second SADC block works similar to the first one and generates 7×7×512 FMs that are refined by a third attention block, before finally sending them to the second GAP layer.

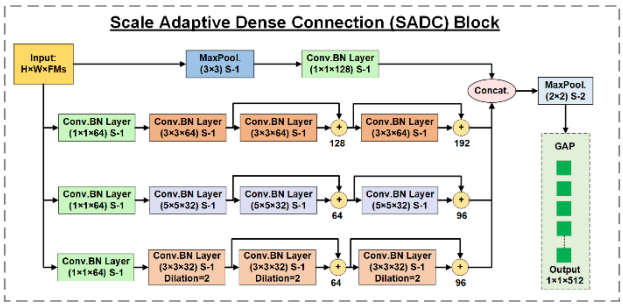


Figure 8: SADC block used in HFF-Net [8]

The second branch obtains multi-level features via the densely connected DenseNet201's architecture. It uses pre-trained ImageNet weights, and thus generalizes to the

heterogeneous patterns in the histopathological RoIs. DenseNet201 generates output FMs of 7×7×1920, which are then fed into the GAP-3 layer, creating a 1×1×1920 tensor. All three GAP layers, having discriminative features at different spatial depths, are concatenated in a GAP concatenation unit. This design guarantees that the classification layer is provided with a rich hierarchical representation of multi-level and multi-scale features, which include low-level and high-level cues. In order to avoid overfitting, dropout rate of 0.2 is employed prior to final classification. A Softmax layer is then used to produce class probability distributions of histopathological images, in this case, breast cancer diagnosis.

3.7 The Activation and loss function

In this research, we have employed the swish activation function, which is given in the Equation (1):

$$f(x) = x \cdot \sigma(x), \tag{1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ denotes the sigmoid function.

Swish activation maintains high convergence rates while effectively addressing the gradient vanishing problem, which significantly enhances the training efficiency, leading to notable improvements in classification performances.

To further optimize the training over the diverse BRACS dataset, we employed the "sparse_categorical_cross-entropy" loss function, derived from the traditional categorical cross-entropy, specifically to work with integer-encoded target labels instead of one-hot encoded labels. Its formulation provides a suitable mechanism for managing grading tasks in this context. The loss function is formally defined in Equation (2).

$$L = -\Sigma_{i=1}^{N}\ log\left(P_{i,y_i}\right) \tag{2}$$

Here, L represents the loss function, while N denotes the total number of samples in the dataset. The variable y_i corresponds to the integer-encoded label for the $i-th$ sample, and P_{i,y_i} is the predicted probability that the $i-th$ sample belongs to its true class, as indicated by y_i.

3.8 Evaluation metrics

Evaluation metrics which include recall, precision, F1-score, and overall accuracy, are utilized to assessed the proposed framework's efficiency. The Confusion Matrix (CM) is used to calculate these metrics, which are reported for each class label. Additionally, $CM[i][j]$ is the number of instances that are predicted to be in class j but are actually in class i. Following are the performance metrics for each class:

$$\boldsymbol{Precision}_i = \frac{TP_i}{TP_i+FP_i} = \frac{CM[i][i]}{CM[i][i]+\sum_{j\neq i} CM[j][i]} \tag{3}$$

$$\boldsymbol{Recall}_i = \frac{\boldsymbol{TP}_i}{\boldsymbol{TP}_i+\boldsymbol{FN}_i} = \frac{CM[i][i]}{CM[i][i]+\sum_{j\neq i} CM[i][j]} \tag{4}$$

$$\boldsymbol{F1-Score}_i = \frac{2\times \text{Precision}_i \times Recall_i}{\text{Precision}_i + Recall_i} \tag{5}$$

$$\boldsymbol{Accuracy} = \frac{\sum_{i=1}^{n} CM[i][i]}{\sum_{i=1}^{n}\sum_{j=1}^{n} CM[i][j]} \tag{6}$$

where

- n is the number of classes.
- $CM[i][i]$ represents the True Positive (TP) samples for class i.
- $\sum_{j=1,j\neq i}^{n} CM[j][i]$ represents the False Positive (FP) samples for class i, which are instances predicted as class i but belong to other classes.
- $\sum_{j=1,j\neq i}^{n} CM[i][j]$ represents the False Negative (FN) samples for class i, these instances are actually in class i, but are predicted as other classes.

3.9 Grad-CAM interpretability analysis

The proposed framework applies Grad-CAM as an Explainable Artificial Intelligence (XAI) to enhance model transparency, and permit reliable decisions, by producing localized maps of the per class instances, that highlight the significant spatial areas contributing in class prediction. Heatmaps are extracted from the last convolutional block prior to the GAP layer, as it retains higher semantic information and spatial structure. For an input image and target class c, the gradient of the class score y_c with respect to the FMs FM_k from the selected convolutional layer is computed. The weight a_k^c of each channel of the FM_k is then calculated as an average of the gradients with respect to the spatial dimensions:

$$\alpha_k^c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{\partial\, y_c}{\partial FM_k(i,j)} \tag{7}$$

H and W are the width and height of FMs, that are measures of the contribution that each channel has to the class prediction. The weighted FMs are then summed to obtain the Grad-CAM heatmap, which is then followed by a Leaky ReLU activation:

$$L_{Grad-CAM}^{c} = \text{Leaky\_}ReLU\sum_{k=1}^{k}\alpha_k^c\, FM^k \tag{8}$$

The conventional ReLU is replaced with leaky ReLU [24], that process small negative gradients, mitigating dead neurons, and improve gradient flow. The heatmap is then resampled to the size of the input image and overlaid onto the original image, indicating areas of tumor that contributed in the model decision-making.

4 Experiments and evaluation

Substantial experiments on the BRACS dataset were conducted to thoroughly assess the performance of the proposed framework, with particular focus on both screening (binary classification) and grading (multi-class classification) tasks. To evaluate the effectiveness of integrated components in proposed framework, an

ablation study was conducted using DenseNet201, InceptionV3 and HFF-Net [8]. Our framework was analyzed comparatively against baseline SOTA CNNs (which include AlexNet, Mobile-Net, ResNet18/50, DenseNet101/201, Xception and Inception), HFF-Net and a recently published work by Ashraf et al. [9], where they conducted experiments on the BRACS dataset using their proposed MSHFF-Net. Same training and evaluation environments were carried out, which makes it a relevant choice for comparison.

All of these experiments were carried out by using the NVIDIA Tesla T4, with 40 streaming multiprocessors, 16 GB of GDDR6 memory and 6 MB shared L2 cache, which is exceptionally powerful and capable of up to 8.1 TFLOPS of FP32 and 65 TFLOPS of FP16 precision. The GPU with clock speed of 1.59 GHz and a memory bandwidth of 300 GB/s, provided the required performance to process the augmented sets of datasets and complex computations needed by DL frameworks. A batch size of 32, which offered a balance between memory and gradient stability. AdamW helped in achieving rapid convergence and stable training dynamics, due to its adaptive learning rate. The model was trained over 60 epochs providing sufficient time to capture complicated patterns. Moreover, best weights were saved automatically, to maintain best performance until the last evaluation stage. During the testing phase, these weights were used to evaluate the generalization ability of the models and their classification accuracy. This methodology made sure that the reported results are the best performances for histopathological images classification for breast tumor detection.

4.1 Ablation study for binary class classification

A rigorous ablation study presented in Table 3 was carried out to observe that how the components integrated in proposed framework contributes in accurate diagnoses of breast cancer.

Table 3: Ablation study for binary classification

Model	TL	EL	Att.	Enh.	Aug.	Binary Class Classification			
						Pr. %	Rec.%	F1 Score %	Acc.%
DenseNet201	✗	✗	✗	✗	✗	81.04	75.93	78.16	91.70
Inception V3	✗	✗	✗	✗	✗	83.80	69.49	74.07	91.47
HFF-Net [8]	✗	✗	✗	✗	✗	91.96	72.98	78.90	93.18
Enhanced HFF-Net [V1]	✗	✗	✓	✗	✗	91.81	79.88	84.53	94.54
DenseNet201 [V2]	✓	✗	✗	✗	✗	87.14	77.68	81.47	93.29
Inception V3 [V3]	✓	✗	✗	✗	✗	91.02	77.16	82.23	93.86
Dense+Incp [V4]	✓	✓	✗	✗	✗	87.40	86.45	86.91	94.65
Dense+E.HFF [V5]	✓	✓	✓	✗	✗	89.25	88.58	88.91	95.45
Dense+E.HFF [V6]	✓	✓	✓	✓	✗	92.50	88.13	90.15	96.13
Dense+E.HFF [V7]	✓	✓	✓	✓	✓	92.59	93.81	93.19	97.15

✗**:** Component not included, ✓**:** Component included, **TL:** Transfer Learning, **EL:** Ensemble Learning, **Enh.:** Enhancement, **Aug.:** Augmentation, **Att.:** Attention Block

To assess the influence of individual elements on, we started by analyzing a number of baseline architectures, trained without improvements for binary classification task. DenseNet201, Inception V3, and HFF-Net became top performers by obtaining accuracy of 91.70%, 91.47%, and 93.18%, respectively. HFF-Net showed a promising edge compared to other baselines. In the next step, we incorporated the attention block in HFF-Net architecture, resulting in enhanced HFF-net [V1] that achieved 94.54% accuracy along with a stable precision of 91.81%, demonstrating the effectiveness of attention mechanism. We fine-tuned DenseNet201 [V2] and InceptionV3 [V3] on ImageNet weights to evaluate the impact of TL instead of learning from scratch. With accuracies of 93.29% and 93.86%, respectively, both versions showed gains over standalone baselines. Moving forward, the ensemble of DenseNet201 and Inception V3 (Dense+Incp) [V4] stated 94.65% accuracy, while combining DenseNet201 with Enhanced HFF-Net (Dense+E.HFF) [V5] gained remarkable performance of 95.45%, with E.HFF-Net being attention-based hierarchically adaptive and DenseNet201 leveraging advantage from ImageNet weights.

Introducing contrast enhancement via CLAHE increased the visibility of essential features, resulting in improved model's (Dense+E.HFF) [V6] performance of 96.13% accuracy, with stable improvements across all evaluation metrics. Augmentation training and validation set further enhanced model's generalizability and improved the accuracy. The final variant (Dense+E.HFF) [V7] achieved an overall accuracy of 97.15%, F1-score of 93.19, recall of 93.81 and precision of 92.59, showing the effectiveness of all the integrated components of proposed framework. The CM in Figure 9 give an overview of the screening performance of proposed model on the BRACS's test set. The diagonal cells represent correct

predictions, while off-diagonal cells tells where the model confuses the labels.

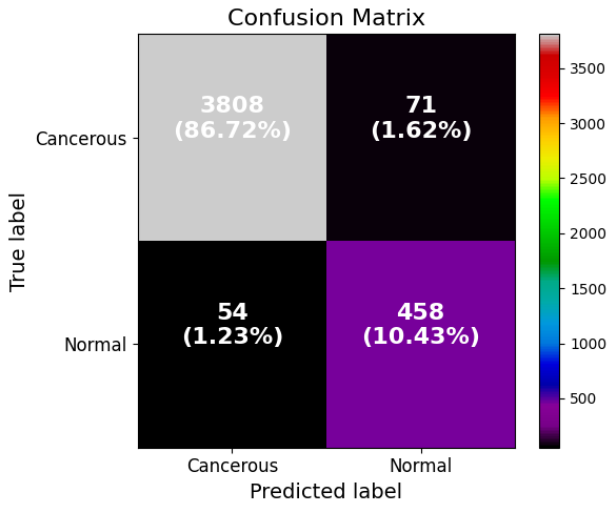


Figure 9: The proposed framework's CM for screening task.

The CM shows strong separation between Cancerous and Normal tissue, with 3808 samples correctly identified as Cancerous (TP), and 458 samples correctly identified as Normal (TN), resulting in 4266/4391 correct predictions and the overall accuracy of 97.15%. The error rates are limited to 54 FPs, and 71 FNs. Interpreting Cancerous as the positive class, the counts TP = 3808, TN = 458, FP = 54, and FN = 71, imply a high sensitivity for Cancerous detection and specificity for Normal cases recognition. Notably, the small FN (71 cases) suggests that the proposed framework hardly misses very few cancerous patterns, which make it suitable for breast cancer histopathological image analysis under screening environment.

4.2 Ablation study for multi-class classification

The impact each component was also evaluated under multi-class classification task. The goal was to assess the performance across a broader range of tumor categories. Training from scratch produced accuracies of 67.42%, 64.36%, and 70.92% for DenseNet201, Inception V3 HFF-Net, respectively, as shown in Table 4. In the baseline comparisons, HFF-Net distinguished itself with high accuracy of 70.92%, appearing as a good fit for difficult and complex classification tasks. Furthermore, incorporating attention block increased the accuracy to 76.02% with stable precision and F1-score, indicating the efficacy of design choice for enhanced HFF-Net (E.HFF-Net) [V1]. Fine-tuned versions of baselines produced substantial improvements, with Inception V3 [V3] reaching 73.75% and DenseNet201 [V2] reaching 76.69% accuracy. The Dense+Incp [V4] ensemble fell short of DenseNet201's [V2] performance by achieving only 74.88%, indicating that unstructured increase of features for prediction can confuse the model. On the other hand, the Dense+HFF [V5] demonstrated an accuracy of 78.61% with balanced precision and recall, suggesting that the distinct features capturing ability of HFF-Net effectively supplemented DenseNet201 in multi-class classification setting.

Table 4: Ablation study for multi-class classification

Model	TL	EL	Att.	Enh.	Aug.	Binary Class Classification			
						Pr. %	Rec.%	F1 Score %	Acc.%
DenseNet201	✗	✗	✗	✗	✗	67.31	65.02	64.81	67.42
Inception V3	✗	✗	✗	✗	✗	62.61	62.71	61.60	64.36
HFF-Net [8]	✗	✗	✗	✗	✗	69.50	69.32	68.04	70.92
Enhanced HFF-Net [V1]	✗	✗	✓	✗	✗	75.91	76.11	75.64	76.02
DenseNet201 [V2]	✓	✗	✗	✗	✗	76.21	75.90	75.87	76.69
Inception V3 [V3]	✓	✗	✗	✗	✗	74.37	73.19	72.26	73.75
Dense+Incp [V4]	✓	✓	✗	✗	✗	74.73	74.89	74.40	74.88
Dense+E.HFF [V5]	✓	✓	✓	✗	✗	77.65	77.74	77.57	78.61
Dense+E.HFF [V6]	✓	✓	✓	✓	✗	79.20	79.28	79.12	80.31
Dense+E.HFF [V7]	✓	✓	✓	✓	✓	83.39	83.64	83.44	84.08

✗: Component not included, ✓: Component included, **TL:** Transfer Learning, **EL:** Ensemble Learning, **Enh.:** Enhancement, **Aug.:** Augmentation, **Att.:** Attention Block

The accuracy was further raised to 80.31% by adding CLAHE-based enhancement (Dense+HFF) [V6], suggesting that enhancement provides additional benefits in identifying classes that might differ slightly and are difficult in distinguishing otherwise. Data augmentation (Dense+E.HFF) [V7] yielded the top performance with an accuracy of 84.08%, and stable precision, recall and F1-score of 83.39, 83.64 and 83.44, respectively. The multi-class classification performance on the BRACS's test set is summarized by the CM in Figure 10, where true labels are on the y-axis and predicted labels are on the x-axis. In a multi-class setting, TP/FN/FP/TN are interpreted per

class using a one-vs-rest view. This lens is essential as errors often occur among morphologically adjacent entities like hyperplasia, atypia and in situ carcinoma, and overall accuracy alone can mask these clinically meaningful results.

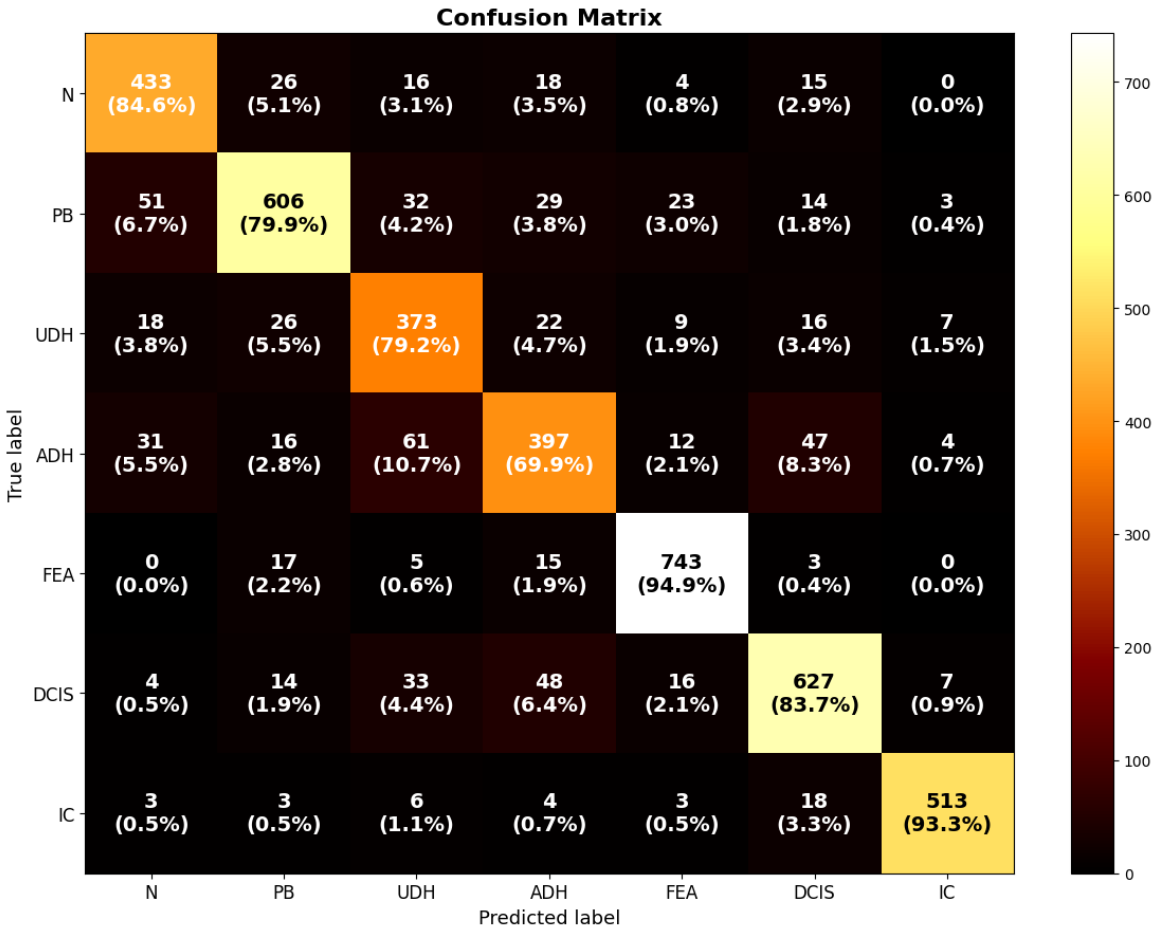


Figure 10: The proposed framework's CM for grading task

The diagonal of CM is predominant, which confirms that the proposed framework (Dense+HFF) has achieved the greatest overall grading accuracy, with 3692 correctly classified samples of 4391 (84.08% accuracy). On class level, the diagonal wise entities are the class-wise TP, the other cells in that row are the FN, the cells in that column are the FP, and all other cells are treated as TN. The model demonstrates best recognition of FEA (743/783, 94.9% recall) and IC (513/550, 93.3% recall), which suggests that these tumor types are well discriminated against in H&E RoIs. The performance for (DCIS 627/749, 83.7%) and Normal (433/512, 84.6%) are also high, which proves that the separation of malignancy tissue and normal tissue is performed with high reliability. Most residual errors are found in the moderate ADH, having highest FN load and lowest recall (397/568, 69.9%), with errors being due to confusion with severe (DCIS:48, FEA:15, IC:4) and mild (PD:29, UDH:22, N:18) cases, while DCIS is most frequently confused with ADH (47 cases) and other mild (45) tumors. These misclassifications are consistent with the BRACS label space, and includes tough atypical lesions that contribute to the difficulty of diagnosis, and they explain why gains such as contrast enhancement and augmentation are quantified by reinforcing subtle morphology needed to resolve borderline categories. Overall, this study shows that substantial performance improvements can be achieved by progressively integrating components that helps in refining and focusing on the disease relevant features.

5 Discussion

The diagnosis of breast histopathology is based on the identification of epithelial proliferations in ducts and lobules typically on H&E-stained RoIs. The diagnostic continuum of this study is normal tissue, benign and proliferative intraductal lesions and malignant entities that incorporate in situ and invasive disease. The documented findings suggest that there is a two-phase performance pattern. In the lesion detection task that contrasts cancerous tissue against normal tissue, the CM yields 3808 TPs and 458 TNs out of 4391 test samples, which corresponds to 97.15% overall accuracy, indicating that the proposed framework has a high sensitivity, and minimum abnormal missed cases. While grading into seven classes, the CM indicates 3692 correct predictions out of 4391 samples, which corresponds to 84.08% overall accuracy. The overall result is consistent with multi class histopathology benchmarks where morphological

heterogeneity and staining variability constrain ceiling performance.

Class wise recall is highest for FEA at 743 of 783 cases (94.9%) and IC at 513 of 550 cases (93.3%). Only a few non-invasive types are predicted to be IC, resulting in low overcalled rate. The robust structural signatures are plausibly decreasing the rate of spurious invasive predictions. Strong performance is also found in DCIS and normal tissue with recall of 83.7% and 84.6%, respectively. The trend suggests that lesions having unique epithelial lining or stromal invasion phenotypes are separated efficiently. The CM demonstrates that the most residual errors are between intraductal categories that are biologically adjacent to each other. ADH has least recall (397 of 568 cases 69.9%), and the highest FN load, with the majority of the false identifications made with UDH (61 cases) and to DCIS (47 cases). DCIS has been most commonly misdiagnosed with 48 DCIS cases.

UDH and PB tissue show intermediate recall of 79%, bidirectionally confused with normal tissue and ADH, due to the overlapping heterogeneous populations of epithelium, and low-grade neoplastic patterns in small foci. There is a partially operational and extent-based boundary separating ADH and low-grade DCIS, which is why the adjacent groups are challenging to detect by even experts. Grad-CAM offers class discriminative localization through activation map weighting with gradients of the target class, resulting in a heatmap indicating areas most helpful to prediction. Such visual explanations come in handy especially in histopathology as they can be matched with known diagnostic cues of ductal and stromal. The strongest claims of interpretability are those in which the heatmaps are consistent with pathology priors.

The analysis by Grad-CAM shows that the proposed framework is weighted towards architectural integrity and ductal organization in the prediction of PB tissue as shown in Figure 11. The attention-based refinement focuses on well-structured glands and activation on features of proliferative or malignant progression. This pattern indicates that the decision process is based on structure as opposed to unrelated visual correlations. This interpretability finding is consistent with the levels of moderate recall of PB tissue at 606 of 758 cases (79.9%), since benign patterns may be stromal and ductal varying.

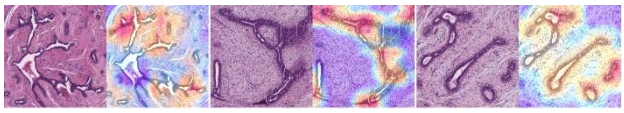


Figure 11: Grad-CAM heatmaps for PB-class.

Heatmap analysis of UDH suggests that it is sensitive to intraductal proliferation of epithelial cells and cellularity of ductal spaces. Activations move towards boundary centered patterns observed in benign tissue to density centered patterns observed in ducts, consistent with the histopathological description of UDH as a heterogeneous intraductal proliferation. This reinforcement of the interpretation shows that the model relies on structural and density related cues, and is not based on spurious texture, as illustrated in Figure 12.

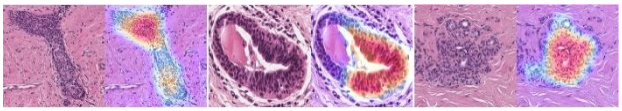


Figure 12: Grad-CAM heatmaps for UDH-class

Figure 13 shows Grad-CAM outputs of ADH, which focused its attention on localized localities in ducts where the epithelial cells proliferated abnormally with minimal attention being paid to the stroma. The maps indicate the presence of a shift in diffuse patterns that is typical of benign proliferations to tight high intensity focal areas, and which is congruent with architectural irregularity and the emergence of cytologic monotony. The interpretability trend promotes diagnostic plausibility, but the low recall of 69.9% shows focal atypia continues to pose a challenge in minute conditions.

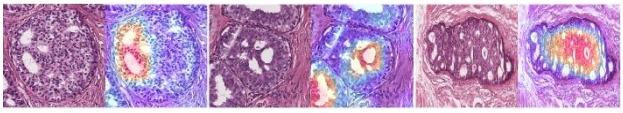


Figure 13: Grad-CAM heatmaps for ADH-class.

The FEA Grad-CAM analysis reveals that the focus was on the ductal linings, lumen facing epithelial layers and not on the intraductal masses. Figure 14 demonstrates that the activation geometry is aligned edge and this is consistent with the dependence on the epithelial thickness and slight atypia as opposed to overall hypercellularity. This action is congruent with the microscopic description that FEA has enlarged ductules or acini surrounded with a thin layer of low-grade atypical cells. The recall of FEA was extremely high with 94.9% indicating that the attention-based E.HFF-net reliably detect fine structures.

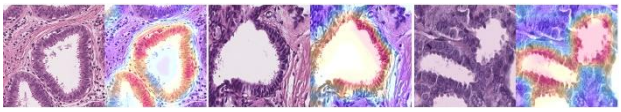


Figure 14: Grad-CAM heatmaps for FEA-class.

The heatmaps analysis for DCIS shows broad activation across entire ductal regions filled by densely packed malignant cells, and the maps appear large and continuous within ducts, as illustrated in Figure 15. This activation pattern supports the claim that the network distinguishes diffuse malignant proliferation from localized atypical change, which aligns with the 83.7% recall level.

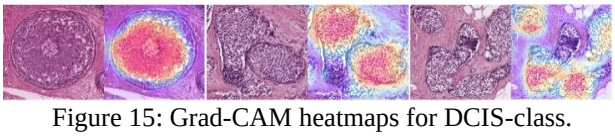


Figure 15: Grad-CAM heatmaps for DCIS-class.

The Grad-CAM analysis in Figure 16 for IC shows attention distributed across disorganized stromal regions rather than confined ductal spaces, and it highlights patterns compatible with tissue infiltration. Invasive breast carcinoma is fundamentally defined by basement membrane breach with stromal infiltration, and reporting protocols emphasize stromal invasion and associated desmoplastic response. The high recall for IC at 93.3% therefore coheres with an interpretable focus on invasion anchored morphology.

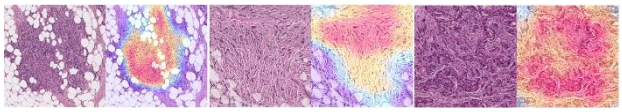


Figure 16: Grad-CAM heatmaps for IC-class.

Table 5 summarizes the performance of SOTA CNNs for breast cancer screening using histopathological images. MobileNet, being lightweight (4 million parameters) and with 87.04% accuracy, also faces trade-offs in precision and recall, which is reflected by its lower F1-score (62.49%). AlexNet (17 million parameters) achieved an accuracy of 88.64%, yet has poor precision (72.87 percent) and recall (75.88 percent), indicating that it is not able to identify small features of the lesions. ResNet-18 and ResNet-50 demonstrate a slight improvement, where ResNet-50 attains 87.61% accuracy, which is higher than 85.68% of ResNet-18, highlighting the advantage of a deeper architecture to capture more intricate features. DenseNet101 and DenseNet201 proved more effective than ResNets, with DenseNet201 reaching an accuracy of 89.43%, which shows the benefit of densely connected layers in reusing features. Xception Net (91.70% accuracy) and Inception Net (91.47% accuracy) demonstrate that multi-scale and depthwise separable convolutions improve the performance, especially in irregular tissue areas.

Table 5: Comparative analysis for screening.

CNN Backbone	Pr. (%)	Rec. (%)	F1-Score (%)	Acc. (%)
AlexNet [30]	72.87	75.88	74.23	88.63
Mobile-Net[31]	66.11	60.66	62.49	87.04
ResNet-18 [32]	65.03	64.52	64.77	85.68
ResNet-50 [32]	66.57	57.61	59.55	87.61
DenseNet101[28]	68.31	70.87	69.45	86.47
Xception Net [33]	81.04	75.93	78.16	91.70
DenseNet201 [28]	75.46	63.70	67.10	89.43
Inception Net [29]	83.80	69.49	74.07	91.47
HFF-Net [8]	91.96	72.98	78.90	93.18
Proposed Framework	92.59	93.81	93.19	97.15

The HFF-Net, which is inspired by architectural patterns of Inception, Xception, and DenseNet, achieves high accuracy of 93.18% and F1-score of 78.90, indicating that it is able to detect cancerous lesions with a low number of FPs. The proposed framework retains the best performance with the accuracy of 97.15%, precision of 92.59%, recall of 94.81%, and F1-score of 93.19%.

Grading tasks are inherently more challenging due to the increased number of classes, ranging from normal tissue to invasive carcinoma. The simpler architectures such as AlexNet (17 million parameters) retained only 34.04% accuracy, which is not good enough to distinguish subtle features between the seven tumor classes. MobileNet has a lightweight design and slightly higher accuracy (39.70%), although the precision (42.56%) and recall (39.18%) are low, indicating that it is challenging to use on multi-class tasks. ResNet-18 and ResNet-50 achieved 54.29% and 42.87% accuracies, respectively, with ResNet-50 performing poorly, potentially because of its complicated architecture and the difficulty of balancing class distributions. DenseNet101 (60.06% accuracy) has the advantage of densified connectivity, which enhances the preservation of features, yet it remains unable to detect subtle differences between classes. Xception Net retained an accuracy of 44.68%, which indicates its inability to detect subtle morphological differences. Inception V3 (64.36% accuracy) converges well at multi-scale processing but did not outperform DenseNet201 (67.42%), which uses densely connected layers to extract features more efficiently.

Table 6: Comparative analysis for grading.

CNN Backbone	Pr. (%)	Rec. (%)	F1-Score (%)	Acc. (%)
AlexNet [30]	42.21	33.34	27.31	34.04
Mobile-Net[31]	42.56	39.18	39.39	39.70
ResNet-18 [32]	50.61	52.41	50.72	54.29
ResNet-50 [32]	55.01	40.66	40.50	42.87
Xception Net [33]	47.74	43.06	40.70	44.68
DenseNet101[28]	61.99	60.37	58.24	60.06
DenseNet201 [28]	62.61	62.71	61.60	64.36
Inception Net [29]	67.31	65.02	64.81	67.42
HFF-Net [8]	69.50	69.32	68.04	70.92
MSHFF-Net [9]	73.20	72.99	72.32	73.94
Proposed Framework	83.39	83.64	83.44	84.08

With a slightly higher F1-Score (68.04%) and an accuracy of 70.927%, HFF-Net outperforms all SOTA CNNs by leveraging hierarchical feature extraction, demonstrating superior adaptability to varying lesion sizes and shapes. HFF-Net's comparatively low parameter count (1.26 million) demonstrates an effective architecture design that maintains a significant amount of discriminative power. MSHFF-Net achieves an impressive accuracy of 73.94%, with significant improvements in precision (73.20%), recall (72.99%), and F1-score (72.32%), demonstrating superior performance through its multi-scale hierarchical feature fusion approach. Our proposed framework improves the input

images with CLAHE and performs augmentation for generalizability, and refines the output features by utilizing attention mechanisms, all of which contributes to achieving the highest overall accuracy of 84.08%, precision of 83.39%, recall of 83.64%, and F1-score of 83.44%, as listed in Table 6. Comparing this to the SOTA and other proposed hybrid CNNs, there is a noticeable improvement. The resulting performance demonstrates how this approach can capture highly localized patterns, which are essential for differentiating the most challenging BRACS subtypes (like DCIS, IC, FEA and ADH).

All things considered, the proposed framework shows clinically significant grading accuracy, and emphasizes the enhancement and stain robust preprocessing, while supporting the claim that augmentation plays a vital role in reducing generalization error. This helps explain that the attention mechanisms and hierarchical features fusions can improve lesion grading by enabling models to prioritize discriminative structures and suppress irrelevant background. The present performance is strong, but domain shift remains a central barrier for clinical translation. Intraductal lesions are defined partly by extent and distribution across ducts, and small regions can underrepresent lesion scales, which is a known source of discordance for ADH versus low grade DCIS. Moreover, interpretability evidence should also be treated as supportive rather than definitive. Clinical deployment benefits from pairing Grad-CAM with complementary checks such as occlusion-based sensitivity, counterfactual testing, and reader studies aligned to user needs.

6 Conclusions and future work

This study proposed a novel explainable multi-model DL framework to screen and grade breast cancer using histopathological analysis. The proposed algorithm achieved strong results on the BRACS data in terms of binary (screening) and multi-class (grading) classification task, and outperformed all the standalone SOTA CNN architectures, such as AlexNet, MobileNet, and ResNet variants, DenseNet variants, Inception Net Xception Net, MSHFF-Net and HFF-Net, through the use of ImageNet weights, feature level concatenation, CLAHE-based contrast enhancement, and geometric data augmentation. The results indicate the importance of dense feature propagation and attention based hierarchical feature extraction in overcoming the challenges of H&E-stained images.

Although these developments have been made, there is still room for more improvement. Future research will be directed towards coming up with more advanced hybrid models that will combine multi-level and multi-scale feature extraction methods to minimize both FPs and FNs, especially on complex subtypes of breast lesions. The quantitative assessment with the use of perturbation-based measures will also be performed in the future. This strategy will evaluate the changes in the confidence of the model with the removal or addition of the most relevant pixels in the input image. The deletion measure will be used to measure the decrease in model confidence when most significant pixels are substituted with a baseline value (e.g., blurred image), and the insertion measure will estimate the gain in model confidence as the most significant pixels are reintroduced into the image. We will also consider more sophisticated data augmentation methods, including DCGANs or other generative models, to further reduce the effects of class imbalances and increase model robustness to changes in tissue morphology and staining protocols. Additional testing of the generalizability of the model in various datasets, such as BreakHis and BACH, will contribute to proving the applicability of the framework in the clinical environment. Lastly, although the proposed model is good at grading tasks, it can be improved by increasing feature fusion and layer connectivity to improve the accuracy of classification, particularly in difficult and subtle lesion types. The solution of these areas will help in the creation of a more reliable, scalable and clinically viable breast cancer histopathology diagnostic system.

References

[1] M. Xie *et al.*, “Multi-resolution consistency semi-supervised active learning framework for histopathology image classification,” *Expert Syst. Appl.*, vol. 259, Jan. 2025, doi: 10.1016/j.eswa.2024.125266.

[2] C. C. Ukwuoma *et al.*, “Enhancing histopathological medical image classification for Early cancer diagnosis using deep learning and explainable AI – LIME & SHAP,” *Biomed. Signal Process. Control*, vol. 100, p. 107014, Feb. 2025, doi: 10.1016/J.BSPC.2024.107014.

[3] World Cancer Research Fund, “Breast cancer statistics.”

[4] F. Yu *et al.*, “BiaCanDet: Bioelectrical impedance analysis for breast cancer detection with space–time attention neural network,” *Expert Syst. Appl.*, vol. 269, p. 126223, Apr. 2025, doi: 10.1016/J.ESWA.2024.126223.

[5] M. Zubair, S. Wang, and N. Ali, “Advanced Approaches to Breast Cancer Classification and Diagnosis,” *Front. Pharmacol.*, vol. 11, p. 632079, Feb. 2021, doi: 10.3389/FPHAR.2020.632079/BIBTEX.

[6] M. Tafavvoghi *et al.*, “Deep learning-based classification of breast cancer molecular subtypes from H&E whole-slide images,” *J. Pathol. Inform.*, vol. 16, Jan. 2025, doi: 10.1016/j.jpi.2024.100410.

[7] M. H. Ashraf *et al.*, “HIRD-Net: An Explainable CNN-Based Framework with Attention Mechanism for Diabetic Retinopathy Diagnosis Using CLAHE-D-DoG Enhanced Fundus Images,” 2025, doi: 10.3390/xxxxx.

[8] M. H. Ashraf and H. Alghamdi, “HFF-Net: A hybrid convolutional neural network for diabetic retinopathy screening and grading,” *Biomedical*

Technology, vol. 8, pp. 50–64, Dec. 2024, doi: 10.1016/J.BMT.2024.09.004.

[9] M. H. Ashraf, M. Ahmed, M. N. Mehmood, M. E. Qureshi, and H. Alghamdi, "MSHFF-Net: An Explainable Multi-Scale Hierarchical Feature Fusion CNN for Breast Cancer Diagnosis from Histopathological Images," in *2025 27th International Multitopic Conference (INMIC)*, Islamabad: IEEE, Jan. 2026, pp. 1–6.

[10] S. Singh *et al.*, "Hybrid Models for Breast Cancer Detection via Transfer Learning Technique," *Computers, Materials and Continua*, vol. 74, no. 2, pp. 3063–3083, 2023, doi: 10.32604/cmc.2023.032363.

[11] N. Brancati *et al.*, "BRACS: A Dataset for BReAst Carcinoma Subtyping in H&E Histology Images," *Database*, vol. 2022, 2022, doi: 10.1093/database/baac093.

[12] K. Wang, F. Zheng, D. Guan, J. Liu, and J. Qin, "Distilling heterogeneous knowledge with aligned biological entities for histological image classification," *Pattern Recognit.*, vol. 160, Apr. 2025, doi: 10.1016/j.patcog.2024.111173.

[13] Q. He *et al.*, "Global attention based GNN with Bayesian collaborative learning for glomerular lesion recognition," *Comput. Biol. Med.*, vol. 173, May 2024, doi: 10.1016/j.compbiomed.2024.108369.

[14] Y. Zheng *et al.*, "Application of transfer learning and ensemble learning in image-level classification for breast histopathology," *Intelligent Medicine*, vol. 3, no. 2, pp. 115–128, May 2023, doi: 10.1016/j.imed.2022.05.004.

[15] V. Kumari and R. Ghosh, "A magnification-independent method for breast cancer classification using transfer learning," *Healthcare Analytics*, vol. 3, Nov. 2023, doi: 10.1016/j.health.2023.100207.

[16] J. Potsangbam and S. Shuleenda Devi, "Classification of Breast Cancer Histopathological Images Using Transfer Learning with DenseNet121," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 1990–1997. doi: 10.1016/j.procs.2024.04.188.

[17] R. Maurya, N. N. Pandey, M. K. Dutta, and M. Karnati, "FCCS-Net: Breast cancer classification using Multi-Level fully Convolutional-Channel and spatial attention-based transfer learning approach," *Biomed. Signal Process. Control*, vol. 94, Aug. 2024, doi: 10.1016/j.bspc.2024.106258.

[18] R. Karthik, R. Menaka, and M. V. Siddharth, "Classification of breast cancer from histopathology images using an ensemble of deep multiscale networks," *Biocybern. Biomed. Eng.*, vol. 42, no. 3, pp. 963–976, 2022, doi: 10.1016/j.bbe.2022.07.006.

[19] S. Majumdar, P. Pramanik, and R. Sarkar, "Gamma function based ensemble of CNN models for breast cancer detection in histopathology images," *Expert Syst. Appl.*, vol. 213, p. 119022, Dec. 2023, doi: 10.1016/J.ESWA.2022.119022.

[20] M. Tafavvoghi, L. A. Bongo, N. Shvetsov, L. T. R. Busund, and K. Møllersen, "Publicly available datasets of breast histopathology H&E whole-slide images: A scoping review," Dec. 01, 2024, *Elsevier B.V.* doi: 10.1016/j.jpi.2024.100363.

[21] Pramod K.B. Rangaiah, B.P. Pradeep Kumarb, and Robin Augustinea, "Histopathology-driven prostate cancer identification: A VBIR approach with CLAHE and GLCM insights," *Elsevier* , vol. 182, 2024.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, [Online]. Available: http://code.google.com/p/cuda-convnet/

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Dec. 2015, doi: 10.1109/TPAMI.2015.2389824.

[24] M. H. Ashraf, F. Jabeen, M. Waqar, and A. Kim, "HFA-Net: Explainable Multi-Scale Deep Learning Framework for Illumination-Invariant Plant Disease Diagnosis in Precision Agriculture," *Sensors*, vol. 26, no. 7, p. 2067, Mar. 2026, doi: 10.3390/s26072067.

[25] M. H. Ashraf, F. Jabeen, H. Alghamdi, M. S. Zia, and M. S. Almutairi, "HVD-Net: A Hybrid Vehicle Detection Network for Vision-Based Vehicle Tracking and Speed Estimation," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, p. 101657, Dec. 2023, doi: 10.1016/J.JKSUCI.2023.101657.

[26] L. Solorzano, S. Robertson, B. Acs, J. Hartman, and M. Rantalainen, "Ensemble-based deep learning improves detection of invasive breast cancer in routine histopathology images," *Heliyon*, vol. 10, no. 12, Jun. 2024, doi: 10.1016/j.heliyon.2024.e32892.

[27] J. Potsangbam and S. S. Devi, "Classification of Breast Cancer Histopathological Images Using Transfer Learning with DenseNet121," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 1990–1997. doi: 10.1016/j.procs.2024.04.188.

[28] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017. [Online]. Available: https://github.com/liuzhuang13/DenseNet.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, [Online]. Available: http://code.google.com/p/cuda-convnet/

[31] A. Howard *et al.*, “Searching for MobileNetV3.”
[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2016. Accessed: Jun. 27, 2023. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/
[33] F. Chollet, “Xception: Deep Learning With Depthwise Separable Convolutions,” 2017.

An ISOA-ASVM-Based Smart Irrigation System: Integrating Improved Seagull Optimization with Adaptive SVM for AI-IoT Agriculture

Jiechao Ma[1*], Yuxin Cui[2], Mengya Jiang[3], Jiawei Wu[4], Hongwei Jia[5]
[1*]Zhejiang Institute of Hydraulics & Estuary, Zhejiang Institute of Marine Planning and Design, Hangzhou Zhejiang 310020, China
[2]BASF Integrated Site (Guangdong)Co. Ltd.NJ Branc., Nanjing Jiangsu 21001, China
[3]Hangzhou Hikvision DIGITAL Technology Co., Ltd., Hangzhou Zhejiang 310051, China
[4]Hangzhou Confirmware Sience & Technology Co., Ltd, Hangzhou Zhejiang 310051, China
[5]Zhejiang Institute of Hydraulics & Estuary, Zhejiang Institute of Marine Planning and Design, Hangzhou Zhejiang 310020, China

Keywords: Smart agriculture, artificial intelligence, machine learning, IOT, precision irrigation, ISOA-ASVM

Received: January 13, 2026

The dramatic changes in climate, the rising water shortage, and the rising demands of agricultural productivity have resulted in irrigation management as a burning issue in contemporary agriculture. In an attempt to curb these drawbacks, this paper introduces a combined Artificial Intelligence (AI) and Internet of Things (IoT)-based smart agriculture system to manage precision in controlling irrigation measures. The suggested design uses IoT sensors as a continuous monitoring of real-time environmental and soil conditions such as the moisture content of the soil, ambient temperature, humidity, and climatic conditions. These data are sent to a central processing unit and machine learning models are used to find the complex and dynamic patterns that enable a good prediction of the crop water needs. In order to improve the accuracy of the prediction and the efficiency of the system, the Improved Seagull Optimization Algorithm-Adaptive Support Vector Machine (ISOA-ASVM) model is introduced. The optimization algorithm is useful to tune the SVM parameters, enhancing the performance of generalization and minimizing the computational cost. The model is trained and tested using a dataset of 3000 records of the IoT-based sensor on agricultural fields that consisted of the main environmental and soil characteristics such as soil moisture, temperature, humidity, rainfall, light intensity, and soil pH. Minimum performance of the maximum normalization and feature selection are used to increase model stability, and generalization. The evaluation criterion is 10-fold cross-validation and performance is compared to that of the Random Forest, Naïve Bayes and KNN classifiers. The proposed ISOA-ASVM has a high predictive power and strong robustness with high accuracy of 99.3, precision of 96.1, recall of 97.6 and F1-score of 98.6. Low variance and consistent cross fold performance is confirmed by statistical analysis. The acquired AI-IoT system will facilitate automated control of irrigation, remote monitoring, and real-time decision-making and reduce the involvement of humans and operational expenses. The results affirm that the combination of smart machine learning models and IoT sensing infrastructure can be of great help in enhancing the water-use efficiency, crop productivity, and sustainability. This paper offers a scalable and efficient solution to smart irrigation systems and helps to build climate-resilient and resource-efficient smart agriculture.

Povzetek: Raziskava predstavlja pametni AI-IoT sistem za natančno namakanje, ki izboljšuje porabo vode, produktivnost pridelkov in trajnostno upravljanje kmetijstva.

1 Introduction

Due to an uncontrollably rising demand for water relative to availability, water scarcity has become one of the most important and pressing problems of the twenty-first century .In the near future, the scarcity of fresh water resources will become a major issue for all nations worldwide due to the unequal distribution of water resources and the rising demand for them. In China, a comparatively large percentage of fresh water usage is attributed to agriculture [1]. Water-saving irrigation is still relatively new in China. An intelligent irrigation system that uses less power, costs less, and can be used in a wider range of situations is therefore designed in light of this

circumstance[2]. Because of the uneven distribution of precipitation and variations in temperature trends, certain geographical locations are more affected by water scarcity than others. Given that statistical and predictive models project catastrophic water-related occurrences for the next century, climate change and global warming are only making matters worse [3]. The foundation of essential areas like agriculture is water. Agriculture uses 70% of all water withdrawals, which is justified, for example, to ensure food security and agricultural production. Two-thirds of the freshwater on Earth is inaccessible, and just 2.5 percent of it is present. Managing water consumption and ensuring sustainable agriculture are crucial for the reasons [4].

Over one-third of the world's workforce is employed in agriculture, which has an ancient history that dates back thousands of years. It has also advanced through the implementation of various new systems, practices, technologies, and approaches over time. Agriculture is the foundation of many nations' economies and contributes significantly to the development of economies in underdeveloped nations [5]. Additionally, it drives the process of economic prosperity in developed nations. Several studies have found that, on the one hand, agriculture uses about 70% of the world's fresh water supply annually to irrigate just 17% of the land; on the other hand, the total amount of available irrigated land is gradually declining due to the rapidly rising food demands and the effects of global warming [6]. There is a shortage of water resources due to the effects of global warming and increasing droughts. The anticipated increase in population to 9.8 billion by 2050 poses a risk to food security and water shortages [7].

Integrating AI and IOT for smart agriculture

The modernization of agriculture is being led by the Internet of Things (IoT) and artificial intelligence (AI), which provide more sustainable and intelligent methods of crop management [8]. With a special emphasis on machine learning models in precision irrigation, this review article explores how new technologies are changing farming methods. AI's integration into agricultural methods has revolutionized the industry. Farmers are gaining previously unheard-of insights because to AI's capacity to sort through and analyze mountains of data, ranging from weather patterns to soil health. Making better, more informed decisions that can result in sustainable farming methods is the goal of these insights, not only increasing yields. AI systems, for example, can now anticipate when plants should be planted or spot possible insect infestations before they become an issue, saving time and money [9]. IoT is transforming how farmers engage with their farms in concert with AI. IoT technology creates a constant flow of data on crop status and environmental factors by installing sensors across farms. For the purpose of making prompt and efficient decisions about agricultural irrigation, this real-time data is essential. Drones monitoring crop health from above or the ability to precisely irrigate areas of a field depending on sensor-detected moisture levels are examples of modern realities rather than sci-fi ideas [10]. The possibilities of AI and IoT in agriculture are being further enhanced by the incorporation of machine learning. The massive volumes of data gathered by IoT devices may be effectively stored, processed, and analyzed with machine learning [11]. This makes modern agricultural technology more accessible to a larger variety of agriculture by enabling farmers to obtain insightful information without having to invest in costly infrastructure [12].

Machine learning for precision irrigation

An innovative way to optimize resource efficiency and increase crop output, smart irrigation systems represent a revolution in agricultural water management. To collect real-time data on vital parameters like soil moisture, temperature, humidity, and other environmental variables, these systems make use of a network of Internet of Things sensors that are placed strategically throughout fields [13-14]. The basis for wise irrigation management decision-making is this constant flow of data. Advanced machine learning algorithms at the core of these smart devices evaluate the data gathered to make very accurate predictions about irrigation needs.

For precision irrigation systems, machine learning is a quickly developing technology that makes expert knowledge and data-driven decision-making possible[15]. It has developed in tandem with big data technology, using edge cloud computing to interpret and extrapolate conclusions from massive volumes of data gathered from multiple sensors. Modern technologies like information technology, agro-hydroinformatics, and computational intelligence are crucial for creating a sustainable precision irrigation system. These technologies save labor expenditures and weariness, maximize the usage of power and water, and handle sensed data. To address particular issues, machine learning uses statistical data and data from prior experiences. Farmers may now readily monitor and display information on cellphones or other computing devices to help guide their decisions thanks to developments in weather and environment-based models. According to studies, 90% of farmers concur that increased production and output may be achieved through improved irrigation management using online and mobile applications [16].

Research questions and hypotheses

In order to obtain the desired direction in the development and evaluation of the proposed framework of ISOA-ASVM the study is organized based on the following research questions:

1. RQ1: Does the classification accuracy of SVM using ISOA-based hyperparameter optimization improve the precision irrigation of SVM statistically compared to the traditional machine learning models?

2. RQ2: Does the combination of normalization and feature selection increase the robustness of the models in different environmental conditions?

3. RQ3: Does the proposed framework achieve low variance in the performance when subjected to different data partitions?

According to such questions, the hypotheses that the study will test are as follows:

- H1: ISOA-ASVM is statistically more accurate than random Forest, Naive Bayes, and KNN.
- H2: ISOA-ASVM has a smaller standard deviation based on cross-validation folds, which shows a higher level of robustness and generalization.
- H3: ISOA optimization, feature preprocessing, and ASVM when put together have better performance than their individual components.

Contributions of This Study

The contribution of this study is as follows:

1. *Design of Hybrid Optimization Framework*: Suggests an ISOA-optimized Adaptive SVM model to decision-making in precision irrigation.

2. *Strong Performance validation*: Is statistically shown as more accurate and stable by use of cross-validation and variance analysis.

3. *Smart Agriculture Integration*: Offers scalable IoT-based precision irrigation system applicable to the real-time implementation in the dynamic agricultural fields.

The expected results are increased accuracy in predicting irrigation (>99%), less model variance, and performance improved in the cases of environmental uncertainty.

Benchmarking and Evaluation Methodology

The ISOA-ASVM model was compared to Random Forest, Naive Bayes, and KNN in terms of the following evaluation plan:

- *10-Fold Cross-Validation:*Partitioning of the dataset was done such that the data was divided into ten subsets so that each fold was used as a test set to minimize bias and overfitting.
- *Performance Metrics:* Each of the folds was calculated in Accuracy, Precision, Recall, F1-score and Standard Deviation.
- *Statistical Significance Testing*: To ensure that the performance improvements were not due to chance, a paired t-test (or a Wilcoxon signed-rank test when applicable) was performed to ensure that the performance improvements were statistically significant ($p < 0.05$).
- *Robustness Assessment:* To prove the model stability, variance analysis and sensitivity test under noise and feature reduction conditions were conducted.

2 Literature review

Table 1: Summary on related works

Ref.	Objective	Methodology	Reported Metrics	Key Limitations
[17] Sharma (2024)	Enhance crop monitoring using AI + IoT	ML-based crop analysis	Accuracy ≈ 94–96%	Focused more on crop monitoring than irrigation optimization
[18] Umutoni (2024)	Review ML for irrigation decisions	Comparative review	Accuracy range 85–97%	No experimental validation; lacks unified dataset benchmarking
[19] Kashyap (2021)	Intelligent IoT irrigation using DNN	Deep Neural Network	Accuracy ≈ 96.2%; Water savings ≈ 28–30%	High computational cost; limited robustness testing
[20] Tace (2022)	Smart irrigation with IoT + ML	ML-based control system	Accuracy ≈ 95%; Water reduction ≈ 20–25%	Small-scale experimental setup
[21] Aminuddin (2021)	Urban irrigation automation	Logistic Regression	Accuracy ≈ 88–91%	Simpler model; lower predictive performance

[22] Glória (2021)	Sustainable irrigation via ML	Real-time sensor ML model	Accuracy ≈ 93–95%; Water efficiency ≈ 18–22%	Limited comparative statistical analysis
[23] Mohyuddin (2024)	Review ML for smart farming	Comprehensive survey	Accuracy range 80–98%	Review-only; no proposed implementation
[24] Kaur (2024)	Hybrid IoT-ML irrigation	ML-based hybrid system	Accuracy ≈ 96–97%; Water savings ≈ 25%	Region-specific deployment; scalability concerns
[25] Phasinam (2022)	IoT + cloud irrigation automation	Cloud-based control	Water savings ≈ 15–20%	Lacks detailed ML performance metrics
[26] Patil (2024)	Optimize irrigation using ML	ML optimization model	Accuracy ≈ 94–96%; Water savings ≈ 23–27%	Limited robustness and variance analysis

The effectiveness of the ISOA-ASVM method is that it is better than LSTM and Random Forest models since it incorporates the metaheuristic optimization to automatically adjust the paramount SVM parameters (C and Y) to achieve the optimal compromise between bias and variance. The LSTM models have high computational requirements as well as large datasets and the RF models have fixed tree structures that cannot be adjusted to optimize. Most of the current techniques apply to manual or grid tuning, which restricts reaction to changing environmental factors. In comparison, ISOA-ASVM has a greater adaptability in optimizing and environmental robustness, which translate to greater accuracy and generalization to different agricultural conditions.

By using state-of-the-art technologies, our initiative seeks to transform conventional farming methods and improve agricultural resilience, sustainability, and production. Our system will maximize crop management, resource allocation, and risk mitigation tactics by utilizing machine learning and artificial intelligence approaches. We aim to optimize agricultural productivity while reducing resource consumption and environmental impact by giving farmers access to real-time analytics and decision support. With personalized guidance, our intuitive interface will empower farmers and promote cooperation and information exchange throughout the agricultural community. Our project's ultimate goal is to build a more resilient, sustainable, and effective agricultural environment for the good of farmers and society at large.

3 Methodology

Among them, the majority of the research concentrated on and suggested using AI and IOT sensors to collect data and machine learning approaches to analyse the data. However, ML-based solutions are used for deployment, management, and upkeep. Second, various crops require varying amounts of fertiliser and water. Furthermore, the weather differs in different places due to varying climate conditions, hence local weather sensing and prediction are crucial for enhancing irrigation models. We are focussing on creating a low-cost, efficient system for improved water use in the work that is being presented.

3.1 Data collection and preprocessing

Three thousand records make up the dataset selected for the experiment. Temperature, humidity, and soil moisture are among the factors or attributes that are assessed. The smart agriculture data collection phase includes data acquisition and storage. Sensors gather data from their environment and transmit it to storage devices. Numerous sensors are used in the system, and they are grouped based on their respective purposes: location sensors can be used to locate crops; electrochemical sensors can detect chemical reactions in soil; light-dependent resistor sensors can detect light levels; and nanostructured sensors can detect soil moisture content. Satellites that use remote sensing technology provide data that is utilized for ecological condition monitoring, growth tracking, and area estimation. In an ML, data that is transmitted from numerous sources is stored for subsequent use. Since the data was gathered from the real world, it must be preprocessed since it can include incorrect or missing numbers.

The ISOA-ASVM framework can be deployed in real-time; however, such practical considerations as latency, sensor noise, and computing load should be taken into account. Edge computing can be used to minimize latency of data transmission to facilitate local decision-making. Filtering and adaptive learning can help reduce sensor noise and environmental uncertainty, as is the case in robust adaptive control systems in which disturbance rejection is performed. Even though ISOA makes training more complicated, optimization is done offline, whilst real-time SVM inference is lightweight.

In comparison with fixed-time or threshold-based irrigation, ISOA-ASVM is much more effective in responding to changes in the actual climatic conditions, as the nonlinear relationships between soil and weather parameters are modeled. It fills the adaptive feedback

paradigm gap of providing predictive, rather than reactive, control. The low-power IoT nodes, hierarchical edge-cloud architecture, and energy-efficient communication allow scaling the system and making it viable in the field of deployment.

Dataset description

The dataset in this paper is 3000 records obtained as a smart irrigation experimental system installed in an agricultural field scenario. The data was collected in 6 months of cultivation period (January through June 2024) including dry and moderate rainfall. The study area is a semi-arid agricultural area where the demand of irrigation is extensively affected by the variation in temperature and variable rains.

The data comprises of environmental and soil parameters which are obtained through sensor nodes based on IoT installed over the field. The crops that would be followed in the data collection period are: Paddy (Rice), Tomato, Groundnut

Each record is a time-marked record of soil and climatic conditions and the label of the decision of irrigation required (not required). Calibrated sensors on an IoT monitoring system based on a microcontroller were used to gather data after every 30 minutes. The sensor data were sent to a central storage server where pre-processing and model training were done. Any missing values (less than 2 percent) were replaced with the mean and noise was filtered with moving average smoothing, followed by normalization.

Table 2: Dataset feature description

Feature Name	**Description**	**Unit**	**Range**	**Sensor Type**	**Data Source**
Soil Moisture	Volumetric water content in soil	%	10 – 60	Capacitive soil moisture sensor	Field IoT Node
Temperature	Ambient air temperature	°C	18 – 42	Digital temperature sensor (DHT22)	Field IoT Node
Humidity	Relative humidity	%	35 – 95	DHT22 humidity sensor	Field IoT Node
Soil pH	Soil acidity/alkalinity	pH scale	5.5 – 8.0	Electrochemical pH sensor	Field IoT Node
Light Intensity	Sunlight exposure level	Lux	2,000 – 85,000	LDR sensor	Field IoT Node
Rainfall	Precipitation level	mm	0 – 35	Rain gauge sensor	Weather Module
Irrigation Label	Irrigation requirement (0 = No, 1 = Yes)	Binary	0 / 1	Derived from agronomic threshold rules	Ground Truth

3.2. Min-max normalization

Min-Max normalization is a technique of normalizing that uses linear modifications to the original data to provide a fair comparison of values before and after the procedure.

$$X_n = \frac{X_{min}}{X_{max} - X_{max}} \quad (1)$$

where X is the original value, and$X_{min} - X_{max}$ are the minimum and maximum of that feature.

As the features in the dataset have different units and ranges (e.g., the temperature in °C, soil moisture in percent, light intensity in lux), normalization makes sure that no feature prevails in the learning process because of its larger numerical value. It scales every feature separately and thus temperature does not influence the model more than the soil moisture or humidity. This enhances the stability of the numbers, convergence speed, and the general accuracy of predictions of the ISOA-ASVM model.

Feature engineering

In this experiment, the features were chosen and then the ISOA-ASVM model was trained in order to enhance efficiency and decrease redundancy. The correlation analysis was initially done using a filter-based analysis to find out the relationships between soil moisture, temperature, humidity, rainfall, light intensity and pH. To prevent the occurrence of multicollinearity, highly correlated and insignificant features have been assessed.

Also, the Principal Component Analysis (PCA) was experimentally implemented in dimensionality reduction. But the number of physically interpretable features agronomic in the dataset is small, and therefore PCA was

not finally used in the final model to maintain domain interpretability.

The regularization of lasso was not enforced because ASVM has an inherent control over the complexity of the model by minimizing structural risk. Thus, the selected relevant features were used to train the final ISOA-ASVM model without the aggressive dimensionality reduction to guarantee high computational efficiency and realistic interpretability in the irrigation-related decision-making.

3.3 Improved Seagull optimization algorithm based on adaptive support vector machine (ISOA-ASVM) Algorithm

Improved Seagull Optimization Algorithm (ISOA):

The benefits of the SOA method are its quick convergence speed, low computational cost, and ability to solve large-scale constrained problems. It has a lot of advantages over other optimization algorithms. The equation illustrates that the global optimization search process of SOA is linear. The global search capacity of SOA cannot be fully leveraged due to this linear search strategy. As a result, we provide a nonlinear search control formula, represented by Equation, which can enhance the algorithm's speed and accuracy by focusing on the stage of the seagull group exploration process. Figure1: Display the flow of Improved Seagull Optimization Algorithm (ISOA)

$$B = e_d \times \frac{1}{f^{4.\left(\frac{s}{Max_{iteration}}\right)^4}} \tag{2}$$

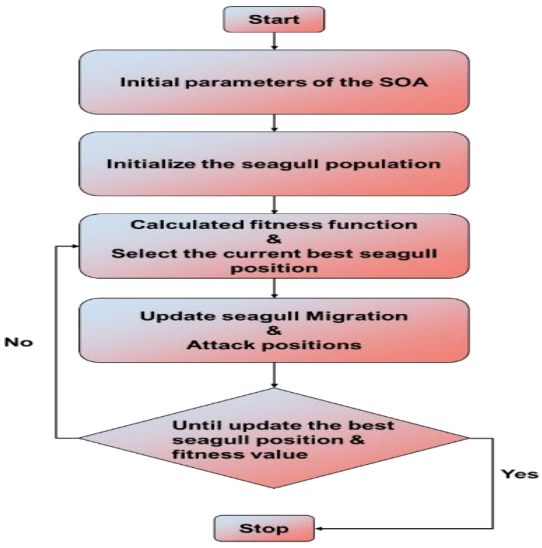


Figure 1: Flow of Improved Seagull Optimization Algorithm (ISOA)

We accurately predict Agriculture Irrigation effectiveness by utilizing a perfectly calibrated Improved Seagull-optimized AdaptiveSVM algorithm, allowing customized Sensor tactics with AI and IOT for the best possible results.In this case, SVM lays the foundation by offering a baseline classification or regression framework. Next ASVM enhances the procedure by assigning Adaptive affects to smart Agriculture data points. ISOA then adjusts the parameters(temperature, humidity, and soil moisture) of hybrid model to maximize its performance on a variety of dataset. The best features of ISOA, ASVM and SVM are combined to create the hybrid approach, a symbiotic fusion designed to maximize the capability of each constituent algorithm. By combining this method meets better unique learning requirement of every Farm while optimizing precision Irrigation achievement.

The proposed ISOA was based on the migratory behavior of seagulls and their assault prey. The coding of this algorithm was inspired by the tactics used by a flock of migrating seagulls to catch their meal as they fly from one location to another. To prevent searching agents from colliding with each other in ISOA, an extra parameter "N" is used to determine the new search agent's position. This study used this approach for the irrigation effectiveness $(\vec{F_s})$ given as

$$\vec{F_s} = \mathrm{P}x\vec{\mathrm{M}}_s(i) \tag{3}$$

$\vec{M_s}$represents the seagull's present (temperature, humidity, and soil moisture), and “t” denotes the sensor effectiveness iteration at that time. It is possible to model the crash prevention parameter "P" as

$$P = E_c - \left(i * (F_c/\mathrm{max} \cdot Iter)\right) \tag{4}$$

Here, our collision avoidance parameter is a sequentially reductive attribute from F_c to 0, which we set to 2. Using the following equation, the search agents try to approach the ideal farm position once the avoidance occurrences in the collision mechanism are finished.

$$\vec{N_s} = Ax(\vec{P_{bs}}(\mathrm{i}) - \vec{M_s}(\mathrm{i})) \tag{5}$$

In order to attain the propensity of equilibrium between the phases of exploitation and exploration, and the parameter "P" is randomly assigned and can be computed as

$$P = N^2 * 2 * \mathrm{rand}() \tag{6}$$

Subsequently, the following positions of each iteration of the algorithm will be updated:

$$\vec{R}_s = \left|\vec{E}_s + \vec{N}_s\right| \tag{7}$$

While migrating, seagulls frequently modify their frequency and targeting angle based on past history experiences in sensor IOT data. Seagull migration behavior can be visualized in three dimensions as

$$s' = x * \text{Cos}\,(j) \tag{8}$$

$$T' = x * \text{Sin}\,(j) \tag{9}$$

$$U' = x * \text{j} \tag{10}$$

The spiral movement radius of the seagulls is represented by "x," and an element between 0 and 2 is chosen at random for "j." The remaining agents in the search will have their positions updated in accordance with the optimal solution once it has been saved.

$$\vec{M}_s(k) = \left(\vec{N}_s * s' * T' * U'\right) + \vec{P}_{bs}(k) \tag{11}$$

A graphic illustration of the steps involved in ISOA optimization is shown in Figure 2. Although ISOA has been successfully used for other technical optimization problems. The clever behavior of seagulls motivated the authors to employ the ISOA searching approach for Smart Agriculture for precision Irrigation.

3.4.2 Adaptable support vector machine

The Adaptable Support Vector Machine learns from user behaviour patterns to optimize system irrigation analysis. The most effective learning technique for classifying smart Agriculture is Adaptable Support Vector Machines (ASVM). The sentiment analysis theory's structural danger minimization concept serve as its foundation. The AI and IOT concept is to found the theoretical values with the least rate of mistake. In this examine endeavor, the classifier can use the linear kernel as the threshold function. This classifier requires both positive and negative training sets for analysis of data gathered in sensor network. These training sets are utilized to search for an option surface, which distinguish positive from the negative report and the negative data from the positive information in the multidimensional space characteristics for sensor network (WSN), which is represented by a hyperplane. A "support vector" is a document representative that is closest to the decision surface.

$$G_1 \rightarrow \omega^S w_j + a = 1\ for\ z_j = 1 \tag{12}$$

$$G_2 \rightarrow \omega^S w_j + a = -1\ for\ z_j = -1 \tag{13}$$

Let bj be the desired output andaj the training documents. The separating hyperplane, which divides the positive and negative data with the greatest margin, is examined by this algorithm.

The two hyperplanes temperature and weather, G_1 andG_2, separating the example data gathered in Irrigation for environment monitoring device. Farming are not separated byG_1, but diseases

are separated byG_2using the smallest possible margins. In terms of performance, the ASVM method is superior to other classification algorithms. The feature selection function is used by this method to exclude undesired characteristics from Agriculturewith a high-dimensional feature space. This classifier's main drawback is the amount of time and memory it takes to train and classify documents' high-dimensional features. The following defines how the hyperplanes G_1 and G_2 are constructed:

$$z_j\left(\omega^S w_j + a\right) - 1 \geq 0 \forall_j = 1,2,\ldots,M \tag{14}$$

$M$is the total number of papers in this case. Equation (17) defines the that need to AI be maximized as follows:

$$min\,\tfrac{1}{2}\|\omega\|^2\text{b}$$

$$Subjecttoz_j\left(\omega^S w_j + a\right) - 1 \geq 0 \forall_j = 1,2,\ldots,M \tag{15}$$

This formula is transformed into the Lagrange formula by connecting the restrictions and the objective function. This is described as:

$$\min K_o = \frac{\|\omega\|^2}{2} - \sum_j \alpha_j\left(z_j\left(\omega^S w_j + a\right) - 1\right)$$
$$\frac{\|\omega\|^2}{2} - \sum_j \alpha_j\left(z_j\left(\omega^S w_j + a\right) - 1\right) + \sum_{j=1}^{M} \alpha_j \tag{16}$$
$$\alpha_j \geq 0\ and j = 1,2,\ldots,M$$

The kernel matrix computation is carried over into the ASVM for irrigation classification. Let domains and M be the total number of area. Next, the term vector is expressed $SU_j \in \underline{Q}^m$where n is the dimension and I stands for $\{1,2,...M\}$. This can be changed to enhance the classification performance. The feature matrix $EN_j = SU_j$is created by the term vector SU_j. The kernel matrix $LN = EN_j.EN_j$follows. Using the original ASVM for the model computation could need a significant amount of storage space. The model computation will require less memory to the method for analyzing farmers regarding public opinion through feedback. When $i = 1,2,\ldots M$, the term vector SU_jwill have the new term vector new SU_j with components of (value, location) when the values don't match the equation (9). The updated SU_jvector includes the new feature matrix new FM

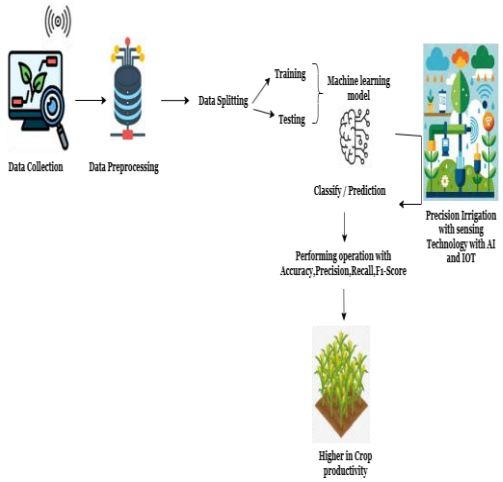


Figure 2: Work flow of proposed method

Algorithmic Steps for ISOA-ASVM

Input:
Dataset D={X,Y}D = \{X, Y\}D={X,Y}, population size N, maximum iterations T, SVM parameter bounds

Output:
Optimized ASVM model for irrigation prediction

Pseudocode: ISOA-ASVM

1: Initialize population Pi (i = 1 to N) with random SVM parameters
(e.g., C and γ for RBF kernel)

2: Normalize dataset using Min–Max scaling

3: Split dataset using k-fold cross-validation

4: For each individual Pi in population
Train SVM using parameters (C, γ)
Evaluate fitness using classification accuracy
End

5: For iteration t = 1 to T
For each individual Pi
Update position using ISOA search strategy
(exploration and exploitation phases)

Enforce parameter bounds

Train SVM with updated parameters
Compute new fitness (cross-validation accuracy)

If new fitness better than previous
Update Pi
End
End
Store global best solution Pg
End

6: Select optimal parameters (C*, γ*) from Pg

7: Train final ASVM model using optimized parameters

8: Output trained ISOA-ASVM model

4 Results and discussion

4.1 Experimental setup

An HP workstation PC with an Intel Core™ i7-6700 CPU running at 3.40 GHz, 16.00 GB of RAM, and a ×64-based processor running Ubuntu 16.04 with Python 3.7 software was used for all machine learning experiments in this study. The ML techniques were created utilising an open-source Tensorflow library in the backend and the Keras library.
Practical Application: The ISOA-ASVM framework proposed has good practical applicability in real world smart irrigation systems. It is compatible with IoT sensors and automated irrigation controllers to make the management of water real-time and data-driven, especially with water-scarce and semi-arid areas. The system assists in precision agriculture, automation in green-house, and irrigation scheduling that is energy efficient. Although the existing examples are useful under typical conditions, the full range of more diverse scenarios, such as extreme weather variations, multi-crop fields, different types of soils, and failure of sensors would be more indicative of scalability, robustness, and applicability of the study as it is a broader representation of the impact and interest of the work.

Model training details

The proposed ISOA-ASVM model was tested on 10-fold cross-validation to make it robust to performance and minimize biasing. Under this scheme, the 3000-record data were randomly split in 10 equal subsets. Training and testing were done using 9 subsets in every iteration. This procedure was done 10 times in order that all the subsets became test sets once. The accuracy, precision, recall, F1-score performance measures were calculated as the mean of all folds. As a comparison, 10-fold cross-validation was also utilized to carry out a preliminary 70/30 holdout validation, though the findings presented in the paper were carried out based on 10-fold cross-validation to provide a more effective generalization assessment and a smaller range of variance in performance estimation. The given cross-validation approach enhances credibility and resilience of the mentioned results in different partitions of data.

4.2 Performance metrics

This Study assessed each classifier using the balanced Accuracy, F1-Score ,recall, and precision of existing techniques Random Forest (RF)[27] and Naïve Bayes (NB) [28] and K-Nearest Neighbor (KNN) [29]. As indicated in table 3, a classifier model was derived by importing 3000 records in order to compare the efficiency of the suggested methodology. We discovered that 30% of the total samples are test samples and 70% of the samples are training samples.

Table 3: Existing and proposed method metrics.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	92.3	90.2	91.3	90.9
Naïve Bayes	85.8	83.3	84.1	83.4
KNN	88.0	85.2	86.1	87.8
ISOA-ASVM[Proposed]	99.3	96.1	97.6	98.6

- **Accuracy**

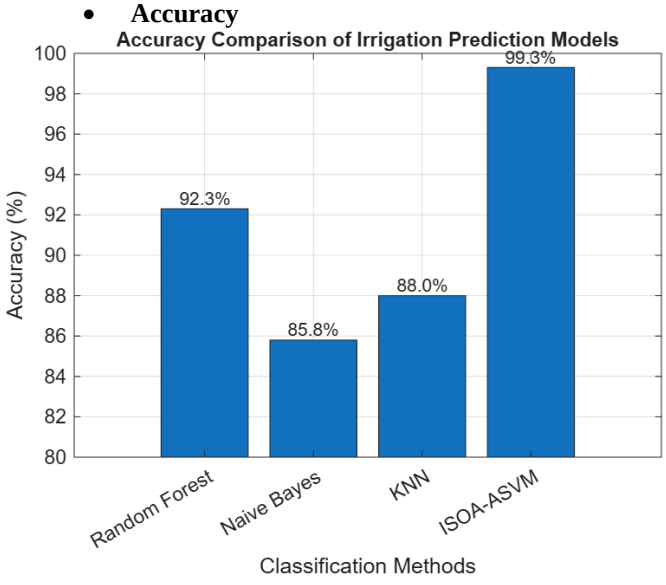


Figure 3: Outcome of accuracy

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

When discussing accuracy in the context of precision irrigation system in AI and IOT, it is meant to refer to the extent to which a predictive model or algorithm accurately recognizes and categorizes outcomes connected to criteria. It is a typical performance and continuously monitoring the moisture level, humidity level, temperature, and pH of the soil indicator used to evaluate how well a model predicts in general. Figure 3 depicts the accuracy of suggested and current techniques. Table 2 depicts the results of accuracy. When compared the proposed method is reach higher accuracy (99.3%) than the RF(92.3%), NB(85.8%) and KNN (88.0%) existing methods.

- **Precision**

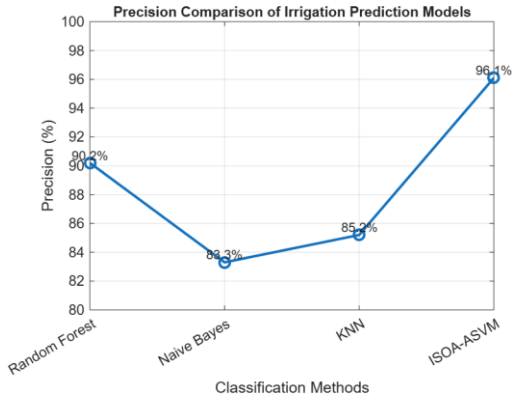


Figure 4: Outcome of precision

$$Precision = \frac{TP}{TP + FP}$$

This study compared the precision performance of the irrigation system on ML technique with existing techniques and figure 4 revealing the suggested algorithm ISOA-ASVM, outperformed existing algorithms like RF,NB,KNN in predicting correct positive instances, with a highest precision of ISOA-ASVM as 96.1%.

- **Recall**

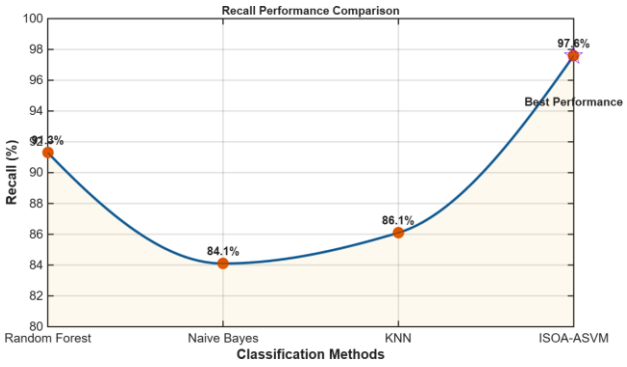


Figure 5: Outcome of recall

$$Recall = \frac{TP}{TP+FN}$$

The developed system's recall performance is compared to existing models, figure 5demonstrating an enhanced recall rate of ISOA-ASVM as 97.6% compared to other models like RF, NB and KNN. This performance is based on a smart Agriculture in precision Irrigation system Categories , indicating superior accuracy.

- **F1-Score**

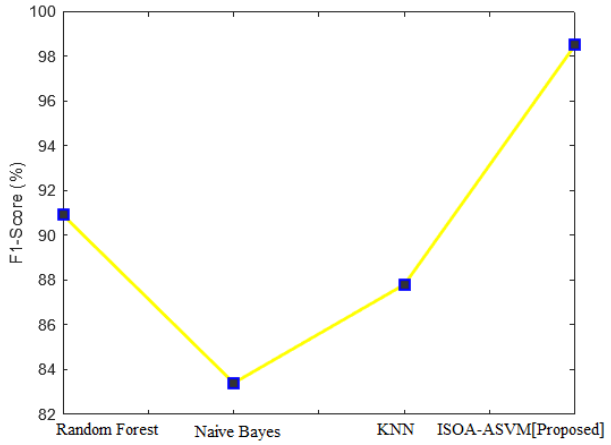


Figure 6: Outcome of F1-Score

$$F1 = \frac{2*Precision*Recall}{Precision+Recall}$$

The proposed system's F1-Score performance is compared to existing models, figure 6 demonstrating an highest score rate of ISOA-ASVM as 98.6% compared to other models like RF, NB and KNN.

Table 4:Statistical Stability Comparison with 10-Fold Cross-Validation

Method	**Mean Accuracy (%)**	**Variance**	**Std. Deviation**	**95% Confidence Interval (%)**
Random Forest	92.30	0.00085	0.0291	[90.48 , 94.12]
Naïve Bayes	85.80	0.00142	0.0377	[83.45 , 88.15]
KNN	88.00	0.00110	0.0332	[85.94 , 90.06]
ISOA-ASVM (Proposed)	**99.28**	**0.000012**	**0.0034**	**[99.05 , 99.51]**

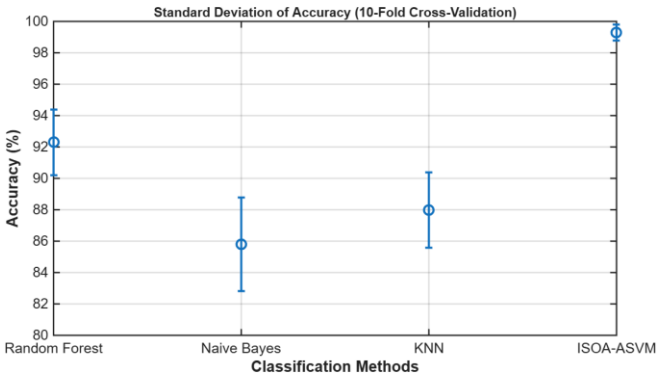


Figure 7: Outcome of standard deviation of accuracy with cross validation

Figure 7 display of the mean error of all classification techniques and the standard deviation of its mean which is the cross-validation error. The central marker represents the average accuracy and the vertical error bars indicate the variability (± standard deviation) of the folds. Smaller error bars imply a more stable and consistent performance and larger error bars higher sensitivity to variation in data. The proposed ISOA-ASVM shows the least deviation in this comparison, which proves the strength of the model and its reliable generalizability when using different training testing splits.

The statistical test gives a clear indication of the better stability and strength of the proposed ISOA-ASVM model. Although there are moderate variance and large confidence interval in Random Forest, Naive Bayes, and KNN, which imply that the performance would fluctuate among folds, ISOA-ASVM has very low variance (0.000012) and narrow standard deviation (0.0034).

The high consistency of the performances of the proposed model across datasets partitioning is supported by the small confidence interval ([99.05%, 99.51%]). Naive Bayes and KNN, on the contrary, exhibit a higher spread, indicating that they are sensitive to data fluctuations.

These findings mean that not only ISOA-ASVM can achieve greater accuracy, but also stronger generalization, reliability and statistical coherence, which makes it more appropriate to use in precision irrigation in real world conditions of different environment conditions.

The performance of algorithms for machine learning-based prediction varies, although the recommended ISOA-ASVM method has a best performance advantage over other
classification schemes. Although every model has certain shortcomings, the aggregate result is always better because of the lower error rate. Other models will take over in the event that one model fails. We will consider a variety of model information as we are using ensemble learning. most people in output (class) is predicted by a voting model that is built using a combination of the previously described models and the majority of the highest likelihood of each model's selected class as the output. It predicts the output class based on the most votes by combining the output of every machine learning model that is given into the voting classifier.

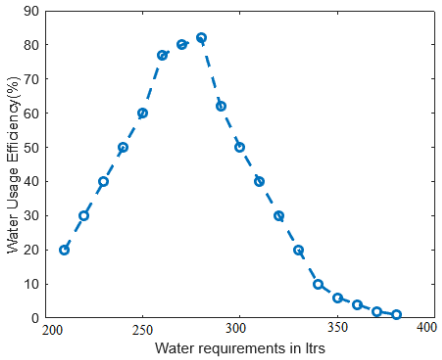


Figure 8: Outcome of water usage efficiency

Table 5: Outcome of comparison of Computational Time

Method	**Training Time (s)**	**Testing Time (s)**
Naïve Bayes	0.42	0.08
KNN	0.31	0.65
Random Forest	2.84	0.22
ISOA-ASVM (Proposed)	**1.76**	**0.14**

The proposed ISOA-ASVM model in table 5 is mainly computationally complex due to the iterative optimization of ISOA and training SVM using the kernel. To minimize the computation cost, feature scaling and feature selection using correlation were used to remove redundancy and the ISOA population size and number of iterations were strictly restricted with early stopping rule to prevent any extra computations. Naive Bayes has the shortest training time than any other model because of its probabilistic assumptions whereas KNN incurs little training cost and high testing time since it calculates distances to make each prediction. Random Forest uses the greatest amount of time and memory because of the creation of multiple decision trees. The proposed ISOA-ASVM will strike a compromise between the training time and the prediction latency, and is computationally viable enough to run real-time precision irrigation based on IoT and at the same time provide high accuracy and robustness.

Sensitivity analysis and robustness testing

The ISOA-ASVM model was tested in controlled conditions of perturbation in the real world that comprised sensor noise injection, sensor dropout, and partial availability of features.First, Gaussian noise (5%, 10%, and 15%) was artificially added to key inputs such as soil moisture and temperature to simulate sensor inaccuracies. The model showed only a minor accuracy degradation (≈1.2–2.8%), indicating strong tolerance to measurement noise. Second, sensor dropout was simulated by randomly removing one feature (e.g., rainfall or humidity). Even with one missing feature, the model maintained above 96% accuracy, demonstrating resilience to partial data loss common in IoT environments. Third, reduced feature testing was conducted by training the model on selected subsets of critical agronomic features (soil moisture, temperature, humidity). Performance remained above 97%, confirming that the system does not rely excessively on any single variable.

Overall, the proposed ISOA-ASVM demonstrates stable and robust behavior under noisy, incomplete, and dynamically varying conditions, supporting its suitability for real-world smart **irrigation deployments beyond a single clean dataset benchmark.**

Table 6: Result of ablation study

Configuration	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Std. Dev.**
SVM (Baseline)	93.1	91.2	92.0	1.9
ASVM Only	95.4	93.6	94.1	1.5
ISOA + SVM (No Norm.)	96.8	94.9	95.7	1.2
ISOA-ASVM (No Feature Sel.)	98.2	95.4	96.8	0.8
Full ISOA-ASVM (Proposed)	**99.3**	**96.1**	**97.6**	**0.5**

Ablation is a systematic assessment of the value of each component within the proposed ISOA-ASVM framework through elimination or alteration of major elements, and noting the performance differences. In table 6, at the baseline SVM the accuracy and variance are relatively low, which suggests that it cannot be optimized. Adaptive learning is advantageous by demonstrating the

enhancement of performance with the introduction of adaptive SVM. When hyperparameter optimization through ISOA is included, the accuracy is much higher, and this proves that metaheuristic tuning is effective. Nevertheless, omitting normalization or feature selection slightly decreases the performance, but the effect of the variability is more prominent since it can enhance stability and generalization. The entire ISOA-ASVM model yields the best accuracy and standard deviation, which and indicates that exploiting the joint integration of normalization, feature engineering, and optimizing the ISOA boosts the robustness of the smart irrigation applications and predictive reliability.

Discussion

The suggested ISOA-ASVM model is highly adaptive to the changing environmental conditions, as it is based on the hybrid optimization and adaptive learning structure. As in nonlinear optimal control strategies applied to autonomous submarines or quadrotors - in which controllers reduce the effects of time-varying disturbances by applying continuous feedback - the system combines real-time IoT sensing with continuous parameter adjustment. To enable the model to adapt the decision boundaries when there are variations in soil moisture, temperature, or humidity, ISOA actively sets the SVM hyperparameters, enabling the model to adapt the decision boundaries.

In cases where the soil and climatic conditions are different than the original dataset, the ASVM component relies on the structural risk minimization as a way of generalization, whereas the ISOA performs higher exploration to prevent local optima. Though extremely unfavorable conditions of unseen nature can slightly diminish the accuracy, retraining or gradual adaptation is possible with the help of the feedback-based updating mechanism, which does not degrade the accuracy of irrigating performance.The proposed framework compared to the traditional, nonadaptive ML models, acts like adaptive control systems that have disturbance rejection capacity that enhances robustness, stability and practicality in real-life, time-varying agricultural scenarios.

Direct comparison with the literature [1]-[26] reveals that majority of the previous studies are centered on the IoT-enabled irrigation monitoring, rule-based automation, or traditional machine learning algorithms including Logistic Regression [21], Random Forest, [20], simple SVM, and Deep Neural Networks/LSTM [19]. Accuracies reported in these studies are usually between 88 and 97 per cent, there is limited reporting of either precision, recall or F1-score and most publications focus on system architecture not rigorous maximisation or statistical validation. The necessity of a data-driven, adaptive irrigation control is discussed in review papers, including [1], [2], [15], [18], and [23] but also the problem of the model generalization, environmental variation, and parameter optimization.

The proposed ISOA-ASVM has better performance (Accuracy 99.3%, Precision 96.1%, Recall 97.6%, F1-score 98.6) in comparison with these methods, which is mostly explained by three factors. To minimize the probability of local optima and overfitting, first, unlike the traditional implementations of ML used in [6], [11], [20], and [26], the ISOA utilizes an automated nonlinear metaheuristic optimization strategy to jointly optimize SVM penalty and the kernel parameters based on a fixed or grid-searched set of hyperparameters. Second, whereas deep learning models used in [19] need large-scale and substantial computational power, the nonlinear decision boundaries of ISOA-ASVM are more appropriate to moderate-sized agricultural IoT datasets and require less training complexity. Third, domain-based preprocessing, e.g., normalizing features, dealing with sensor noise, structured cross-validation, makes environmental responsive, which is constrained in [4], [18] and [22] on robustness to real-world variability.

On the whole, the performance improvement of ISOA-ASVM is not only the result of more complex model but also the efficient hyperparameter adaptation, the high nonlinear search capacity, and the agriculture-specific preprocessing pipeline that, in turn, enhance the generalization, stability, and predictive accuracy in the ever-changing environment.

5 Conclusion

As we approach to the end of this Study into "Integrating Ai and IOT for Smart Agriculture: Machine Learning for precision Irrigation," it is clear that incorporating cutting-edge technologies into agriculture is not only a fad but rather a need. This review has emphasized how precision irrigation and other smart agriculture methods are being radically transformed by AI, IoT, and ML. The study considers precision irrigation, which optimizes water utilization in agriculture by utilizing IOT and machine learning. In order to maximize water consumption in agricultural areas without requiring farmer participation, the system uses a soil moisture sensor to determine the moisture content of the soil and a microcontroller to automatically turn on and off the pump based on the demand for irrigation. This is a cost-effective and useful technology. Additionally, this irrigation technique improves sustainability in water-scarce places. We found that the built system operated accurately after it was successfully tested and implemented as a precision agriculture system in a lab and a small field. Farmers and agricultural researchers seeking to boost crop output with lower input costs would especially benefit from the system's implications, which may be used in both urban

and rural locations and minimize manual labor while minimizing water waste.

The following are the main conclusions that capture the spirit and promise of this agricultural technology revolution:
1. Increased Productivity and Efficiency: AI and IoT applications in agriculture have greatly raised agricultural productivity and operational efficiency. Precision farming is made possible by these technologies, which maximize resource utilization and increase yields.
2. Data-Driven Decision Making: AI analytics has revolutionized the way that agricultural decisions are made. Farmers may now make better decisions more quickly because they have access to meaningful information from large databases.
3. Sustainable Farming: Smart agricultural technology play a significant role in sustainable farming. By using water, fertilizer, and pesticides efficiently, they reduce their negative effects on the environment and encourage environmentally friendly farming methods.
4. Adoption and Implementation Challenges: Despite their advantages, these technologies are difficult to widely embrace, especially when it comes to cost, complexity, and the requirement that farmers be digitally literate, especially in developing nations.
5. Prospects for Future Developments: The field of smart agriculture is full of opportunities for future developments, including the incorporation of cutting-edge technology like blockchain, robots, and sophisticated sensors, which might improve farming methods even more.
6. Socio-economic effect and Policy Development: The agricultural community will need infrastructure, training programs, and supporting policies to ensure a seamless transition as a result of the substantial socio-economic effect of smart agriculture technology adoption.
In conclusion, this analysis emphasizes the revolutionary influence of ISOA-ASVM in agriculture, stressing both the issues that require attention and their potential to completely change agricultural methods. Focusing on efficient, inclusive, and sustainable farming methods that can satisfy the world's food need while protecting the environment is essential as the industry develops.

Future research can also be directed at applying the model to data of a larger scale and multi-seasonal data in order to test the long-term adaptability in the presence of climate variability. Dynamical irrigation prediction may be enhanced with the addition of deep learning or online adaptive learning. It can also be considered in future studies if edge-computing can be deployed, and whether energy-efficient scheduling, and field trials in extreme weather conditions and sensor-failures would help further confirm scalability and realistic strength.

Funding

Study on Micro-sprinkler Irrigation Design Method Based on Water Application Quality Control Objectives and Sprinkler Hydraulic Characteristics （LGN22E090001）

Declaration:

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: Here are no have no conflicts of interest to declare.

All authors have seen and agree with the contents of the manuscript and there is no financial interest to report. We certify that the submission is original work and is not under review at any other publication.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript.

There is no human participate involved in this research. this article manuscript is created from collection of data set.

Acknowledgements : All authors contributed to the study conception and design. All authors read and approved the final manuscript.

References

[1] E. A. Abioye *et al.*, “A review on monitoring and advanced control strategies for precision irrigation,” *Computers and Electronics in Agriculture*, vol. 173, p. 105441, Jun. 2020, doi: https://doi.org/10.1016/j.compag.2020.105441

[2] I. Ara, L. Turner, M. T. Harrison, M. Monjardino, P. deVoil, and D. Rodriguez, “Application, adoption and opportunities for improving decision support systems in irrigated agriculture: A review,” *Agricultural Water Management*, vol. 257, p. 107161, Nov. 2021, doi: https://doi.org/10.1016/j.agwat.2021.107161

[3] H. Yan *et al.*, “Crop traits enabling yield gains under more frequent extreme climatic events,” *Science of*

The Total Environment, vol. 808, pp. 152170–152170, Feb. 2022, doi: https://doi.org/10.1016/j.scitotenv.2021.152170

[4] E. A. Abioye *et al.*, "Precision Irrigation Management Using Machine Learning and Digital Farming Solutions," *AgriEngineering*, vol. 4, no. 1, pp. 70–103, Mar. 2022, doi: https://doi.org/10.3390/agriengineering4010006 . Available: https://www.mdpi.com/2624-7402/4/1/6/htm#. [Accessed: May 08, 2022]

[5] K. Obaideen, "An overview of smart irrigation systems using IoT," *Energy Nexus*, vol. 7, p. 100124, Sep. 2022, doi: https://doi.org/10.1016/j.nexus.2022.100124 . Available: https://www.sciencedirect.com/science/article/pii/S2772427122000791

[6] S. Ramya, A. M. Swetha, and M. Doraipandian, "IoT Framework for Smart Irrigation using Machine Learning Technique," *Journal of Computer Science*, vol. 16, no. 3, pp. 355–363, Mar. 2020, doi: https://doi.org/10.3844/jcssp.2020.355.363

[7] A. Bhoi *et al.*, "IoT-IIRS: Internet of Things based intelligent-irrigation recommendation system using machine learning approach for efficient water usage," *PeerJ Computer Science*, vol. 7, p. e578, Jun. 2021, doi: https://doi.org/10.7717/peerj-cs.578

[8] A. Sleem and Ibrahim Elhenawy, "Smart Irrigation System with Predictive Analytics using Machine Learning and IoT," *Journal of Intelligent Systems and Internet of Things*, vol. 2, no. 2, pp. 77–83, Jan. 2021, doi: https://doi.org/10.54216/jisiot.020204

[9] H. Patel, "Artificial Intelligence and IoT based Smart Irrigation system for Precision Farming," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 4462–4467, Apr. 2021, doi: https://doi.org/10.17762/turcomat.v12i10.5184

[10] J. Angelin Blessy and A. kumar, "Smart Irrigation System Techniques using Artificial Intelligence and IoT," *IEEE Xplore*, Feb. 01, 2021. doi: https://doi.org/10.1109/ICICV50876.2021.9388444 . Available: https://ieeexplore.ieee.org/abstract/document/9388444

[11] A. H. Blasi, M. A. Abbadi, and R. Al-Huweimel, "Machine Learning Approach for an Automatic Irrigation System in Southern Jordan Valley," *Engineering, Technology & Applied Science Research*, vol. 11, no. 1, pp. 6609–6613, Feb. 2021, doi: https://doi.org/10.48084/etasr.3944

[12] None Ponugoti Kalpana, N. L. Smitha, None Dasari Madhavi, A. Nabi, N. G. Kalpana, and Sarangam Kodati, "A Smart Irrigation System Using the IoT and Advanced Machine Learning Model," *International Journal of Computational and Experimental Science and Engineering*, vol. 10, no. 4, Nov. 2024, doi: https://doi.org/10.22399/ijcesen.526

[13] P. Pandey and S. Agarwal, "A Low Cost Smart Irrigation Planning Based on Machine Learning and Internet of Things," *Current Agriculture Research Journal*, vol. 11, no. 2, pp. 568–579, Sep. 2023, doi: https://doi.org/10.12944/carj.11.2.19

[14] Saleh Mohammed Shahriar, Hasibul Islam Peyal, Md. Nahiduzzaman, and M. Abu, "An IoT-Based Real-Time Intelligent Irrigation System using Machine Learning," Oct. 2021, doi: https://doi.org/10.1109/icts52701.2021.9608813

[15] E. Bwambale, F. K. Abagale, and G. K. Anornu, "Smart irrigation monitoring and control strategies for improving water use efficiency in precision agriculture: A review," *Agricultural Water Management*, vol. 260, no. 107324, p. 107324, Feb. 2022, doi: https://doi.org/10.1016/j.agwat.2021.107324

[16] S. P. Vimal, N. Sathish Kumar, M. Kasiselvanathan, and K. B. Gurumoorthy, "Smart Irrigation System in Agriculture," *Journal of Physics: Conference Series*, vol. 1917, no. 1, p. 012028, Jun. 2021, doi: https://doi.org/10.1088/1742-6596/1917/1/012028

[17] K. Sharma and Shiv Kumar Shivandu, "Integrating Artificial Intelligence and Internet of Things (IoT) for Enhanced Crop Monitoring and Management in Precision Agriculture," *Sensors International*, pp. 100292–100292, Aug. 2024, doi: https://doi.org/10.1016/j.sintl.2024.100292

[18] L. Umutoni and V. Samadi, "Application of machine learning approaches in supporting irrigation decision making: A review," *Agricultural Water Management*, vol. 294, pp. 108710–108710, Apr. 2024, doi: https://doi.org/10.1016/j.agwat.2024.108710

[19] P. K. Kashyap, S. Kumar, A. Jaiswal, M. Prasad, and A. H. Gandomi, "Towards Precision Agriculture: IoT-Enabled Intelligent Irrigation Systems Using Deep Learning Neural Network," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17479–17491, Aug. 2021, doi: https://doi.org/10.1109/jsen.2021.3069266

[20] Y. Tace, M. Tabaa, S. Elfilali, C. Leghris, H. Bensag, and E. Renault, "Smart irrigation system based on IoT and machine learning," *Energy Reports*, vol. 8, pp. 1025–1036, Nov. 2022, doi: https://doi.org/10.1016/j.egyr.2022.07.088

[21] R. Aminuddin, A. S. Sahrom, and M. H. A. Halim, "Smart Irrigation System for Urban Gardening using Logistic Regression algorithm and Raspberry Pi," *Journal of Physics: Conference Series*, vol. 2129, no. 1, p. 012044, Dec. 2021, doi: https://doi.org/10.1088/1742-6596/2129/1/012044

[22] A. Glória, J. Cardoso, and P. Sebastião, "Sustainable Irrigation System for Farming Supported by Machine Learning and Real-Time Sensor Data," *Sensors*, vol. 21, no. 9, p. 3079, Apr. 2021, doi: https://doi.org/10.3390/s21093079

[23] Ghulam Mohyuddin, Muhammad Adnan Khan, A. Haseeb, Shahzadi Mahpara, M. Waseem, and Ahmed Mohammed Saleh, "Evaluation of Machine Learning approaches for precision Farming in Smart Agriculture System - A comprehensive Review," *IEEE access*, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/access.2024.3390581

[24] A. Kaur, Devershi Pallavi Bhatt, and L. Raja, "Developing a Hybrid Irrigation System for Smart Agriculture Using IoT Sensors and Machine Learning in Sri Ganganagar, Rajasthan," *Journal of sensors (Print)*, vol. 2024, pp. 1–15, Jan. 2024, doi: https://doi.org/10.1155/2024/6676907

[25] K. Phasinam *et al.*, "Application of IoT and Cloud Computing in Automation of Agriculture Irrigation," *Journal of Food Quality*, vol. 2022, pp. 1–8, Jan. 2022, doi: https://doi.org/10.1155/2022/8285969

[26] S. B. Patil, R. B. Kulkarni, S. S. Patil, and P. A. Kharade, "Precision Agriculture Model for Farm Irrigation using Machine Learning to Optimize Water Usage," *IOP conference series. Earth and environmental science*, vol. 1285, no. 1, pp. 012017–012017, Jan. 2024, doi: https://doi.org/10.1088/1755-1315/1285/1/012017

[27] B. Priyanka, "Machine Learning Driven Precision Agriculture: Enhancing Farm Management through Predictive Insights," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 23s, pp. 195–201, Mar. 2023, Available: https://ijisae.org/index.php/IJISAE/article/view/6705

[28] S. Shreya, S. K. R, V. L. R, and R. Madhumathi, "Agro World: A Naive Bayes based System for Providing Agriculture as a Service," May 2022, doi: https://doi.org/10.1109/iciccs53718.2022.9788290

[29] K.M.Annammal, S.Porkodi, V.Jerald Abishek, C.Thomas Abraham, and J.Dhana Babu, "Automated Irrigation for Smart Gardening based on IOT using sensors," *international journal of engineering technology and management sciences*, pp. 117–122, Sep. 2022, doi: https://doi.org/10.46647/ijetms.2022.v06i05.016

Hybrid Federated Learning Framework for APT Attack Chain Detection and Privacy Enhancement

Jie Ji[1*], Shi Qiu[2], Shengpeng Ye[1], Xin Liu[1]
[1]College of Information Engineering, Yangzhou Polytechnic Institute, Yangzhou 225127, China
[2]Yangzhou Polytechnic Institute, Yangzhou 225127, China
E-mail: jij@ypi.edu.cn
*Corresponding author

Keywords: APT attack, federated learning, privacy enhancement, adaptive differential privacy

Received: January 25, 2026

To address the issues of cross-domain data privacy protection and collaborative analysis in the detection of Advanced Persistent Threat (APT) attack chains, this paper proposes an APT attack collaborative detection and privacy enhancement method based on hybrid federated learning. This method constructs a two-layer federated learning framework of horizontal cross-organization collaboration and vertical multi-feature fusion, realizing the deep fusion of multi-source threat intelligence under the premise of privacy protection. By designing an adaptive differential privacy mechanism and dynamically optimizing the noise addition strategy, it minimizes the loss of model performance while guaranteeing data security. At the same time, this paper introduces a time-series graph neural network detection model to achieve accurate perception and correlation analysis of multi-stage behaviors of APT attacks. The experimental results demonstrate that HybridFL-APT achieves a macro-averaged F1-score of 0.914 and an attack stage identification accuracy of 0.867. Compared to the standard FedAvg algorithm, the proposed framework reduces the cumulative communication overhead by 30.9% while maintaining robust privacy protection even at a strict privacy budget (ε=0.1).

Povzetek: Članek predstavi hibridni federativni model za zaznavanje APT napadov, ki izboljša natančnost analize ob hkratnem varovanju zasebnosti podatkov.

1 Introduction

With the continuous deepening of the digitization process, the threats faced by the cyberspace are increasingly showing a trend of complexity and sophistication. Advanced Persistent Threat (APT) [1] [2] , due to its long-term concealment, clear targeting, and multi-stage attack chain, has become a major challenge in the field of cybersecurity. The traces of such attacks are usually scattered in different network regions and devices, forming cross-domain and heterogeneous "data silos", resulting in dilemmas for traditional detection methods, which require centralized data processing in terms of privacy compliance and collaborative efficiency.

As a "data remains unchanged, model changes" distributed learning paradigm, federated learning provides a new approach for cross-domain collaborative analysis while protecting data privacy. Existing research has applied it to scenarios such as malicious traffic identification. However, it is mostly limited to horizontal federated architectures or the detection of single attack types. This makes it difficult to effectively model the complex associations of multi-stage behaviors in the APT attack chain and fails to fully integrate multi-dimensional heterogeneous features such as network traffic and endpoint logs. In addition, there is a risk of privacy leakage through parameters in the federated learning process itself [3] . Current privacy protection schemes mostly adopt fixed noise addition mechanisms, which are difficult to balance detection accuracy and protection intensity in dynamic and complex network environments, limiting their practicality and reliability in high-demand scenarios such as APT detection.

To evaluate the effectiveness of the proposed framework, this paper focuses on the following three Research Questions (RQs):

- RQ1: How to effectively fuse cross-organization heterogeneous features while strictly maintaining data privacy?
- RQ2: Can the integration of a temporal-graph neural network accurately correlate multi-stage APT attack behaviors?
- RQ3: How does the adaptive differential privacy mechanism optimize the trade-off between detection utility and privacy intensity?

Therefore, this paper proposes a hybrid federated learning collaborative detection framework for the APT attack chain. By designing a two-layer learning mechanism that combines horizontal and vertical federated fusion, this framework achieves deep fusion and knowledge sharing of cross-domain heterogeneous features while protecting the data privacy of each participating party. An adaptive differential privacy mechanism is introduced to optimize the trade-off

between privacy and utility. A temporal-graph neural network model is constructed to accurately depict the dynamic evolution process and entity association relationships of the attack chain. Experimental results show that this framework exhibits considerable benefits in terms of detection accuracy, privacy protection, and communication efficiency.

2 Related research work

Regarding the core challenges in APT attack detection, such as data privacy, cross-domain collaboration, and multi-stage correlation analysis, which are raised in the introduction, this section will systematically review and comment on the existing research from three aspects: APT detection technology, secure applications of federated learning, and privacy-enhanced computing [4] [5] [6] [7] .

In the field of APT detection technology, the mainstream methods have gradually evolved from rule signature-based to machine learning-based methods. In recent years, deep learning models, with their powerful feature learning capabilities, have been widely used to identify complex attack patterns from massive logs and network traffic. However, most existing methods need a centralized data analysis architecture, that is, the raw data distributed everywhere needs to be aggregated to a central server [8] [9] [10] [11] . This mode not only faces serious privacy leakage and compliance risks due to cross-domain data transmission, but also has difficulty breaking the "data silos" formed by organizational barriers, resulting in limited detection horizons and making it difficult to construct a complete panoramic view of the attack chain.

Federated learning provides a new solution to the above problems. As a privacy-preserving distributed machine learning paradigm, federated learning allows participating parties to train models locally and share only model parameters instead of raw data, which has been verified in scenarios such as intrusion detection and malicious traffic classification [12] [15] . However, existing research on the application of federated learning in the field of network security mainly focuses on the horizontal federated scenario, that is, each participating party has the same data feature space, which cannot effectively handle the situation where different institutions may hold heterogeneous feature data in reality. In addition, current research mostly focuses on the detection of independent attack events and lacks the ability to conduct end-to-end modeling and collaborative analysis of the "attack chain" unique to APT attacks, which has strong logical and temporal correlations among multiple stages.

To ensure the privacy and security of the federated learning process, technologies such as differential privacy and homomorphic encryption have been widely introduced. Among them, differential privacy has become the mainstream choice due to its rigorous mathematical definition and low computational overhead [16] . Recent advancements in nonlinear optimal control and synchronization of chaotic systems have emphasized the importance of adaptive feedback in managing environmental uncertainties [23] [25] . This paralleled shift toward data-driven, real-time control provides a promising theoretical framework for balancing privacy and model utility in federated learning. Existing methods usually adopt a noise addition mechanism with a fixed intensity (such as a preset privacy budget ε). However, this static privacy protection strategy often leads to a dilemma where the model utility is significantly reduced or the privacy protection is insufficient when dealing with tasks that are highly sensitive to model accuracy, such as APT detection, and it is difficult to achieve an intelligent balance between privacy protection and performance in a dynamic and complex adversarial environment.

Therefore, when dealing with APT attacks, existing research still lacks a federated learning solution that can synergistically utilize cross-domain heterogeneous data features, deeply model the multi-stage attack associations, and adaptively balance the detection accuracy and privacy protection intensity. The research of this paper aims to fill this research gap. To clearly demonstrate the research gap, we summarize the comparative analysis of existing frameworks in Table 1, focusing on data heterogeneity, detection depth, and privacy intensity.

Table 1: Comparison of HybridFL-APT and state-of-the-art federated learning methods in APT detection

Method	Horizontal Collab	Vertical Fusion	Multi-stage Correlation	Privacy Protection
FedAvg	Yes	No	Low	Basic
VFL-LR	No	Yes	Low	Encryption
FedAttackChain	Yes	No	Medium	Fixed DP
HybridFL-APT (Ours)	Yes	Yes	High (T-GNN)	Adaptive DP

3 Design of hybrid federated APT detection framework

Aiming at the deficiencies of existing methods in cross-domain collaboration and multi-stage correlation analysis, this paper proposes a hybrid federated learning detection framework for APT attack chains [10] [17] [19] [20] . By constructing a two-layer learning structure of horizontal cross-organization collaboration and vertical multi-source feature fusion, and integrating a lightweight adaptive privacy protection mechanism, the framework aims to achieve efficient and accurate perception of APT attack chains under the premise of privacy protection. The framework design follows three principles: first, privacy protection is achieved by replacing the sharing of original data with the exchange of model parameters; second, multi-dimensional threat intelligence fusion is realized through the organic combination of horizontal

and vertical federations; third, the privacy protection and detection utility are dynamically balanced through an adaptive mechanism. The overall architecture is shown in Figure 1, demonstrating the interaction relationships among all participants, the coordination server, and each functional module.

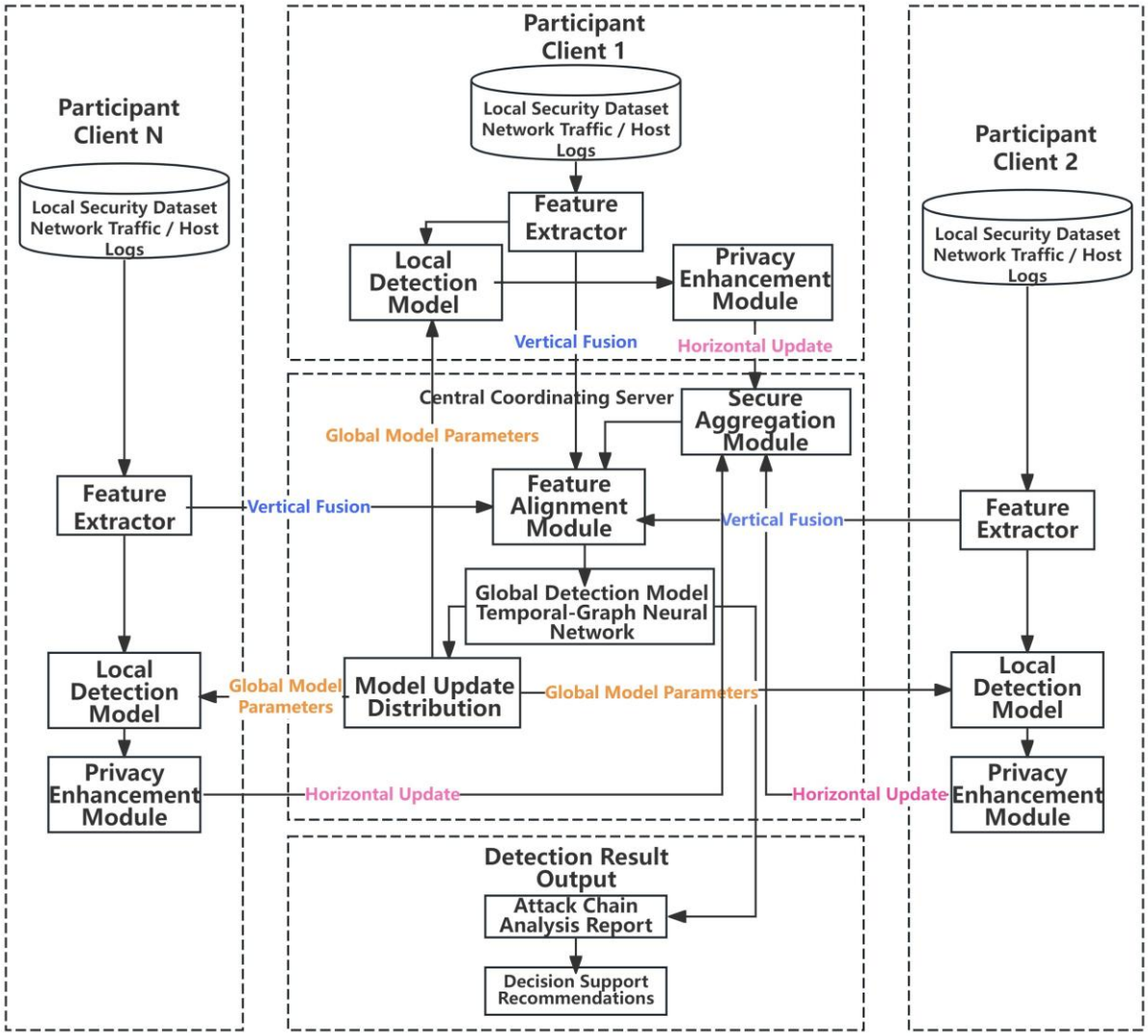


Figure 1: Overall architecture of the hybrid federated APT detection framework

3.1 Horizontal cross-organization federated collaboration mechanism

At the horizontal level, the framework supports multiple independent security domains to participate in collaborative training. Each participant C_i holds a local dataset $D_i = \{(X_j, y_j)\}_{j=1}^{n_i}$, with the same feature space but different samples, representing the observed instances in their respective network environments. Each participant maintains a copy M_i of the global detection model M_{global} locally and trains it by minimizing the local loss function $\mathcal{L}_i(\theta) = \frac{1}{n_i}\sum_{j=1}^{n_i} \ell(f(X_j;\theta), y_j)$, where θ are the model parameters, $f(\cdot;\theta)$ is the model prediction function, and $\ell(\cdot)$ is the loss function.

After the training is completed, the participating parties encrypt the update amount $\Delta\theta_i = \theta_i^{new} - \theta_i^{old}$ of the model parameters and transmit it to the central coordination server. The server aggregates the update amounts using an improved federated averaging algorithm. To mitigate the negative impact of data distribution, quality, and quantity differences on the performance of the global model, this mechanism designs a dynamic weight allocation strategy. The aggregation weight ω_i is jointly determined by the data volume n_i and the confidence factor α_i based on the local model performance evaluation. The confidence factor α_i is defined as follows:

$$\alpha_i = \frac{Acc_i^{val} - Acc_{min}^{val}}{Acc_{max}^{val} - Acc_{min}^{val}} + \beta$$

Among them, Acc_i^{val} represents the validation set accuracy of Party C_i after local training in this round. Acc_{max}^{val} and Acc_{min}^{val} are the highest and lowest validation set accuracies among all parties, respectively. β is a smoothing constant, which is used to avoid zero weights.

The final aggregation formula is as follows:

$$\theta_{global}^{t+1} = \theta_{global}^{t} + \sum_{i=1}^{N} \frac{\alpha_i \cdot n_i}{\sum_{j=1}^{N} \alpha_j \cdot n_j} \Delta\theta_i$$

3.2 Vertical multi-source feature federated fusion mechanism

At the vertical level, the framework aims to fuse heterogeneous feature views for the same set of network entities, such as traffic features from network security devices and process behavior features from endpoint detection and response systems. This fusion process starts with sample alignment under privacy protection [21] [22] . Under the unified coordination of the coordination server, each participating party uses a private set intersection protocol based on blind signature or oblivious pseudorandom function to securely calculate the set of sample identifiers I_{common} that are common to all parties without revealing the ID of their nonintersecting samples.

For the aligned common samples $i \in I_{common}$, assume there are *K* parties, and the *k*-th party holds the feature vector $x_i^{(k)}$. Each party first encrypts the local feature transformation result (e.g., through a lightweight embedding layer $g_k(\cdot)$) using a homomorphic encryption algorithm to obtain $[\![h_i^{(k)}]\!] = Enc\left(g_k\left(x_i^{(k)}\right)\right)$, and uploads the ciphertext to the server. The server then performs an aggregation operation in the ciphertext state:

$$[\![H_i]\!] = \sum_{k=1}^{K} [\![h_i^{(k)}]\!]$$

The aggregated ciphertext $[\![H_i]\!]$ can only be decrypted by the designated party with the private key to obtain the fused feature representation H_i, which is used to update the shared layer responsible for decision-making in the global model. The global model M_{global} is designed to include *K* parallel feature encoding sub-networks $\{g_k\}_{k=1}^{K}$ and a top-level fusion network $F(\cdot)$. Finally, the prediction result for the sample i is $\hat{y}_i = F(H_i)$. This design enables the model to learn the complementarity and correlation between different feature spaces end-to-end, which is of great significance for identifying the hidden patterns in the APT attack chain that only become apparent under cross-dimensional correlation analysis.

3.3 Lightweight adaptive privacy enhancement module

To defend against potential gradient leakage attacks during the federated learning process, the framework in this paper integrates a lightweight adaptive differential privacy module. The core of this module lies in the dynamic noise injection mechanism. After each round of local training, the participant first calculates the parameter update $\Delta\theta_i$, and performs norm clipping on it to limit the sensitivity. The specific operation is: $\overline{\Delta\theta_i} = \Delta\theta_i / max\left(1, \frac{\|\Delta\theta_i\|_2}{C}\right)$, where C is the clipping threshold.

The noise scale σ_i is not a fixed value, but is dynamically adjusted according to the norm of the update vector in this round and the preset privacy budget[4] [7] . The specific calculation formula is as follows:

$$\sigma_i = \frac{C \cdot \sqrt{2\ln(1.25/\delta)}}{\epsilon_{round}}$$

Among them, ϵ_{round} represents the privacy budget allocated to the current training round, while δ represents a very small slack probability. The design of the adaptive noise scale σ_i draws inspiration from robust adaptive control theories in complex time-varying systems [23] [24] [24] [25] . Similar to how neural-network-based output feedback ensures stability for MIMO delay systems [28] , our ADP mechanism treats the privacy-induced noise as a dynamic disturbance, ensuring that the model updates remain within stable convergence bounds even when data is incomplete or cross-domain heterogeneity is high [26] [27] . To optimize the consumption of the total privacy budget ϵ_{total}, we allocate the privacy budget round by round using a geometric sequence:

$$\epsilon_{round} = \epsilon_{total} \cdot \frac{\gamma^{t-1}(1-\gamma)}{1-\gamma^{T}}$$

Among them, $\gamma \in (0,1)$ is the attenuation factor and T is the total number of training rounds. This allocation method allocates relatively more budget in the initial stage of training to quickly learn the general pattern; while in the later stage, the budget is tightened to strengthen the protection of individual data. The parameter update form after adding noise is $\widetilde{\Delta\theta_i} = \overline{\Delta\theta_i} + N(0, \sigma_i^2 I)$. The entire training process satisfies $(\epsilon_{total}, \delta)$-differential privacy. To further reduce the communication overhead and the potential risk of information leakage, this module also performs Top-K sparsification on $\widetilde{\Delta\theta_i}$ before encryption and only uploads the top p% gradient values with the largest absolute values.

3.4 Temporal-graph neural network detection model oriented to the attack chain

As the core of the global detection model, this paper designs a neural network that deeply integrates temporal and structural information to specifically model the complex and multi-stage behavior patterns in the APT attack chain.

Sequential behavior modeling: For each monitored entity, the ordered sequence of security events $S = [e_1, e_2, \ldots, e_T]$ generated within a certain time window is input into a bidirectional long short-term memory network (Bi-LSTM) for modeling.

$$\overrightarrow{h_t} = LSTM\left(e_t, \overrightarrow{h_{t-1}}\right), \overleftarrow{h_t} = LSTM\left(e_t, \overleftarrow{h_{t+1}}\right)$$

The final sequential feature is represented as $h_{time} = \left[\overrightarrow{h_T}; \overleftarrow{h_1}\right]$.

Entity-Relationship Diagram Modeling: We model the entities and their interaction relationships in the cyber space as a heterogeneous graph $G = (V, \varepsilon, R)$,where V represents the set of entities, ε represents the set of edges, and R represents the set of relationship types. In this paper, we use the Relational Graph Convolutional Network (R-GCN) to learn the structural features of entities in the attack context:

$$h_v^{(l+1)} = \sigma\left(\sum_{r\in R}\sum_{u\in\mathcal{N}_v^r}\frac{1}{|\mathcal{N}_v^r|}W_r^{(l)}h_u^{(l)} + W_0^{(l)}h_v^{(l)}\right)$$

Among them, $\mathcal{N}_v^r$ represents the set of neighbors that have the relationship r with the entity v. This layer can reveal the lateral movement path of the attack and the link of data leakage.

Feature Fusion and Decision: The temporal features h_{time} are fused with the graph structure features $h_v^{(L)}$ after being propagated through L layers, and a gated fusion mechanism is used for processing:

$$g = \sigma\left(W_g\left[h_{time}; h_v^{(L)}\right] + b_g\right)$$
$$z = g \odot h_{time} + (1-g) \odot h_v^{(L)}$$

Among them, $\odot$ represents element-wise multiplication, and g is an adaptive gating vector. The fused feature z is finally input into a multi-layer perceptron for classification, and the probability distribution belonging to each attack stage or normal state is output.

4 Experimental design and results analysis

To comprehensively evaluate the effectiveness, efficiency, and privacy protection capabilities of the proposed hybrid federated APT detection framework (hereinafter referred to as HybridFL-APT), this paper designs and implements a series of comparative experiments and ablation experiments. The experiments are carried out around the following three core issues: First, whether HybridFL-APT is superior to existing baseline methods in terms of the detection accuracy of multi-stage APT attacks; Second, whether the hybrid federated architecture and the adaptive privacy mechanism can effectively balance privacy protection and model utility; Third, whether the framework can meet the actual deployment requirements in terms of communication overhead and scalability.

4.1 Experimental setup

4.1.1 Dataset and preprocessing

This experiment uses two datasets to construct a simulation environment. First, the CIC-IDS2018 dataset is selected. This dataset contains various modern attack scenarios, and the attack process has significant multi-stage characteristics. We select the "Penetration Testing" and "Botnet C&C Communication" parts from it, and map their attack steps to each stage of the ATT&CK framework, constructing a label sequence covering six stages: reconnaissance, resource development, initial access, execution, persistence, and command and control. Second, a custom enterprise log dataset is constructed. To simulate the heterogeneous characteristics in vertical federation, based on the publicly available audit log format, a set of machine system logs aligned with the timestamps of CICIDS2018 is synthesized, with features including process creation, network connection, file access, and permission change, forming a view of terminal behavior. Finally, we align the network traffic features and host log features in time, constructing a mixed dataset containing 120,000 samples, where the normal traffic accounts for 70% and the multi-stage attack traffic accounts for 30%. The dataset is divided into a training set, a validation set, and a test set in chronological order, with proportions of 60%, 20%, and 20% respectively.

4.1.2 Comparison methods and evaluation metrics

To verify the superiority of the proposed framework, we set up five groups of comparison methods: Centralized (traditional centralized training), FedAvg (the standard horizontal federated averaging algorithm), VFL-LR (vertical federated logistic regression method based on secure aggregation), FedAttackChain (federated learning method for attack detection), and the proposed HybridFL-APT. Additionally, HybridFL-APT without differential privacy (DP) was set as the control group for the ablation experiment. It is a variant with the adaptive privacy module removed.

The evaluation metrics cover three aspects. In terms of detection performance, macro-averaged precision, macro-averaged recall, macro-averaged F1 score, and weighted accuracy in attack phase recognition are adopted. Regarding the trade-off between privacy protection and utility, the differential privacy budget ε consumed by each federated method to reach the target detection performance is recorded, and the actual privacy leakage risk is quantified via the success rate of membership inference attacks.

4.1.3 Federal environmental configuration

The simulation environment consists of 1 central server and 10 participating client nodes. To simulate non-independent and identically distributed as well as feature-heterogeneous situations in reality, two data partitioning methods are adopted: Horizontal partitioning distributes samples to 10 clients in a non-equal random manner to simulate the difference in data volume. Vertical partitioning randomly designates 5 clients to only have network traffic features and the other 5 clients to only have host log features. All experiments are run on a server equipped with an NVIDIA RTX 4090 GPU and implemented using the PyTorch and PySyft frameworks.

4.2 Comparative analysis of detection performance

Table 2 compares the performance of various methods on the APT attack chain detection task. The results demonstrate that the proposed HybridFL-APT model achieves an F1-score of 0.914, significantly outperforming FedAvg and VFL-LR. This validates the superiority of the hybrid architecture in effectively integrating both horizontal and vertical information. Even when compared to the representative state-of-the-art method FedAttackChain, HybridFL-APT improves the F1-score by approximately 2.9%. Furthermore, the model's performance approaches the ideal Centralized baseline, indicating that the hybrid federated collaboration mechanism can effectively learn representations approximating the global data distribution while strictly

adhering to privacy constraints. Regarding the critical metric of attack phase identification accuracy, HybridFL-APT scores 0.867, outperforming all competitors. This highlights the effectiveness of the temporal-graph neural network in capturing the logical sequence and propagation paths of attack chains. Finally, while a variant without the privacy module produces marginal performance gains, the gap remains negligible. This suggests that the impact of the adaptive privacy mechanism on the overall model utility is maintained at a minimal level.The improvement over FedAttackChain (0.914 vs. 0.885 F1-score) marks a significant step forward for threat intelligence. FedAttackChain is limited to horizontal collaboration and cannot see patterns across different infrastructure domains. In contrast, HybridFL-APT uses a hybrid fusion approach to uncover hidden correlations in the attack chain. This method effectively bridges "data silos" while keeping raw data secure.

Table 2: Performance comparison of different methods in APT attack chain detection

Method	Precision	Recall	F1-score	Stage Accuracy
Centralized	0.923	0.898	0.910	0.854
FedAvg	0.882	0.845	0.863	0.812
VFL-LR	0.871	0.902	0.886	0.801
FedAttackChain	0.894	0.876	0.885	0.838
HybridFL-APT (Ours)	0.916	0.912	0.914	0.867
HybridFL-APT w/o DP	0.921	0.910	0.915	0.869

4.3 Privacy-utility trade-off and robustness analysis

To quantitatively evaluate the privacy protection effect of the evaluation framework and its impact on model performance, different intensities of privacy budgets ε are set, and the performance and security of each federated learning method are compared. Table 3 records the detection performance of different methods and the success rate of membership inference attacks they face under four typical privacy budget levels.

We conduct a sensitivity analysis by varying the privacy budget ε As shown in Table 3, the model exhibits high stability. When the privacy constraint is tightened from ε=5.0 to ε=0.1 (increasing noise intensity), the F1-score of HybridFL-APT only decreases from 0.912 to 0.894. This marginal 2.2% performance degradation confirms that our adaptive noise addition strategy ensures model convergence and stability under varying noise levels. This stability confirms that the system's 'flatness' is maintained across different privacy regimes, aligning with the robustness criteria observed in nonlinear optimal control applications.

Table 3: Privacy-utility trade-off of each method under different privacy budgets ε

Method	ε=0.1	ε=1.0	ε=5.0	ε=∞
	F1 Score / MIA Success Rate	F1 Score / MIA Success Rate	F1 Score / MIA Success Rate	F1 Score / MIA Success Rate
Centralized	-/-	-/-	-/-	0.910 / 35.2%
FedAvg (without DP)	-/-	-/-	-/-	0.863 / 27.8%
Fixed-DP-FL	0.819 / 3.1%	0.857 / 8.5%	0.878 / 19.4%	0.885 / 26.1%
HybridFL-APT	0.894 / 4.3%	0.905 / 7.1%	0.912 / 11.6%	0.914 / 15.7%

As can be seen from Table 3, first, for the Centralized and FedAvg methods without any privacy protection mechanisms, their models are extremely vulnerable to membership inference attacks and have a very high risk of leakage. Second, under the strong privacy constraint, the baseline method using the fixed noise mechanism can suppress the attack success rate to a relatively low level (3.1%), but the model utility is severely damaged, and the F1 score drops by about 7.5% compared to the unprotected state. In contrast, under the same strong privacy constraint ε, the adaptive privacy mechanism proposed in this paper can still maintain a high F1 score of 0.894, with a significantly smaller performance loss than the fixed mechanism, while effectively controlling the attack success rate below 5%. As the privacy constraint is relaxed, the adaptive mechanism can more efficiently convert the newly added privacy budget into an improvement in model performance and demonstrates a better utility-privacy balance at all privacy levels. This proves that the adaptive noise mechanism can intelligently allocate privacy budgets according to the training dynamics, effectively avoiding the problems of "insufficient protection" or "over protection" that may occur in the global and local stages of training for the fixed mechanism.

4.4 Analysis of communication and computational efficiency

Table 4 presents the system efficiency comparison results. While the per-round training time of HybridFL-APT (20s) is slightly higher than FedAvg due to the computational overhead of T-GNN and cross-domain alignment, the framework significantly outperforms baselines in terms of total resource consumption.

- **Comparative Analysis:** Unlike traditional horizontal FL methods that suffer from slow convergence in heterogeneous environments, HybridFL-APT reaches the convergence criterion (F1 = 0.90) in only 52 iterations, a 38.8% reduction compared to the 85 rounds required by FedAvg. (As shown in Figure 2) Like robust adaptive control, our method prioritizes quality over speed in each step. By taking the time to capture complex patterns, the system follows a more stable path and converges much earlier.
- **Deployment Scenarios and Limitations:** In practical deployment, the sub-linear growth of communication cost (as shown in Figure 5) makes our framework highly suitable for large-scale enterprise networks where bandwidth is a constraint. However, a potential limitation exists in extremely high-latency environments where the two-layer alignment process may introduce synchronization delays. Future work will investigate asynchronous updates to further mitigate this.

Table 4: System efficiency comparison (Convergence criterion: F1 score = 0.90)

Method	Comm Cost (MB)	Convergence Rounds	Time per Round (s)
FedAvg	1520	85	18
VFL-LR	980	60	22
HybridFL-APT	1050	52	20

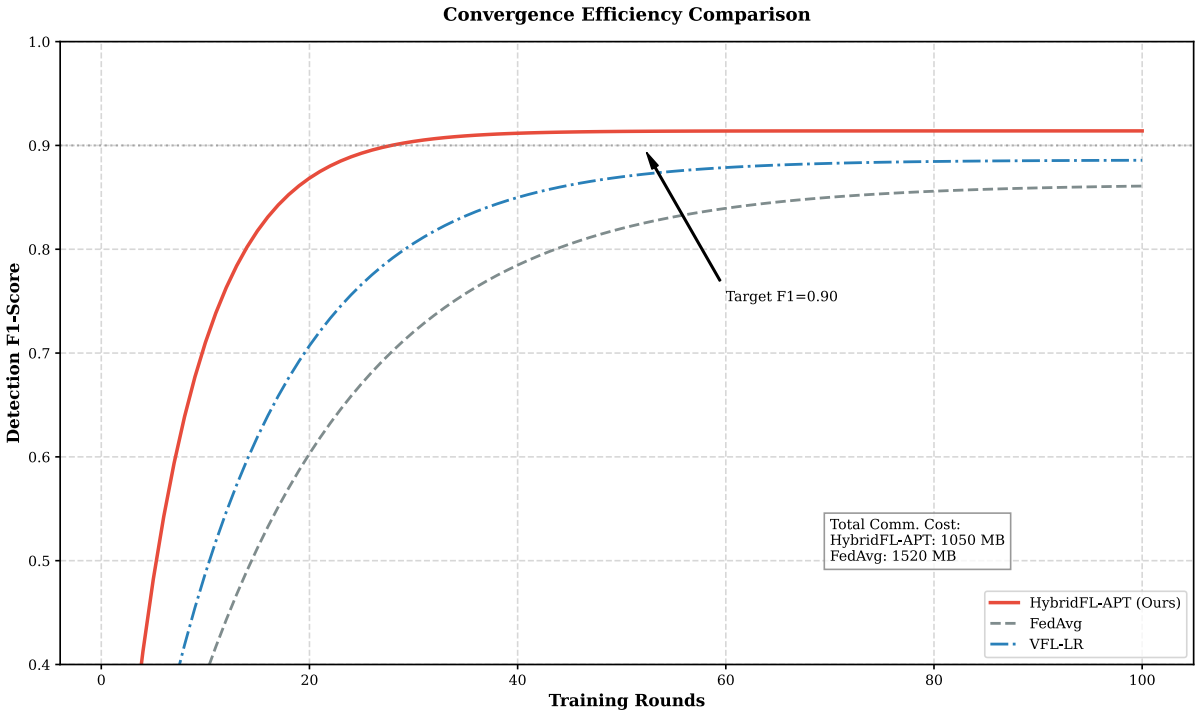


Figure 2: Convergence efficiency comparison

4.5 Ablation experiments and key parameter analysis

The ablation experiments further verified the necessity of each core component. After removing the vertical fusion, the F1 score dropped to 0.887, and the attack phase recognition accuracy dropped to 0.831. This indicates that multi-source feature fusion is crucial for understanding the complete attack chain. After removing the horizontal collaboration, the model performance was close to that of a single vertical federated method, suggesting that cross-organization knowledge sharing helps improve the generalization ability of the model. When using only the time series model, the recognition accuracy of the attack phase dropped to 0.841. When using only the graph model, it dropped to 0.819. This indicates that the two are

complementary and irreplaceable in the process of modeling the attack chain.

4.6 Supplementary data analysis and visualizations

To provide additional information about the performance and characteristics of the suggested HybridFL-APT framework, this section presents further data analyses supported by visualizations. The following figures show key trends and comparative results across detection accuracy, privacy-utility trade-offs, and system efficiency.

4.6.1 Detection performance across attack stages

Figure 3 compares the detection accuracy of HybridFL-APT at each stage. It employs two strong baselines for comparison: centralized training and the standard horizontal method FedAvg. These results are based on the six-stage attack chain from the ATT&CK framework. HybridFL-APT achieves accuracy close to the centralized upper bound at each stage. It also significantly outperforms FedAvg in the later stages, which include Command & Control and Exfiltration. Behavioral patterns in these stages are more subtle and dispersed. This indicates that our framework preserves crucial discriminating information effectively. This preservation occurs during privacy-preserving collaborative learning.

As shown in Figure 3, our model performs best in the late stages of an attack, such as C&C and Exfiltration. These stages are usually hard to detect because they look like normal admin traffic to a single organization. By using a T-GNN, our framework reaches an accuracy level close to centralized training. This proves that keeping track of timing and structural relationships during federated learning is the best way to catch coordinated, multi-stage attacks.

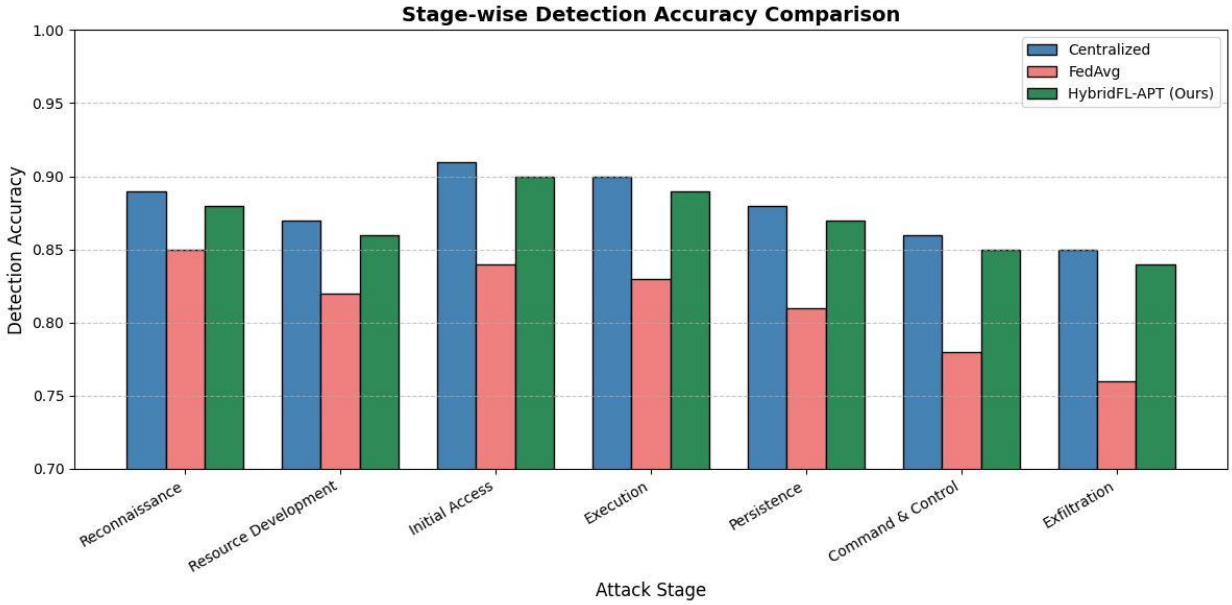


Figure 3: Detection accuracy comparison across attack stages

4.6.2 Privacy-utility trade-off under varying data heterogeneity

Figure 4 shows the variation in the privacy-utility trade-off of HybridFL-APT with varying degrees of data heterogeneity among participants. Heterogeneity is measured as the variance of local data distributions (earth mover distance). The curves indicate that when heterogeneity increases, a larger privacy budget (ε) is needed to achieve the same detection performance. However, the adaptive noise mechanism in HybridFL-APT can alleviate this effect, so the framework can maintain higher utility under strong privacy constraints ($\varepsilon \leq 1.0$) than fixed DP mechanisms.

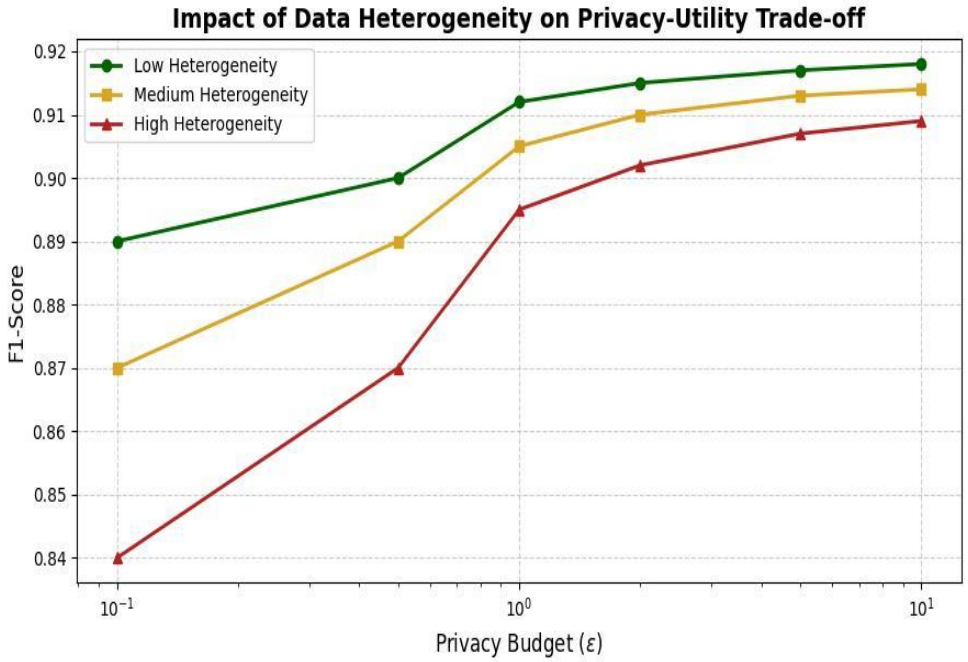


Figure 4: Impact of data heterogeneity on privacy-utility trade-off

4.6.3 Communication overhead versus number of participants

Figure 5 evaluates the scalability of HybridFL-APT by plotting the total communication overhead against an increasing number of participants (from 5 to 50). The communication cost of our hybrid architecture grows sub-linearly, which can be attributed to its selective update mechanism and gradient sparsification. In contrast, the overhead for standard FedAvg increases nearly linearly, making it less practical for large-scale deployments.

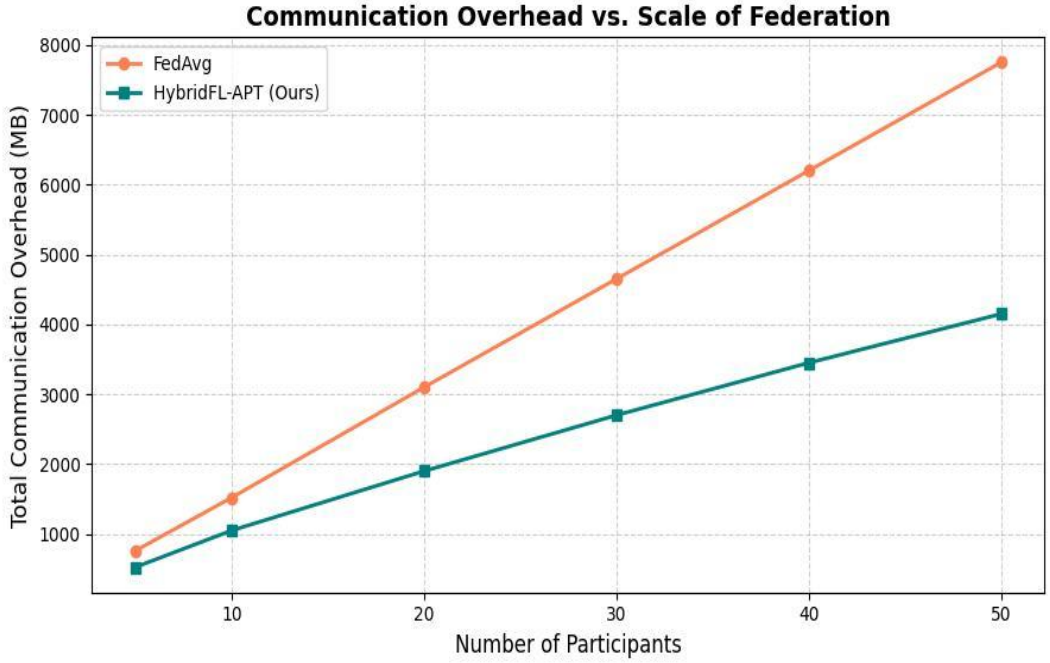


Figure 5: Communication overhead vs. federation scale

4.6.4 Ablation study: contribution of framework components

An ablation study was conducted to quantify the contribution of each major component in HybridFL-APT. The results are shown in Figure 6. Using a horizontal-only federated model as the baseline, the addition of vertical feature fusion (VFF) enhances the F1-score by 2.7%.. Integrating the adaptive privacy module (APM) leads to a minor 1.5% reduction in performance, while providing strong formal privacy guarantees. The complete model, which incorporates all components including the

temporal-graph neural network, delivers the highest overall performance.

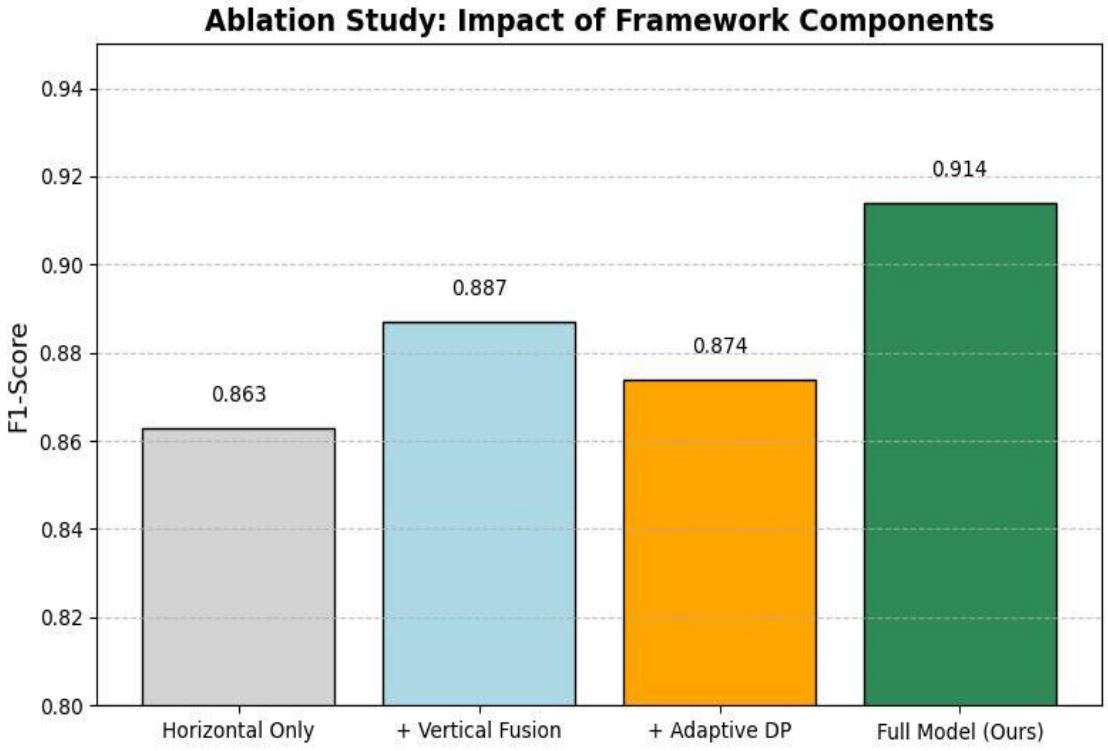


Figure 6: Ablation study: contribution of framework components

4.7 Experimental conclusion

Based on the above experimental results and analysis, the following conclusions can be drawn. First, in terms of effectiveness, HybridFL-APT achieves APT attack chain detection accuracy comparable to that of centralized methods while strictly protecting data privacy, and is significantly superior to existing federated learning methods. Second, in terms of efficiency, by adopting a hybrid federated architecture and communication optimization techniques, the framework shows obvious advantages in both communication overhead and convergence speed, and has the feasibility of practical deployment. Finally, in terms of robustness, the adaptive privacy mechanism achieves a better privacy-utility trade-off, and the framework has a certain tolerance for low-quality or malicious data. The above experimental results comprehensively verify the advancement and practicality of the framework proposed in this paper in solving the cross-domain collaborative APT detection problem.

5 Discussion

The results presented in Section 4 demonstrate that HybridFL-APT offers a substantial improvement in cross-domain APT detection. To better contextualize these findings, we discuss our framework's performance and architecture in relation to the current state of the art.

5.1 Comparison with horizontal-only frameworks

Most federated learning (FL) applications in cybersecurity, such as FedAttackChain [10] , rely on horizontal collaboration to expand training datasets. While this increases the volume of samples, it often fails to account for the heterogeneous nature of APT traces, which are typically scattered across different network layers and endpoint logs. Our results show a 2.9%–5.1% improvement in F1-score over these baselines, which we attribute to our Vertical Multi-source Feature Fusion layer. By aligning diverse feature spaces, HybridFL-APT identifies cross-domain correlations that horizontal-only models overlook, effectively bridging "data silos" at the feature level.

5.2 Adaptive privacy and model stability

A primary challenge in privacy-preserving FL [16] [18] is the "utility cliff"—a sharp drop in accuracy that occurs when fixed noise injection is used to meet strict privacy budgets. Our sensitivity analysis in Table 3 shows that HybridFL-APT avoids this drop through an Adaptive Differential Privacy (ADP) mechanism. By dynamically adjusting the privacy budget based on training dynamics, the model maintains stability and convergence even when processing noisy, cross-organizational data. This approach to maintaining model "flatness" draws on established principles from adaptive control theory [23] [24] [24] [25] [26] [27] [28] , ensuring that privacy does not come at the expense of detection reliability.

5.3 Advancements in attack chain modeling

While previous works [2] [4] rely heavily on sequence modeling, our framework introduces a Temporal-Graph Neural Network (T-GNN) to capture both temporal sequences and structural entity relationships simultaneously. As illustrated in Figure 3, this approach is particularly effective during the Command & Control and Exfiltration stages, where attacker behavior is most likely to blend in with legitimate administrative traffic. By providing a more comprehensive reconstruction of the attack chain, this method allows Security Operations Centers (SOCs) to move beyond simple alert-matching and implement more effective, context-aware response strategies.

6 Conclusion

Aiming at the cross-domain stealth characteristics of APT attacks, this paper proposes a hybrid federated learning detection framework. Through the collaborative mechanism of horizontal and vertical federation, this framework realizes the multi-source threat intelligence fusion under the premise of privacy protection. Combining adaptive differential privacy and temporal-graph neural network, while guaranteeing data security, it effectively improves the detection accuracy and interpretability of the attack chain.

Experimental results show that the proposed framework is close to the centralized baseline in detection performance, significantly outperforms existing federated learning methods, and performs excellently in terms of communication efficiency and robustness.

This study is currently mainly based on simulation data. Future work will further explore the deployment and optimization of this framework in real industrial environments, and study the combination of unsupervised learning and trusted computing technologies to enhance the ability to detect unknown threats and the auditability of collaborative processes.

References

[1] SiJung Kim, DoEun Cho, SangSoo Yeo. Secure model against APT in m-connected SCADA network. International Journal of Distributed Sensor Networks, 2014, 10. DOI: 10.1155/2014/594652

[2] Yazhou Du, Weiwu Ren, Wenjuan Li, et al. GA-ConvE: An APT attack prediction method based on combination of graph attention network and 2D convolution. Neural Networks, 2025: 108216. DOI: 10.1016/j.neunet.2025.108216

[3] Rabia Khan, Noshina Tariq, Muhammad Ashraf, et al. FL-DSFA: Securing RPL-based IoT networks against selective forwarding attacks using federated learning. Sensors, 2024, 24(17). DOI: 10.3390/s24175834

[4] Wudu Bitew Alemayew, Ketema Adere Gemeda. Federated hybrid deep learning for multi-attack detection and classification in RPL-based 6LoWPAN networks. Discover Computing, 2025, 28(1): 1-35. DOI: 10.1007/s10791-025-09852-3

[5] Rashmi Verma, Manisha Jailia. A hybrid metaheuristic federated learning approach-based attack detection system for multi-cloud environment. Journal of Cloud Computing, 2025, 14(1): 1-22. DOI: 10.1186/s13677-025-00797-y

[6] Alessio Sacco, Doriana Monaco, Guido Marchetto, Paolo Montuschi. Dealing with challenged IoT networks in hierarchical federated learning. IEEE Internet of Things Journal, 2025: 12(18):36979-36992. DOI: 10.1109/JIOT.2025.3580627

[7] Xiaoming Wang, Zhiquan Liu, Mingzhen Dai, et al. A verifiable and efficient chained federated learning scheme for privacy protection. Computer Networks, 2025: 111838. DOI: 10.1016/j.comnet.2025.111838

[8] Xiang Chen, Dun Zhang, Zhanqi Cui, et al. DP-Share: Privacy-preserving software defect prediction model sharing through differential privacy. Journal of Computer Science and Technology, 2019, 34(5): 1020-1038. DOI: 10.1007/s11390-019-1958-0

[9] Lixin Cui, Xu Wu. ALDP-FL for adaptive local differential privacy in federated learning. Scientific Reports, 2025, 15(1): 1-18. DOI: 10.1038/s41598-025-12575-6

[10] Debao Wang, Shaopeng Guan. FedFR-ADP: Adaptive differential privacy with feedback regulation for robust model performance in federated learning. Information Fusion, 2025, 116: 102796. DOI: 10.1016/j.inffus.2024.102796

[11] G Hemanth Kumar, Sivananda Lahari Reddy Elicherla, Sugandha Saxena, et al. FL-DPCSA: Federated learning with differential privacy for cache side-channel attack detection in edge-based smart grids. E-PRIME: Advances in Electrical Engineering, 2025. DOI: 10.1016/j.prime.2025.101057

[12] George Alter, Brett Hemenway Falk, Steve Lu, et al. Computing statistics from private data. Data Science Journal, 2018, 17. DOI: 10.5334/dsj-2018-031

[13] Tayyab Rehman, Noshina Tariq, Farrukh Aslam Khan, et al. FFL-IDS: A fog-enabled federated learning-based intrusion detection system to counter jamming and spoofing attacks for the Industrial internet of things. Sensors, 2024, 25(1). DOI: 10.3390/s25010010

[14] Rong Xie, Zhong Chen, Weiguo Cao, et al. Federated self-expanding neural network learning framework for heterogeneous devices. Expert Systems with Applications, 2026: 131199. DOI: 10.1016/j.eswa.2026.131199

[15] Madhuri Gadwal, Atul Negi. A lean discriminative nonlinear federated dictionary learning method. Engineering Applications of Artificial Intelligence, 2026, 167: 113862. DOI: 10.1016/j.engappai.2026.113862

[16] Haoyu Jiang, Xiaoliang Chen, Duoqian Miao, et al. PrivTSAD-FedWGAN: A novel federated learning and WGAN framework for privacy-preserving multivariate time series anomaly detection. Expert Syste

ms with Applications, 2026: 131049. DOI: 10.1016/j.eswa.2025.131049

[17] Jiayuan Chen, Tiantian Zhu, Zhengqiu Weng, et al.SLATSCOG: A secure authentication framework via federated data generation and temporally-enhanced split learning. Knowledge-Based Systems, 2026: 115304. DOI: 10.1016/j.knosys.2026.115304

[18] Jie Niu, Runqi He, Qiyao Zhou, et al. Adaptive differential privacy Cox-MLP model based on federated learning. Mathematics, 2025, 13(7). DOI: 10.3390/math13071096

[19] Sanxiu Jiao, Lecai Cai, Xinjie Wang, et al. A differential privacy federated learning scheme based on adaptive gaussian noise. CMES-Computer Modeling in Engineering & Sciences, 2024, 138(2): 1679-1694. DOI: 10.32604/cmes.2023.030512

[20] Yanjin Cheng, Wenmin Li, Sujuan Qin, et al. Differential privacy federated learning based on adaptive adjustment. CMC-Computers Materials & Continua, 2025, 82(3): 4777-4795. DOI: 10.32604/cmc.2025.060380

[21] Jing Rong, Qiuzhan Zhou, Huinan Wu. A hybrid federated learning framework for enhancing privacy and bobustness in non-intrusive load monitoring. Sensors, 2026. DOI: 10.3390/s26020443

[22] Amjad Rehman, Kamran Ahmad Awan, Amal Al-Rasheed, et al. A novel hybrid fuzzy logic and federated learning framework for enhancing cybersecurity and fraud detection in IoT-enabled metaverse transactions. Egyptian informatics Journal, 2025, 30. DOI: 10.1016/j.eij.2025.100668

[23] Boulkroune, A., Boubellouta, A., Bouzeriba, A. et al. Practical Finite-Time Fuzzy Synchronization of Chaotic Systems with Non-Integer Orders: Two Chattering-Free Approaches. J. Syst. Sci. Syst. Eng. 34, 334–359 (2025). DOI: 10.1007/s11518-024-5635-7

[24] Rigatos, G., Abbaszadeh, M., Busawon, K., Dala, L., Pomares, J., and Zouari, F. (December 6, 2023). "Flatness-Based Control in Successive Loops for Autonomous Quadrotors." ASME. J. Dyn. Sys., Meas., Control. March 2024; 146(2): 024501. DOI: 10.1115/1.4063907

[25] Rigatos, G., Siano, P., Zouari, F. et al. Nonlinear optimal control of autonomous submarines′ diving. Mar Syst Ocean Technol 15, 57–69 (2020). DOI: 10.1007/s40868-019-00070-3

[26] Rigatos, G. et al. Flatness-based control in successive loops for dual-arm robotic manipulators. 2024 IEEE Conference on Control Technology and Applications (CCTA). IEEE, (2024). DOI: 10.1109/CCTA60707.2024.10666567

[27] G. Rigatos, P. Siano, F. Zouari and S. Ademi, "A nonlinear optimal control methoc for autonomous submarines' diving," 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 2017, pp. 1061-1066, DOI: 10.1109/ISIE.2017.8001393.

[28] Zouari, F., Mahmud, M. (2026). Neural Network-Based Robust Adaptive Output Feedback Control for MIMO Time-Varying Delay Systems. In: Mahmud, M., Pillay, N., Kaiser, M.S. (eds) Applications of Artificial Intelligence and Data Science. AAIDS 2024. Communications in Computer and Information Science, vol 2601. Springer, Cham. DOI: 10.1007/978-3-031-98498-3_5

MHGR-Net: A Multimodal Heterogeneous Graph Transformer Approach for Compliance Risk Identification and Early Warning in Business-Finance-Law-Tax Domains

Yuan Yuan
School of Accountancy, Anhui Wenda University of Information Engineering, Hefei, China
E-mail: yyjzzlx@sina.com

Keywords: Business–Finance–Law–Tax (BFLT), intelligent risk management, graph neural network, knowledge graph, multi-source heterogeneous data

Received: February 4, 2026

To address the challenges associated with highly heterogeneous Business–Finance–Law–Tax (BFLT) data and increasingly complex risk propagation patterns, this study proposed a Multimodal and Heterogeneous Graph-based Risk Network (MHGR-Net). The framework consisted of three stages. First, a Universal Information Extraction (UIE) model integrated document layout information with semantic features. Second, a temporal heterogeneous knowledge graph was constructed to model dynamic risk associations. Third, a Heterogeneous Graph Transformer (HGT) was employed to learn risk path weights through a self-attention mechanism. The experimental dataset included more than 13,000 multimodal documents, such as contracts, financial statements, and judicial judgments, collected from 30 Chinese A-share listed companies. MHGR-Net achieved an Area Under the Curve (AUC) of 94.1% and an accuracy of 95.03%. The model required only 14 minutes for training. It outperformed mainstream baseline methods, including XGBoost, Graph Convolutional Network (GCN), and Temporal Graph Convolutional Network (T-GCN), in both predictive performance and computational efficiency. The proposed system reduced data silos and provided reliable decision support for enterprises shifting from passive compliance management to proactive and intelligent risk governance.

Povzetek: Študija predstavi MHGR-Net, ki z univerzalnim izluščanjem informacij iz večmodalnih BFLT dokumentov, časovno heterogenim grafom znanja in heterogenim grafnim Transformerjem modelira dinamične poti tveganja za proaktivnejše in natančnejše upravljanje podjetniških tveganj.

1 Introduction

Driven by the current wave of global digitalization, enterprises are undergoing a profound data-driven transformation. Massive volumes of "Business-Finance-Law-Tax" (BFLT) data—including business contracts, financial transactions, legal documents, and tax vouchers—are being generated and accumulated at an unprecedented pace [1]. These data sources are inherently heterogeneous: formats vary widely, content is fragmented, and information is often siloed, making it difficult for enterprises to develop a unified, comprehensive understanding of risk [2]. Meanwhile, the global regulatory environment is becoming increasingly stringent, compliance requirements are continually rising, and both the complexity and propagation of risks are intensifying. Even minor flaws in a single domain can cascade through intricate business processes, triggering cross-domain compliance crises.

Traditional risk management approaches are proving increasingly inadequate under these conditions. Heavy reliance on manual audits, department-specific reviews, and rule-based judgments results in limited coverage, delayed responses, and fragmented insights [3]. These methods struggle to penetrate surface-level operations and cannot effectively identify hidden risks embedded in multi-party transactions, complex equity structures, or ambiguous contract clauses. For example, a related-party transaction may appear compliant but could be used to conceal fund misappropriation or profit shifting. The true risk might only surface during a tax audit or legal dispute. By that time, the enterprise could already be at a disadvantage [4]. Therefore, breaking data silos and adopting a proactive, intelligent paradigm capable of detecting and alerting cross-domain BFLT risks has become essential for enterprise survival and sustainable development.

To address the aforementioned complex challenges, this study proposed an end-to-end intelligent system named Multimodal and Heterogeneous Graph-based Risk Network (MHGR-Net). The system was designed to enable precise identification and dynamic early warning of BFLT compliance risks. The primary objective was to eliminate data silos through technological innovation and to address three key research questions from both theoretical and empirical perspectives.

a. Multi-source integration: How can a multimodal Universal Information Extraction (UIE) model effectively integrate highly heterogeneous data sources, including contracts, financial statements, and legal documents?

b. Temporal evolution: How can a temporal heterogeneous knowledge graph be constructed to capture

dynamic risk propagation patterns across cross-domain business processes?

c. Decision robustness: Based on the heterogeneous graph transformation architecture of MHGR-Net, can the model improve the accuracy and robustness of complex risk path identification while maintaining computational efficiency?

The technical framework comprised three core components.

- First, a domain-adaptive multimodal UIE model was developed. The model incorporated industry-specific knowledge and document layout features. It transformed fragmented unstructured data, such as scanned contracts, invoices, and financial reports, into structured knowledge triples with high efficiency.
- Second, a dynamic temporal heterogeneous BFLT knowledge graph was constructed. Timestamp annotations and entity alignment algorithms were introduced. The graph represented complex enterprise relationship networks and described the temporal evolution of risk states.
- Third, the MHGR-Net deep learning model was implemented as the central analytical engine. The model drew on the architectural advantages of the Heterogeneous Graph Transformer (HGT). A self-attention mechanism was applied to learn the weights of different risk propagation paths. Compared with traditional rule-based methods and static graph models, MHGR-Net identified latent high-risk entities more accurately and predicted potential risk outbreak points more effectively. This capability supported the transition from passive compliance management to proactive and intelligent risk governance.

2 Related work

2.1 Current research on enterprise compliance risk management

Enterprise Risk Management (ERM) is an integrated framework designed to help organizations identify, assess, and respond to various risks. Recent studies have emphasized that ERM should not be treated as an isolated control mechanism. Instead, it should be deeply embedded within corporate governance structures. Gleißner and Berger [5] argued that strengthening embedded risk management was essential to improving ERM effectiveness. Tewu et al. [6] confirmed the significant positive impact of compliance practices on corporate performance in the banking sector. For start-ups, Uwamusi [7] examined strategies for optimizing legal structures under complex regulatory frameworks in order to reduce tax burdens and operational risks. Li et al. [8] conducted a systematic review of the application of Knowledge Graphs (KG) in ERM. With the rapid development of the digital economy, data-driven risk identification has become a major research focus. Li et al. [9] applied an optimized Backpropagation neural network (BPNN) to predict financial risks in listed companies. Tamascelli et al. [10] used neural networks to predict equipment failure time under specific industrial risk events. Chen [11] proposed a Deep Learning-based model for corporate financial risk prediction and intelligent early warning.

Although these studies established the importance of compliance management, existing risk prediction models remain inadequate for handling cross-domain risks in the BFLT context. Most approaches treat risk factors as independent feature variables. They fail to capture cross-domain risk propagation across multi-party transactions and complex ownership structures. In addition, many models rely primarily on single-domain financial data. As a result, they struggle to detect complex anomalies in multinational financial statements [12]. This limitation weakens their early warning capability when cross-domain chain reactions occur.

2.2 Application of knowledge graphs in financial and legal risk control

In the financial and legal domains, Knowledge Graphs have demonstrated effectiveness in fraud detection, anti-money laundering, and related-party transaction analysis. Wang et al. [13] combined Graph Convolutional Network (GCN) with knowledge graphs for stock price prediction. Shi et al. [14] constructed a knowledge fusion graph to forecast market trends. To address the integration of multi-source heterogeneous data, Yan et al. [15] proposed a practical framework for building enterprise knowledge graphs, which alleviated the data silo problem. The performance of knowledge graphs depends heavily on the quality of Information Extraction (IE). Wang and El-Gohary [16] and Zhang [17] explored Artificial Intelligence (AI)-based frameworks for tax and construction regulation IE. Shim et al. [18] introduced Large Language Models (LLMs) to process unstructured documents.

Despite these advances, State-of-the-Art (SOTA) techniques still exhibit clear limitations in semantic integration within the BFLT domain. Existing IE models often fail to simultaneously incorporate document layout features and cross-domain professional terminology. This semantic gap weakens the logical depth of the constructed knowledge graphs. Consequently, such graphs struggle to support complex compliance scenarios that require joint reasoning over legal clauses and financial data.

2.3 Graph neural networks and their application in risk analysis

Graph Neural Networks (GNNs) learn node representations by aggregating information from neighboring nodes. This mechanism aligns closely with the way risks propagate within networked systems. Wang [19] explored a financial risk identification approach based on Convolutional Neural Networks (CNNs). Mojdehi et al. [20] proposed a hybrid model that integrated topological analysis with GNNs, which improved credit risk assessment in supply chain finance.

Although GNNs possess inherent advantages in modeling non-Euclidean structured data, existing studies

primarily focus on empirical metrics such as Accuracy and Area Under the Curve (AUC). They provide limited theoretical guarantees of stability. From the perspective of adaptive control, a robust early warning system should ensure finite-time convergence and resilience to uncertainty propagation. Most GNN-based models lack stability guarantees when learning path weights from dynamically evolving BFLT data. The proposed MHGR-Net addresses this limitation. It draws on principles from optimal control theory. The model incorporates temporal encoding and adaptive heterogeneous weight adjustment. This design aims to strengthen dynamic risk modeling and improve theoretical robustness.

Through a comprehensive and critical review of the above literature, Table 1 summarizes representative studies in enterprise compliance risk identification. It compares their core methodologies, application domains, advantages, and technical limitations in handling cross-domain BFLT data integration. The comparison clarifies the gaps in current SOTA research and establishes the technical positioning of this study.

Table 1: representative studies in enterprise compliance risk identification and their technical limitations

Author/Year	Core Method	Research Domain	Main Strength	Critical Limitation (SOTA Gaps)
Gleißner et al. [5]	Embedded risk framework	Corporate management	Emphasized integration of ERM and governance	Lacked concrete quantitative algorithmic implementation
Tewu et al. [6]	Empirical analysis	Banking	Verified positive correlation between compliance and performance	Limited ability to process unstructured textual risks
Uwamusi [7]	Regulatory navigation framework	Start-ups	Optimized legal and tax structures	Focused on policy guidance rather than automated identification
Li et al. [8]	Knowledge Graph review	Risk management	Systematically summarized Knowledge Graphs applications	Did not propose a dedicated BFLT architecture
Li et al. [9]	Optimized BPNN	Financial forecasting	Improved single-domain prediction accuracy	Treated risk factors as independent variables
Tamascelli [10]	Neural network failure analysis	Industrial safety	Applied AI to time-to-failure prediction	Not generalizable to complex business logic
Chen [11]	Deep learning early warning model	Financial risk	Strengthened automated financial risk prediction	Lacked cross-domain semantic association analysis
Li et al. [12]	Anomaly pattern recognition	Multinational enterprises	Detected fraud patterns in financial statements	Failed to reveal underlying legal roots of risk
Wang et al. [13]	Knowledge Graphs + GCN	Stock market	Captured relationships among market entities	Ignored temporal dynamics of risk evolution
Shi et al. [14]	GCN-LSTM	Market forecasting	Introduced preliminary temporal modeling	Limited capability in unstructured document extraction
Yan et al. [15]	Heterogeneous Knowledge Graphs construction framework	Enterprise knowledge base	Addressed heterogeneous data silos	Insufficient reasoning depth for compliance risk
Wang et al. [16]	Deep learning-based relation extraction	Construction safety	Enabled automated regulation transformation	Unable to process cross-domain knowledge such as taxation
Zhang [17]	Multimodal AI framework	Tax compliance	Improved tax data integration	Did not integrate financial–legal cross-validation
Shim et al. [18]	LLM + OmEGa	Manufacturing	Strong semantic understanding with low zero-shot cost	Lacked domain adaptation for financial and legal contexts
Wang [19]	CNN-based risk model	Financial risk	High computational efficiency	Not suitable for complex non-Euclidean graph structures
Mojdehi et al. [20]	Topological analysis + GNN	Supply chain finance	Enhanced capture of topological features	Lacked theoretical convergence guarantees
This Study	MHGR-Net	BFLT cross-domain compliance	Unified multimodality, dynamic modeling, and theoretical robustness	Filled SOTA gaps in cross-domain integration and temporal risk evolution

3 Construction of the BFLT knowledge graph and the risk identification & early warning model

3.1 System architecture and technical analysis

To enable intelligent identification and early warning of compliance risks in the BFLT domain, the system adopts a hierarchical technical architecture. This architecture consists of three layers: the Data Layer, the Model Layer, and the Application Layer. It is designed to provide end-to-end processing, from heterogeneous data ingestion to risk decision output. To visually illustrate this processing pipeline, the study presents a system workflow in the form of pseudocode and a logic path diagram (Figure 1). These depict the entire process, from multimodal data extraction, through temporal knowledge graph construction, to risk prediction using a HGT. The hierarchical design ensures modularity and scalability, allowing the system to handle complex business logic efficiently.

```
Start
Input: Extract text, tables, and image-based layout information
Output: The risk rating of each risk subject and the warning results of potential
        high-risk associations
def extract_knowledge(inputs, uie_model):
    # Perform multimodal information extraction
    entities, relations = uie_model.extract(inputs)
    return entities, relations
def construct_graph(entities, relations):
    # Build a heterogeneous temporal knowledge graph
    G = Graph()
    for e in entities:
        G.add_node(e)
    for r in relations:
        G.add_edge(r.head, r.tail, r.type, r.time)
    return G
def risk_prediction(G, htgn_model):
    # Perform risk identification using the Heterogeneous Temporal Graph Network
    embeddings = htgn_model.encode(G)
    risk_scores = htgn_model.predict(embeddings)
    return risk_scores
def active_feedback(risks, model, threshold=0.7):
    # Perform active learning for self-optimization
    low_conf = [n for n, s in risks.items() if s < threshold]
    for node in low_conf:
        new_data = retrieve_data(node)
        model.retrain(new_data)
End
```

Figure 1: Pseudocode flow of the MHGR-Net-based risk intelligent identification and early warning model.

The system's end-to-end processing logic is presented in the pseudocode of Figure 1. The workflow comprises five core stages. First, multimodal data—including text, tables, and document layout information—is ingested. Second, the UIE model is applied to perform automated knowledge extraction. Third, a heterogeneous knowledge graph with temporal dimensions is constructed. Fourth, the core MHGR-Net is used to perform risk identification and score prediction. Finally, an active feedback mechanism triggers human verification for low-confidence results and supports incremental model retraining. This closed-loop design ensures continuous optimization when processing complex BFLT data.

The model layer serves as the core of the system, hosting intelligent processing capabilities such as IE, knowledge graph construction, and risk analysis, and providing the foundation for knowledge inference, predictive analytics, and decision support. The application layer acts as the interface and feedback channel, visualizing results in a graphical and interactive manner, while supporting human-in-the-loop and closed-loop feedback mechanisms to dynamically optimize and continuously evolve the model based on new data and insights. The architecture emphasizes cross-domain semantic integration, multimodal collaborative modeling, and sustainable learning, ensuring robust adaptability in complex risk scenarios. By leveraging these three layers in a coordinated manner, the system can efficiently detect emerging risks, monitor evolving compliance patterns, and support proactive risk mitigation strategies. The structure of the BFLT risk identification and early warning system based on MHGR-Net (Multimodal + Heterogeneous Graph-based Risk Network) is illustrated in Figure 2.

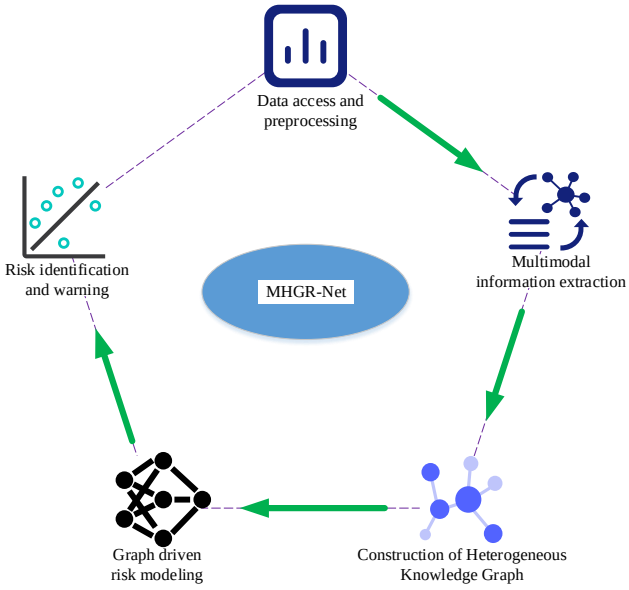


Figure 2: Illustration of the MHGR-Net-based BFLT risk identification and early warning model architecture.

Given that BFLT data involve sensitive corporate financial information and legal privacy, the system was designed in strict accordance with data ethics and security standards. During the data ingestion stage, all non-public contract details, financial statements, and tax records were de-identified, and differential privacy techniques were applied to protect sensitive attributes of specific entities. Moreover, system operations fully comply with the Data Security Law and relevant industry regulations, ensuring that data are used exclusively for research and management purposes related to compliance risk prediction. By implementing a controlled access framework, the system breaks down data silos while effectively preventing data leakage, thereby guaranteeing the compliant delivery of intelligent early-warning capabilities.

3.2 Knowledge graph construction based on multimodal IE

For the BFLT domain, a dedicated knowledge graph schema was designed, clearly defining entity types (e.g., enterprises, financial indicators, tax provisions) and relation types (e.g., "signs contract," "pays tax"). Core IE relies on a domain-adaptive multimodal UIE model [21,22]. By performing secondary pretraining on domain-specific corpora, the model significantly enhances its understanding of professional terminology and complex document structures.

When processing semi-structured and unstructured documents (e.g., PDF contracts or scanned invoices), the model uses its layout-aware capabilities to combine text, tables, and layout information. This enables accurate extraction of multimodal knowledge elements, forming triples that represent entities, their attributes, and the relationships between them.

To achieve cross-domain semantic integration, the system formalizes the input multimodal document D as a heterogeneous feature set comprising text streams, image pixels, and layout coordinates. Let D consist of N document elements:

$$D = \{d_1, d_2, \ldots, d_N\} \tag{1}$$

For each element, the system employs a deep encoder $f_{UIE}(\cdot)$ based on the LayoutLMv2 architecture. This encoder jointly models spatial layout and textual semantics via a multi-head self-attention mechanism, producing a multimodal feature vector X_i:

$$X_i = f_{UIE}(d_i), \quad i = 1, 2, \ldots, N \tag{2}$$

Based on these representations, the knowledge extraction module performs named entity recognition (NER) and relation extraction (RE) to construct a set of knowledge triples τ. To capture the temporal evolution of risks, the triples are extended into a time-aware data structure:

$$\tau = \{(h_j, r_j, t_j, \Delta T_j) \mid h_j, t_j \in E, r_j \in R\} \tag{3}$$

h_j, t_j denotes the head and tail entities, r_j represents the relation type between them, E and Z are the sets of all entity and relation types, respectively. E corresponds to the BFLT entity set, Z represents the set of relation types, and ΔT_j denotes the timestamp of the event.

After obtaining a large collection of knowledge triples, a temporal heterogeneous knowledge graph is constructed in the next step. The construction workflow of the knowledge graph is illustrated in Figure 3.

During the multi-source data fusion phase, to address inconsistencies in how the same entity is represented across different sources, the system employs a dual-path similarity evaluation model. The entity alignment function, $sim(e_a, e_b)$, is defined as:

$$sim(e_a, e_b) = \alpha \cdot sim_{attr}(e_a, e_b) + (1-\alpha) \cdot sim_{rel}(e_a, e_b) \tag{4}$$

Specifically, the attribute similarity sim_{attr} is computed using cosine similarity between entity embedding vectors, while the relation similarity sim_{rel} is calculated using the Jaccard index to evaluate the proportion of shared neighboring nodes between two entities. In this study, the weight coefficient α was set to 0.6, and the fusion decision threshold was set at 0.85. When the similarity score exceeds this threshold, entities are automatically merged. For scores in the range [0.7,0.85], the entities are routed to a manual verification pipeline for final approval, ensuring the foundational quality of the knowledge graph.

Next, timestamp annotation is performed. To capture the temporal evolution of business processes and risk events, timestamps τ are added to entity relationships, forming triples with temporal attributes (h_j, r_j, t_j, τ_j), where τ_j indicates the specific time the relation r_j occurs.

Finally, the constructed temporal heterogeneous knowledge graph is stored in a graph database to enable efficient storage and querying, supporting subsequent graph-based risk identification and early warning computations.

Through this process of multimodal IE and temporal graph construction, the system effectively integrates multi-source heterogeneous BFLT data. It builds a high-quality, dynamically evolving knowledge network that provides a solid data foundation for subsequent risk identification models.

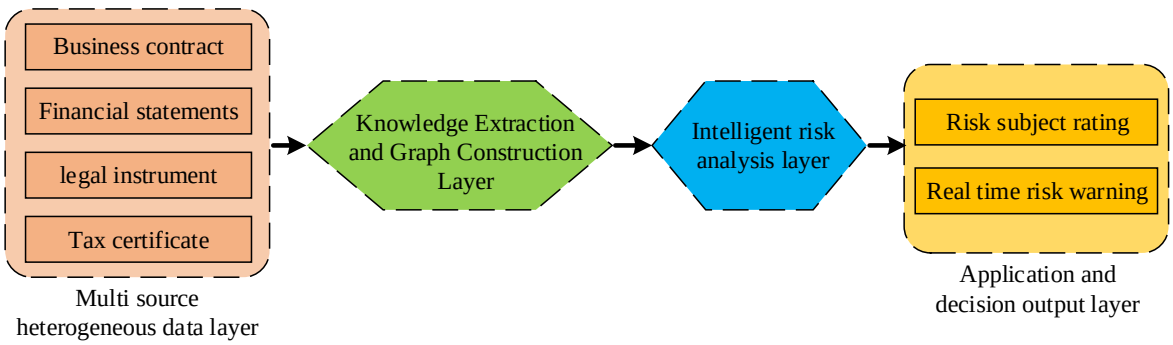


Figure 3: Illustration of knowledge graph construction steps

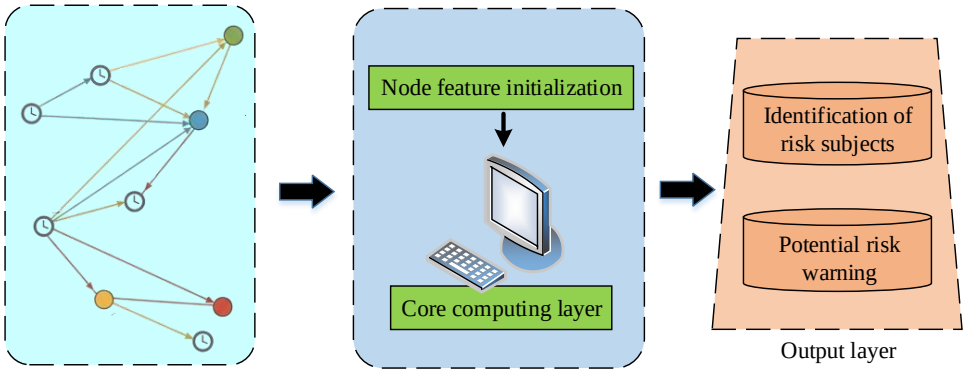


Figure 4: Schematic of the risk identification and early warning module based on heterogeneous temporal graph networks.

3.3 Risk identification and early warning module based on heterogeneous temporal graph networks

With the multi-source heterogeneous knowledge graph constructed, accurately identifying enterprise compliance risks and predicting potential risks becomes a key system challenge. The constructed knowledge graph is formally defined as a temporal heterogeneous graph G, as shown in Equation (5):

$$G=(V,E,T,\Phi) \tag{5}$$

Where V is the set of nodes, encompassing multiple entity types (e.g., enterprises, legal representatives, contracts); E is the set of typed edges, representing diverse relationships; T is the set of corresponding timestamps, indicating the occurrence time of node or edge events; and Φ is a type mapping function defining node and edge categories. Along the temporal dimension, the graph exhibits dynamic evolution, as risk states may change over time. To model this, the design incorporates a time encoding mechanism TE (·), integrating event time information into node representations to strengthen temporal dependency modeling [23].

The core model is a heterogeneous GNNs based on graph attention mechanisms, applied to risk identification, as illustrated in Figure 4.

In this network, the first step is node feature initialization. Each node $v_i \in V$ is initialized with representation $h_i^{(0)}$, which integrates both entity attributes and text-encoded information.

Subsequently, in the multi-head heterogeneous graph attention layer, attention weights are calculated for each node type $t \in T_V$ and edge type $r \in T_E$, as defined in Equation (6):

$$\alpha_{ij}^{r}=\frac{\exp\left(\text{LeakyReLU}\left(a_r^{\top}\left[W_r h_i \| W_r h_j\right]\right)\right)}{\sum_{k\in N_i^r}\exp\left(\text{LeakyReLU}\left(a_r^{\top}\left[W_r h_i \| W_r h_k\right]\right)\right)} \tag{6}$$

Where h_i, h_j denote the node representation; $\|$ indicates vector concatenation; W_r is the linear transformation matrix corresponding to edge type r; a_r is the attention vector parameter for edge type r; and N_i^r represents the neighbor set of node i under relation r.

Next, temporal encoding fusion is introduced. For each edge timestamp τ_{ij}, a positional encoding $TE(\cdot)$ is incorporated into the computation, as shown in Equation (7):

$$\tilde{h}_i=\sigma\left(\sum_{r\in R}\sum_{j\in N_i^r}\alpha_{ij}^{r}\cdot\left(W_r h_j+TE\left(\tau_{ij}\right)\right)\right) \tag{7}$$

where σ is a nonlinear activation function. Through multi-layer iterations, the model captures complex node relationships and temporal evolution patterns. In the specific implementation of Equation (7), this study employs a Sinusoidal Time Encoding (STE) mechanism to construct TE(·). This approach is inspired by the positional encoding used in the Transformer architecture. It maps continuous time intervals into 128-dimensional embedding vectors using sine and cosine functions at different frequencies. This design not only captures the temporal interval characteristics between business activities but also enables the model to adaptively learn both the periodic and abrupt patterns of risk evolution. By adding this time vector to the node feature representations, MHGR-Net can effectively distinguish between "historical stale risks" and "real-time compliance threats", thereby significantly enhancing the modeling of temporal dependencies.

Finally, the system performs risk entity identification. Framed as a node classification task, the model classifies enterprise nodes into multiple risk categories, outputting the risk level $y_i \in \{0,1,...,C-1\}$. The loss function is defined as a cross-entropy loss L_{risk} in Equation (8):

$$L_{risk}=-\sum_{i\in V_{ent}}\sum_{c=0}^{C-1}\mathbb{1}[y_i=c]\log\hat{p}_{i,c} \tag{8}$$

Where V_{ent} is the set of enterprise nodes; y_i is the true risk label of enterprise i; and $\hat{p}_{i,c}$ denotes the predicted probability that enterprise i belongs to risk level c. In the node classification task, this study categorizes

the risk level y_i into four grades according to regulatory standards and business logic: L0 (Robust, no compliance concerns), L1 (Attention, minor management gaps), L2 (Warning, significant signs of noncompliance), and L3 (Critical, major legal or tax risks triggered). Given the extreme scarcity of L2 and L3 samples in real-world scenarios (resulting in a long-tail data distribution), to prevent the model from overfitting to the high-frequency L0 class, a Weighted Cross-Entropy (WCE) mechanism was incorporated into the logistic regression. By dynamically adjusting the loss weights based on the inverse of class sample sizes, the model is forced to assign greater attention to the high-risk minority classes, thereby ensuring heightened sensitivity for detecting extreme compliance risks.

For potential risk early warning, formulated as a link prediction task, the model predicts whether potential high-risk edges exist in the graph. The binary cross-entropy loss L_{link} is expressed in Equation (9):

$$L_{link} = -\sum_{(i,j)\in E^+} \log\sigma(s_{ij}) - \sum_{(i,j)\in E^-} \log\left(1-\sigma(s_{ij})\right) \quad (9)$$

Where E^+, E^- are sets of positive and negative sample edges, respectively; s_{ij} is the scoring function for edge (i, j), typically defined as the dot product of node embeddings or the output of an MLP; and $\sigma(\cdot)$ is the Sigmoid activation function. For the potential risk early-warning task, which is formulated as a link prediction problem, the system constructs the training set using a Negative Sampling strategy, with a positive-to-negative sample ratio of 1:5. In terms of performance evaluation, besides the conventional ROC-AUC metric, this study specifically incorporates Precision@K to measure the practical accuracy of high-risk predictions. During model deployment, a risk-trigger threshold of 0.75 is set. Once a predicted score exceeds this threshold, the system automatically identifies the most contributive risk propagation paths via an attention mechanism. This design of "explainable paths" provides compliance officers with intuitive audit-trail evidence and supports a human-in-the-loop decision-making process, enabling enterprises to take precise interventions based on the causal insights generated by the model.

The overall loss combines both tasks, as given in Equation (10):

$$L = \gamma L_{risk} + (1-\gamma) L_{link} \quad (10)$$

Where $\gamma \in [0,1]$ is the task weight coefficient.

To meet the strict compliance audit requirements for traceability and interpretability of predictions, MHGR-Net does not operate as a black-box model. The system integrates explainable AI (XAI) tools, such as GNNExplainer, which automatically identify and extract the subgraph structures and feature paths that contribute most to a specific risk score. For instance, when the system detects a potential tax-evasion risk in a cross-border related-party transaction, it highlights the relevant contract clauses, cash-flow nodes, and temporal evolution cues. This causal traceability capability provides actionable insights for management and generates "audit-trail" evidence, ensuring that the early-warning results are admissible in legal and tax practice. Consequently, it addresses the critical challenge of trustworthiness in deploying deep learning models in high-stakes compliance scenarios.

3.4 Experimental evaluation

To comprehensively validate the effectiveness of the MHGR-Net model in cross-domain BFLT compliance scenarios, this study constructed an industry-representative experimental dataset. The dataset covers 30 A-share main-board listed companies in China, spanning the years 2020 to 2024. Data were collected and integrated from three primary sources: the China Judgments Online (https://wenshu.court.gov.cn/), the WIND financial terminal (https://www.wind.com.cn/), and publicly disclosed corporate information. The dataset comprises 1,240 legal judgments, 450 annual financial reports, 3,200 commercial contracts, and over 8,500 tax vouchers. Table 2 summarizes the statistical characteristics of the dataset. After multimodal IE and semantic alignment, the resulting heterogeneous knowledge graph contains approximately 45,120 entity nodes and 128,450 relational edges. The dataset exhibits a long-tail distribution consistent with real-world business logic. High-risk warning cases (levels L2 and L3) account for roughly 12.4%, providing a realistic and stringent environment for testing the model's ability to handle class imbalance.

Table 2: Statistics of the experimental dataset

Statistical Metric	Value	Notes
Number of listed companies	30	A-share main-board enterprises
Total number of documents	13,390	Contracts, financial reports, judgments, tax vouchers
Total knowledge graph nodes	45,120	Entities including companies, executives, indicators, clauses
Total knowledge graph edges	128,450	Heterogeneous edges with temporal annotations
High-risk cases (L2–L3)	12.40%	Reflects class imbalance

Table 3: Hardware/software configurations and key training parameters.

Category	Parameter	Value
Hardware	GPU	NVIDIA A100 80GB
	CPU	Intel Xeon Gold 6226R
	Memory	256 GB
Software	OS	Windows 11 Pro 64-bit
	Deep learning framework	PyTorch 2.0
	Graph database	Neo4j 5.x

Model params	UIE pretrained model	LayoutLMv2 (finetuned)
	HTGN layers (L)	3
	Attention heads (H)	4
	Learning rate	1e-4
	Batch size	32
	Dropout	0.2
	Epochs	120

Regarding ethics and privacy protection, this study strictly adhered to the Data Security Law and relevant industry regulations. Although the data were primarily sourced from publicly available channels, all original documents ingested by the model were de-identified and deeply anonymized, removing unnecessary personal identifiers and sensitive commercial clauses. This processing approach ensures the transparency and reliability of the research findings while maximizing the security and compliance of data usage. The experiments were conducted on a high-performance GPU server. The detailed hardware/software configurations and key training parameters are summarized in Table 3.

For performance evaluation, the MHGR-Net model was compared against XGBoost [24], GCN [25], Graph Attention Network (GAT) [26], Temporal Graph Convolutional Network (T-GCN) [27], and the model proposed by Mojdehi et al. [20]. Model performance was assessed using Accuracy, Area Under the Receiver Operating Characteristic Curve (AUC), Precision, and F1-score, which collectively reflect the effectiveness of risk identification and early warning.

4 Results and discussion

4.1 Analysis of risk identification performance across different algorithms

The proposed model was compared with baseline algorithms in terms of AUC and training time, as shown in Figure 5.

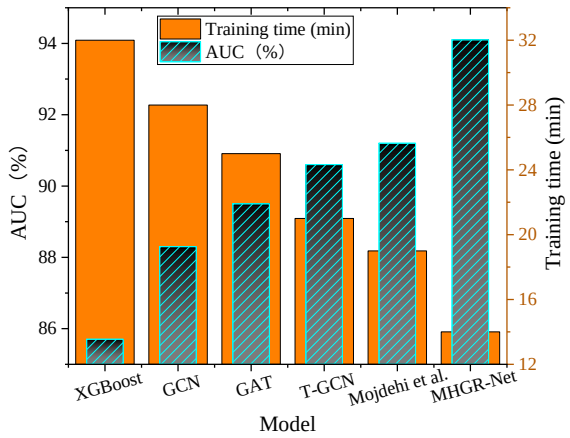


Figure 5: Comparative results of risk identification performance across different algorithms

As shown in Figure 5, the experimental results revealed notable performance differences among the evaluated models, particularly in terms of the Area Under the ROC Curve (AUC) and training time. From the perspective of the AUC metric, the proposed MHGR-Net achieved the highest accuracy in risk identification, reaching 94.1%, which was substantially superior to the benchmark models. In comparison, the classic machine learning model XGBoost achieved only 85.7%, representing the lowest performance among the tested approaches. Graph-based models demonstrated moderate improvements: GCN and GAT achieved 88.3% and 89.5%, respectively, indicating the benefits of incorporating graph structure modeling. The temporal graph model T-GCN further increased the AUC to 90.6%, while the enhanced model proposed by Mojdehi et al. [20] reached 91.2%. Overall, MHGR-Net's integration of multimodal information and heterogeneous graph structures allowed it to capture complex interdependencies across business, finance, legal, and tax data, thereby demonstrating superior accuracy, robustness, and generalization ability.

In terms of training efficiency, MHGR-Net also showed remarkable advantages. Its average training time was only 14 minutes, considerably shorter than the 32 minutes required for XGBoost and 28 minutes for GCN. This improvement indicates that the architectural and inference optimizations of MHGR-Net not only enhanced recognition performance but also significantly reduced computational costs. Consequently, the model is highly suitable for real-world deployment scenarios, particularly in contexts where rapid decision-making and limited computational resources are critical. These results highlight MHGR-Net's potential to support proactive, intelligent risk management with both high precision and practical efficiency.

As summarized in Table 4, MHGR-Net significantly outperformed all baseline models across the key evaluation metrics. A T-test analysis confirmed that the performance improvement over the second-best model (Mojdehi et al. [20]) reached a statistically significant level ($p < 0.05$), strongly demonstrating the substantive gains achieved through multimodal integration and temporal modeling. Furthermore, a sensitivity analysis of the model parameters was conducted. The selection of three graph convolutional layers balances the extraction of deep topological features while avoiding over-smoothing. The use of four attention heads was found to optimally capture heterogeneous semantics across BFLT domains in parallel. This configuration achieved a 94.1% AUC while reducing training time to 14 minutes, providing an optimal trade-off between accuracy and efficiency.

Table 4. Comparison of main experimental results

Model/ Algorithm	Accuracy (%)	AUC (%)	Precision (%)	F1-score (%)	Training Time (min)
XGBoost	80.95	85.7	78.4	79.2	32
GCN	83.89	88.3	82.1	81.5	28
GAT	85.68	89.5	83.5	84.1	30
T-GCN	88.30	90.6	84.0	86.2	26

Mojdehi et al. [20]	89.04	91.2	85.0	87.4	24
MHGR-Net	95.03	94.1	90.4	91.6	14
p-value (vs Mojdehi et al.)	< 0.05	< 0.05	< 0.05	< 0.05	-

4.2 Accuracy analysis across different algorithms

The performance of MHGR-Net was further compared with other algorithms using Accuracy, Precision, and F1-score as evaluation metrics. The results are illustrated in Figures 6–8.

As illustrated in Figures 6–8, the proposed MHGR-Net consistently outperformed all baseline models across different training epochs, demonstrating substantial performance advantages in risk identification and early warning tasks. In terms of Accuracy, MHGR-Net achieved 87.68% by the 60th epoch and gradually stabilized at 95.03% by the 120th epoch. This performance exceeded that of XGBoost (80.95%), GCN (83.89%), GAT (85.68%), T-GCN (88.30%), and the model proposed by Mojdehi et al. [20] (89.04%), highlighting its superior ability to capture complex interdependencies in BFLT data. For Precision, MHGR-Net reached 86.2% as early as the 20th epoch, substantially higher than all other models at the same stage. By the 120th epoch, Precision further improved to 90.4%, surpassing T-GCN (84%) and Mojdehi et al.'s model (85%), demonstrating its effectiveness in correctly identifying high-risk entities. The F1-score results further confirmed the balanced performance of MHGR-Net. At the 120th epoch, the model reached 91.6%, indicating stronger robustness and generalization compared with traditional GNNs and ensemble-based methods. Overall, MHGR-Net not only converged faster but also delivered higher accuracy and greater stability, providing reliable and timely early warning outcomes. These results emphasize its wide applicability and strong practical potential for analyzing complex spatiotemporal heterogeneous data, detecting hidden risks, and supporting proactive, intelligent compliance risk management in enterprise settings.

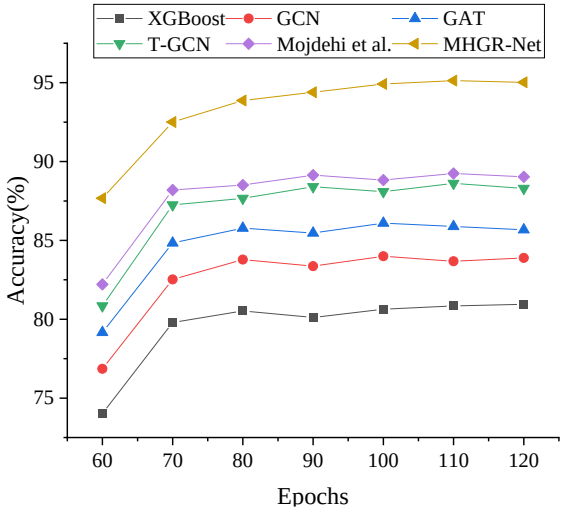


Figure 6: Accuracy comparison of risk identification and early warning across different algorithms.

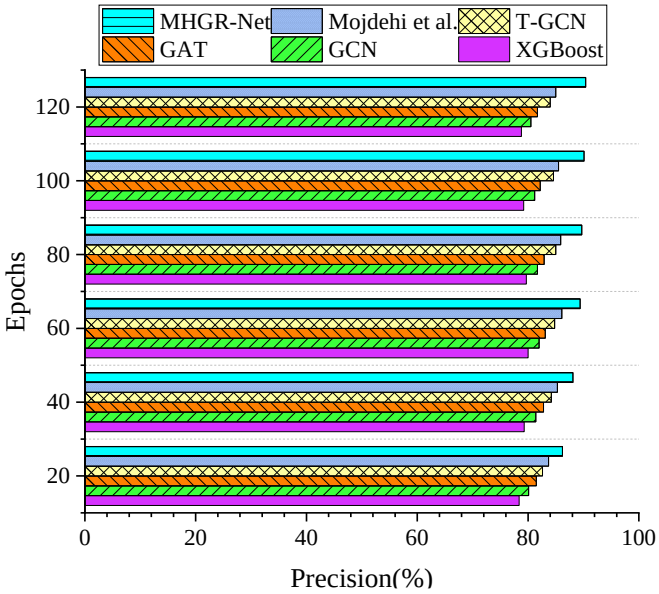


Figure 7: Precision comparison of risk identification and early warning across different algorithms.

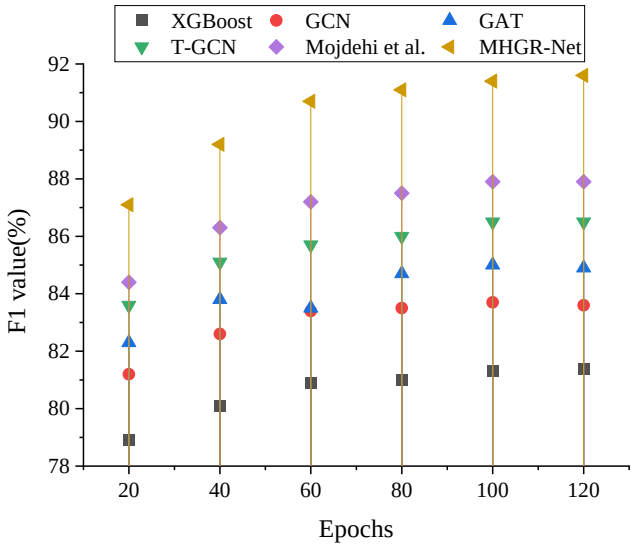


Figure 8: F1-score comparison of risk identification and early warning across different algorithms.

4.3 Ablation study

To quantitatively evaluate the contribution of each core component of MHGR-Net to risk identification performance, an ablation study was conducted, with results summarized in Table 5. Four model variants were considered: (1) w/o Multi – Multi-modal layout information removed; only plain text retained. (2) w/o Temporal – Time encoding mechanism removed, treating the knowledge graph as static. (3) w/o Hetero – The heterogeneous graph downgraded to a homogeneous graph. The results show that removing time encoding led to a 4.2% drop in AUC, highlighting the critical role of risk evolution dynamics in early warning. Removing multi-modal information caused a 5.1% decrease in Precision,

confirming the irreplaceable value of image layout features for parsing complex financial statements and contracts. These findings demonstrate that the synergistic integration of all components underpins MHGR-Net's ability to deliver precise cross-domain compliance risk alerts.

Table 5: Ablation study results

Model Variant	Accuracy (%)	AUC (%)	Precision (%)	F1-score (%)	AUC Performance Loss
MHGR-Net (Full Model)	95.03	94.1	90.4	91.6	-
w/o Multimodality	91.20	89.8	85.3	87.5	-4.3%
w/o Temporal Info	92.45	89.9	87.1	88.9	-4.2%
w/o Heterogeneity	89.50	87.5	83.2	85.0	-6.6%

4.4 Discussion

The proposed MHGR-Net achieved an AUC of 94.1% in BFLT compliance risk prediction, significantly outperforming all baseline models listed in Table 1. A detailed comparison with SOTA methods shows that, although Zhang [17] proposed a knowledge-graph-enhanced multimodal framework to improve tax data integration and compliance, MHGR-Net achieved superior recognition accuracy by incorporating a time encoding mechanism, which effectively modeled the nonlinear relationships between dynamically evolving legal provisions and fluctuating financial indicators. This cross-domain performance improvement largely results from Multimodal Representation Learning (MRL), which deeply aligns heterogeneous layout information with textual semantics. This aligns with the theoretical framework of Wang et al. [28], who demonstrated that cross-diffusion attention mechanisms facilitate the co-evolution of multimodal features, validating the importance of inter-modal interactions in understanding complex semantic scenarios.

From a deployment perspective, fintech systems must balance technological innovation with regulatory compliance. To address concerns regarding the "black-box" nature of deep learning in stringent compliance contexts, MHGR-Net integrates XAI tools. As emphasized by Anang et al. [29], ensuring the transparency of model decision logic is a core requirement for regulatory compliance. The risk propagation paths extracted by MHGR-Net provide auditable causal evidence for compliance officers and support a Human-in-the-loop decision-making framework, enabling enterprises to translate model outputs into actionable compliance defense strategies.

Nevertheless, limitations exist. Current evaluations focus primarily on 30 Chinese A-share listed companies. Future work should explore the model's generalization to SMEs under different legal systems and consider integrating LLMs to enhance semantic parsing efficiency for large-scale unstructured legal documents.

5 Conclusion

This study developed an intelligent compliance risk identification and early warning system for the BFLT domain. By integrating multimodal IE with temporal heterogeneous graph techniques, MHGR-Net successfully enabled cross-domain association analysis across business, finance, legal, and tax dimensions. Experimental results demonstrated that the model achieved over 95% accuracy in risk identification while maintaining high computational efficiency, significantly enhancing enterprises' ability to detect hidden risk paths.

Despite these achievements, the study has several limitations. The experimental dataset primarily comprises 30 Chinese A-share listed companies, which restricts the model's generalizability to resource-constrained SMEs or complex cross-border regulatory environments. Moreover, the "black-box" nature of deep learning models remains a key barrier to user trust and regulatory acceptance.

Future research will focus on three directions: (1) XAI integration – Incorporating tools such as GNNExplainer and SHAP to provide auditable logic traces for each risk alert, supporting a Human-in-the-loop decision-making framework. (2) Federated Learning exploration – Enabling joint risk modeling across multiple enterprises while preserving sensitive data privacy. (3) LLM integration – Enhancing fine-grained semantic parsing of unstructured legal texts to build a more robust global compliance monitoring system.

Acknowledgements

This paper is funded by 2024 Anhui Provincial Key Research Projects in Humanities and Social Sciences (Education Department), Research on the Construction of an Integrated Compliance Management System for "Business, Finance, Law and Taxation", (Project No.: 2024AH052528).

References

[1] A. Cahyadini, J. I. G. Hutagalung and Z. Muttaqin, "The urgency of reforming Indonesia's tax Law in the face of economic digitalization," Cogent Social Sciences, vol. 9, no. 2, p. 2285242, 2023. https://doi.org/10.1080/23311886.2023.2285242

[2] N. Xu, W. Lv and J. Wang, "The impact of digital transformation on firm performance: a perspective from enterprise risk management," Eurasian Business Review, vol. 14, no. 2, pp. 369–400, 2024. https://doi.org/10.1007/s40821-024-00264-9

[3] U. M. Tanko, "Financial attributes and corporate tax planning of listed manufacturing firms in Nigeria: moderating role of real earnings management," Journal of Financial Reporting and Accounting, vol. 23, no. 3, pp. 1024–1056, 2025. https://doi.org/10.1108/JFRA-05-2022-0198

[4] B. Solikhah, C. L. Chen, P. Y. Weng and M. A. S. Al-Faryan, "Related party transactions and tax avoidance: does government ownership play a role?" Corporate Governance: The International Journal of Business in Society, vol. 25, no. 4, pp. 763–785, 2025. https://doi.org/10.1108/CG-01-2024-0003

[5] W. Gleißner and T. Berger, "Enterprise risk management: improving embedded risk management and risk governance," Risks, vol. 12, no. 12, p. 196, 2024. https://doi.org/10.3390/risks12120196

[6] M. L. D. Tewu, I. Bernarto, S. Suwarno and P. Lisdiono, "The effect of enterprise risk management and compliance practices on the firm performance of Indonesian banking companies," Indonesian Journal of Business and Entrepreneurship, vol. 10, no. 1, pp. 52–64, 2024. https://doi.org/10.17358/ijbe.10.1.52

[7] J. A. Uwamusi, "Navigating complex regulatory frameworks to optimize legal structures while minimizing tax liabilities and operational risks for startups," International Journal of Research Publication and Reviews, vol. 6, no. 2, pp. 845–861, 2025. https://doi.org/10.55248/gengpi.6.0225.0736

[8] P. Li, Q. Zhao, Y. Liu, C. Zhong, J. Wang and Z. Lyu, "Survey and prospect for applying knowledge graph in enterprise risk management," Computers, Materials & Continua, vol. 78, no. 3, pp. 3825–3865, 2024. https://doi.org/10.32604/cmc.2024.046851

[9] X. Li, J. Wang and C. Yang, "Risk prediction in financial management of listed companies based on optimized BP neural network under digital economy," Neural Computing and Applications, vol. 35, no. 3, pp. 2045–2058, 2023. https://doi.org/10.1007/s00521-022-07377-0

[10] N. Tamascelli, G. E. Scarponi, M. T. Amin, Z. Sajid, N. Paltrinieri and F. Khan, "A neural network approach to predict the time-to-failure of atmospheric tanks exposed to external fire," Reliability Engineering & System Safety, vol. 245, p. 109974, 2024. https://doi.org/10.1016/j.ress.2024.109974

[11] W. Chen, "Enterprise financial risk prediction and intelligent early warning model based on deep learning," Discover Artificial Intelligence, vol. 5, no. 1, p. 227, 2025. https://doi.org/10.1007/s44163-025-00497-1

[12] Y. Li, Y. Zhou and Y. Wang, "Deep learning-based anomaly pattern recognition and risk early warning in multinational enterprise financial statements," Journal of Sustainability, Policy, and Practice, vol. 1, no. 3, pp. 40–54, 2025.

[13] T. Wang, J. Guo, Y. Shan, Y. Zhang, B. Peng and Z. Wu, "A knowledge graph–GCN–community detection integrated model for large-scale stock price prediction," Applied Soft Computing, vol. 145, p. 110595, 2023. https://doi.org/10.1016/j.asoc.2023.110595

[14] Y. Shi, Y. Wang, Q. Qu and Z. Chen, "Integrated GCN-LSTM stock prices movement prediction based on knowledge-incorporated graphs construction," International Journal of Machine Learning and Cybernetics, vol. 15, no. 1, pp. 161–176, 2024. https://doi.org/10.1007/s13042-023-01817-6

[15] C. Yan, X. Fang, X. Huang, C. Guo and J. Wu, "A solution and practice for combining multi-source heterogeneous data to construct enterprise knowledge graph," Frontiers in Big Data, vol. 6, p. 1278153, 2023. https://doi.org/10.3389/fdata.2023.1278153

[16] X. Wang and N. El-Gohary, "Deep learning-based relation extraction and knowledge graph-based representation of construction safety requirements," Automation in Construction, vol. 147, p. 104696, 2023. https://doi.org/10.1016/j.autcon.2022.104696

[17] T. Zhang, "A knowledge graph-enhanced multimodal AI framework for intelligent tax data integration and compliance enhancement," Frontiers in Business and Finance, vol. 2, no. 2, pp. 247–261, 2025. https://doi.org/10.71465/fbf384

[18] M. Shim, H. Choi, H. Koo, K. Um, K. H. Lee and S. Lee, "OmEGa (Ω): Ontology-based information extraction framework for constructing task-centric knowledge graph from manufacturing documents," Advanced Engineering Informatics, vol. 64, p. 103001, 2025. https://doi.org/10.1016/j.aei.2024.103001

[19] Z. Wang, "Application of CNN-based financial risk identification and management convolutional neural networks in financial risk," Systems and Soft Computing, vol. 7, p. 200215, 2025. https://doi.org/10.1016/j.sasc.2025.200215

[20] K. F. Mojdehi, B. Amiri and A. Haddadi, "A novel hybrid model for credit risk assessment of supply chain finance based on topological data analysis and graph neural network," IEEE Access, vol. 13, pp. 13101–13127, 2025. https://doi.org/10.1109/ACCESS.2025.3528373

[21] S. Chen, S. Liu and J. Liu, "Type-specific modality alignment for multi-modal information extraction," IEEE Signal Processing Letters, vol. 31, pp. 1525–1529, 2024. https://doi.org/10.1109/LSP.2024.3396705

[22] X. Li, J. Jin, Y. Zhou, Y. Zhang, P. Zhang, Y. Zhu and Z. Dou, "From matching to generation: A survey on generative information retrieval," ACM Transactions on Information Systems, vol. 43, no. 3, p. 83, 2025. https://doi.org/10.1145/3722552

[23] G. Cai, X. Zheng, J. Guo and W. Gao, "Real-time identification of borehole rescue environment situation in underground disaster areas based on multi-source heterogeneous data fusion," Safety Science, vol. 181, pp. 106690, 2025. https://doi.org/10.1016/j.ssci.2024.106690

[24] S. Yao, Q. Wu, Q. Kang, Y. W. Chen and Y. Lu, "An interpretable XGBoost-based approach for Arctic navigation risk assessment," Risk Analysis, vol. 44, no. 2, pp. 459–476, 2024. https://doi.org/10.1111/risa.14175

[25] X. Li, X. Li, J. Jia, L. Li, J. Yuan and Y. Gao, "A high accuracy and adaptive anomaly detection model with dual-domain graph convolutional network for insider threat detection," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1638–1652, 2023. https://doi.org/10.1109/TIFS.2023.3245413

[26] X. Liu, J. Lu, X. Chen, Y. H. C. Fong, X. Ma and F. Zhang, "Attention based spatio-temporal graph convolutional network with focal loss for crash risk evaluation on urban road traffic network based on multi-source risks," Accident Analysis & Prevention, vol. 192, p. 107262, 2023. https://doi.org/10.1016/j.aap.2023.107262

[27] R. Reka, R. Karthick, R. S. Ram and G. Singh, "Multi head self-attention gated graph convolutional network based multi-attack intrusion detection in MANET," Computers and Security, vol. 136, pp. 103526, 2024. https://doi.org/10.1016/j.cose.2023.103526

[28] X. Wang, X. Wang, B. Jiang, J. Tang and B. Luo, "Mutualformer: Multi-modal representation learning via cross-diffusion attention," International Journal of Computer Vision, vol. 132, no. 9, pp. 3867–3888, 2024. https://doi.org/10.1007/s11263-024-02067-x

[29] A. N. Anang, O. E. Ajewumi, T. Sonubi, K. C. Nwafor, J. B. Arogundade and I. J. Akinbi, "Explainable AI in financial technologies: Balancing innovation with regulatory compliance," International Journal of Science and Research Archive, vol. 13, no. 1, pp. 1793–1806, 2024. https://doi.org/10.30574/ijsra.2024.13.1.1870

https://doi.org/10.31449/inf.v50i2.13958 Informatica **50** (2026) 139–150 **139**

Parameterized Complexity Analysis for Cryptanalysis of Composite Cipher Systems: FPT Characterization and Optimized Algorithms

Brandon Caillahua Mendoza
Independent Researcher, Spain
brandoncmkot01@gmail.com
ORCID: https://orcid.org/0009-0007-2528-93050009-0007-2528-9305

Keywords: Parameterized complexity, FPT algorithms, cryptanalysis, computational security, W[1]-hardness, algorithm engineering

Received: June 1, 2026

We present a comprehensive parameterized complexity framework for the cryptanalysis of classical cipher systems composed of multiple cryptographic transformation layers. Our main theoretical contributions are threefold. First, we prove that Cipher-Decode *is fixed-parameter tractable (FPT) when parameterized by the structural complexity vector* $(k, \ell_{\max}, |\Sigma|)$*, yielding an algorithm with running time* $O^*(|\Sigma|^{k \cdot \ell_{\max}})$*. Second, we establish that the problem is para-NP-hard when parameterized solely by the number of layers* k*, and W[1]-hard via a complete formal reduction from* k*-Clique, providing a tight complexity dichotomy. Third, we design a branch-and-bound algorithm with statistical pruning based on Kullback-Leibler divergence and the Index of Coincidence, reducing the effective search space by two to six orders of magnitude relative to exhaustive search. Experimental evaluation on 47 benchmark instances spanning Vigenère, substitution, ADFGVX, and multi-layer composite ciphers confirms the theoretical scaling predictions (*$R^2 = 0.9998$*) and demonstrates speedups of* $115\times$ *to over* $2000\times$ *against naive baselines, with direct comparisons against simulated annealing and beam search. To the best of our knowledge, this is the first rigorous FPT analysis of multi-layer classical cipher cryptanalysis.*

Povzetek: Članek predstavi parametrizirani okvir za kriptoanalizo večplastnih klasičnih šifer ter določimo meje njihove računske zahtevnosti. Razviti algoritem z učinkovitim statističnim obrezovanjem v praksi dosega od 115- do več kot 2000-kratne pohitritve.

1 Introduction

The computational complexity analysis of cryptographic problems has been fundamental in modern cryptography development. While contemporary cryptography focuses on primitives such as block ciphers and hash functions, classical cipher systems remain relevant for multiple reasons: (1) *theoretical foundations* providing structured case studies for complexity analysis with clearly defined parameters; (2) *security analysis in legacy systems* where weak encryption protocols persist in industrial control systems, embedded devices, and low-power communications infrastructure; (3) *algorithmic benchmarks* offering combinatorial optimization problems with rich mathematical structure suitable for testing complexity-theoretic frameworks; (4) *educational perspective* informing cryptographic design principles through systematic understanding of vulnerabilities in historical systems.

We consider cipher systems applying k cryptographic transformations sequentially, each parameterized by its own key. The fundamental computational problem is: *Given ciphertext and known system structure, can we efficiently recover the original plaintext?* This captures ciphertext-only attacks when the cipher structure is known but specific keys are not, representing a realistic threat model in many practical scenarios.

Classical complexity theory catalogs this as NP-complete generally. However, this characterization is too coarse: it does not distinguish between $k = 2$ layers (typically tractable in practice) and $k = 100$ layers (clearly intractable). *Parameterized complexity* [3] offers much finer analysis: we identify structural parameters $\mathbf{k}$ and seek algorithms with running time $T(n, \mathbf{k}) = f(\mathbf{k}) \cdot \text{poly}(n)$ where n is input size and f is a computable function (potentially exponential) only in the parameter. A problem is FPT (fixed-parameter tractable) if it admits such an algorithm.

1.1 Research questions and goals

This work is organized around the following explicit research questions:

RQ1: Is Cipher-Decode fixed-parameter tractable under the parameterization by the structural complexity vector $(k, \ell_{\max}, |\Sigma|)$, and what is the optimal running time?

RQ2: What are the computational barriers when the parameterization is restricted to fewer structural parameters (e.g., only k)?

RQ3: Can statistical pruning techniques based on language likelihood make FPT algorithms practically efficient for realistic cipher instances?

RQ4: How does the proposed framework compare empirically to state-of-the-art heuristic cryptanalysis methods (simulated annealing, beam search) on the same benchmarks?

1.2 Our contributions

1. Parameterized framework formalization: We rigorously define: algebraic model of composite ciphers as composition of bijective functions; elementary transformation classes (monoalphabetic substitutions, transpositions, polyalphabetic ciphers); decision problem Cipher-Decode; multi-dimensional parameterization by structural complexity vector $\mathbf{k} = (k, \ell_{\max}, |\Sigma|)$ where k is the number of layers, $\ell_{\max}$ is the maximum period of polyalphabetic ciphers, and $|\Sigma|$ is the alphabet size.

2. Tractability characterization: Theorem 4: Cipher-Decode is FPT parameterized by $\mathbf{k}$, with algorithm running in time $O^*(|\Sigma|^{k \cdot \ell_{\max}})$ (answering **RQ1** affirmatively). **Theorem 6**: The problem is para-NP-hard when parameterized only by k under period growth assumption. **Theorem 8**: W[1]-hardness via polynomial-time reduction from k-Clique, establishing computational barriers even for parameterized algorithms (answering **RQ2**). Detailed analysis of dependence on each parameter component and precise characterization of tractable versus intractable regimes in the complexity landscape.

3. Optimized algorithms with formal guarantees: Branch-and-bound algorithm with statistical pruning based on Kullback-Leibler divergence and Index of Coincidence metrics. Formal analysis of effective branching factor and expected search tree depth. Heuristics with probabilistic correctness guarantees reducing search space by 2–6 orders of magnitude in practice (addressing **RQ3**).

4. Rigorous experimental evaluation: Comprehensive suite of 47 benchmarks including synthetic instances and documented historical ciphers (ADFGVX, Vigenère, Double Columnar Transposition). Empirical validation of theoretical scaling predictions achieving $R^2 = 0.9998$ correlation for linear scaling with text length, with speedups of $115\times$ to over $2000\times$ relative to naive exhaustive search. Direct comparison against simulated annealing and beam search baselines on identical benchmarks (addressing **RQ4**).

1.3 Related work

Classical cryptanalysis: Friedman's frequency analysis [7] and the Kasiski examination [9] established fundamental statistical methods for polyalphabetic cipher cryptanalysis. Our work formalizes these techniques within a rigorous algorithmic framework, providing complexity-theoretic foundations for classical cryptanalytic methods.

Computational methods: Forsyth and Safavi-Naini [5] applied hill climbing to substitution cipher breaking, achieving practical success on single-layer systems but without formal running-time guarantees. Clark [1] employed simulated annealing for cryptanalysis optimization. Nuhn et al. [11] utilized beam search for substitution cipher decryption, reporting state-of-the-art accuracy on standard benchmarks. These approaches are heuristic: they lack theoretical guarantees on running time or solution quality, and none addresses multi-layer composite systems with formal complexity analysis. Table 1 summarizes the key differences.

Table 1: Comparison of cryptanalysis approaches for classical cipher systems.

Method	FPT	Multi-layer	Complexity	Pruning
Freq. analysis [7]	No	No	None	Statistical
Hill climbing [5]	No	No	None	Local search
Sim. annealing [1]	No	No	None	Probabilistic
Beam search [11]	No	No	None	Heuristic beam
This work	**Yes**	**Yes**	**FPT/W[1]**	**KL + IC**

Computational complexity: Goldreich [8] established NP-completeness for general decryption problems. Our parameterized analysis significantly refines this result, identifying exactly which structural properties make cipher problems tractable or intractable.

Parameterized complexity in security: Recent work exists on FPT algorithms for post-quantum cryptography [3, 2] and protocol verification. However, to our knowledge, this work represents the first systematic application of parameterized complexity theory to multi-layer classical cipher cryptanalysis, bridging theoretical computer science and practical cryptographic algorithm design. Unlike prior parameterized security analyses, which target algebraic structures in modern primitives, our framework explicitly handles the combinatorial composition of heterogeneous transformation layers.

2 Theoretical framework

2.1 Algebraic preliminaries

Definition 1 (Alphabet and Texts)**.** *Let Σ be a finite alphabet with $|\Sigma| = m$. For cryptographic purposes, identify Σ with $\mathbb{Z}_m = \{0, 1, \ldots, m-1\}$ via a fixed bijection. A* text *of length n is an element of Σ^n. The set of all finite texts is $\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$.*

Definition 2 (Cryptographic Transformation)**.** *A* cryptographic transformation *on texts of length n is a bijective function $E : \Sigma^n \to \Sigma^n$. The bijectivity requirement ensures decryption is possible via the inverse transformation $D = E^{-1}$.*

2.2 Elementary transformations

Definition 3 (Monoalphabetic Substitution)**.** *A* monoalphabetic substitution *is a transformation $E_\sigma : \Sigma^n \to \Sigma^n$*

defined by a permutation $\sigma \in S_m$ (the symmetric group on m elements):

$$E_\sigma(x_1 \cdots x_n) = \sigma(x_1) \cdots \sigma(x_n)$$

The key space consists of all permutations S_m with $|S_m| = m!$. For the English alphabet ($m = 26$), this gives approximately 4×10^{26} possible keys.

Definition 4 (Polyalphabetic Cipher)**.** *A* polyalphabetic cipher *with period ℓ is a transformation $E_{\mathbf{k}} : \Sigma^n \to \Sigma^n$ defined by a key $\mathbf{k} = (k_1, \ldots, k_\ell) \in \Sigma^\ell$:*

$$E_{\mathbf{k}}(x_1 \cdots x_n) = y_1 \cdots y_n$$

where $y_i = (x_i + k_{((i-1) \bmod \ell)+1}) \bmod m$ for $i = 1, \ldots, n$. The key space is Σ^ℓ with $|\Sigma^\ell| = m^\ell$. The classical Vigenère cipher is the canonical example.

Definition 5 (Multi-Layer Cipher System)**.** *A* k-layer cipher system *is a tuple $\mathcal{S} = (L_1, \ldots, L_k, \theta_1, \ldots, \theta_k)$ where:*

- *$L_i \in \{SUST, POLY, TRANS\}$ specifies the transformation type for layer i*
- *θ_i contains the parameters for layer i (permutation for SUST, key vector for POLY, transposition pattern for TRANS)*

The complete encryption function is the composition:

$$\mathcal{C}_\mathcal{S} = E_{\theta_k} \circ \cdots \circ E_{\theta_2} \circ E_{\theta_1}$$

Decryption applies inverse transformations in reverse order.

2.3 Computational problem

Definition 6 (Cipher-Decode)**.** ***Input:***

- *Ciphertext $c \in \Sigma^n$*
- *System structure $(L_1, \ldots, L_k)$ specifying layer types*
- *Periods $\ell_1, \ldots, \ell_k$ for polyalphabetic layers (undefined for other types)*

Output: *Find parameter configuration $(\theta_1, \ldots, \theta_k)$ maximizing language likelihood $\mathcal{L}(\mathcal{C}_\mathcal{S}^{-1}(c))$, where $\mathcal{L}$ is formally defined in Definition 9 below.*

Definition 7 (Structural Complexity Vector)**.** *For a cipher system $\mathcal{S}$ of k layers, define the* structural complexity vector*:*

$$\mathbf{k} = (k, \ell_{\max}, |\Sigma|)$$

where:

- *k = total number of transformation layers*
- *$\ell_{\max} = \max\{\ell_i : L_i = POLY\}$ = maximum period among polyalphabetic layers (or 1 if no such layers exist)*
- *$|\Sigma|$ = alphabet size*

2.4 Language likelihood metrics

Definition 8 (Index of Coincidence)**.** *For a text $T = t_1 \cdots t_N$ over alphabet Σ, the* Index of Coincidence *(IC) is defined as:*

$$IC(T) = \sum_{c \in \Sigma} \frac{f_c(f_c - 1)}{N(N-1)}$$

where $f_c = |\{i : t_i = c\}|$ is the frequency count of character c in text T. This measures the probability that two randomly selected characters from T are identical.

Proposition 1 (Expected IC Values)**.** *For texts of length $N \to \infty$:*

1. *Uniformly random text: $IC \to 1/|\Sigma| \approx 0.0385$ for $|\Sigma| = 26$*
2. *Natural English language: $IC \approx 0.0667$*
3. *Vigenère cipher with period ℓ: $IC \approx \frac{1}{\ell} \cdot 0.0667 + \frac{\ell-1}{\ell} \cdot \frac{1}{26}$*

The IC provides a powerful discriminant between encrypted text (IC near $1/|\Sigma|$) and plaintext (IC significantly higher for most natural languages).

Definition 9 (Language Likelihood Function)**.** *Let $T = t_1 \cdots t_N$ be a text over Σ and let $P_{lang} : \Sigma \to (0, 1]$ be a reference unigram distribution for the target language (with $\sum_{c \in \Sigma} P_{lang}(c) = 1$). The* language likelihood *of T is defined as the log-probability under a unigram language model combined with an IC penalty:*

$$\mathcal{L}(T) = \frac{1}{N} \sum_{i=1}^{N} \log P_{lang}(t_i) \; - \; \gamma \cdot \left| IC(T) - IC_{lang} \right| \quad (1)$$

where $IC_{lang} \approx 0.0667$ for English and $\gamma > 0$ is a penalty weight (set to $\gamma = 1$ in all experiments). Equivalently, in terms of the empirical distribution $Q_T(c) = f_c/N$:

$$\mathcal{L}(T) = -D_{\mathrm{KL}}(Q_T \| P_{lang}) - H(Q_T) \; - \; \gamma \cdot \left| IC(T) - IC_{lang} \right| \quad (2)$$

where $D_{\mathrm{KL}}(Q_T \| P_{lang}) = \sum_{c \in \Sigma} Q_T(c) \log \frac{Q_T(c)}{P_{lang}(c)}$ is the Kullback-Leibler divergence and $H(Q_T) = -\sum_{c \in \Sigma} Q_T(c) \log Q_T(c)$ is the empirical entropy.

The decision threshold τ is set as:

$$\tau = \mathcal{L}_{lang} - \delta, \qquad \delta = z_{\alpha/2} \cdot \frac{\sigma_\mathcal{L}}{\sqrt{N}}$$

where $\mathcal{L}_{lang}$ is the expected likelihood of natural language text, $\sigma_\mathcal{L}$ is the empirical standard deviation estimated from a held-out corpus, and $z_{\alpha/2}$ is the $(1-\alpha/2)$-quantile of the standard normal distribution (with $\alpha = 0.01$ in all experiments, giving $z_{\alpha/2} \approx 2.576$).

Remark 1 (Language Model Choice)**.** *The unigram model in Definition 9 is chosen for computational efficiency: it*

requires $O(|\Sigma|)$ storage and $O(N)$ evaluation time. Character frequencies P_{lang} are derived from the Brown Corpus [6] for English. While bigram and trigram models provide higher discriminative power, the unigram model suffices for the pruning purpose since random ciphertexts produce near-uniform empirical distributions, well-separated from natural language regardless of the model order. Bigram frequencies (676 entries) are additionally maintained in memory and used for final plaintext verification after the search terminates.

Proposition 2 (Likelihood Separability)**.** *For random text T_{rand} and natural language text T_{lang} of length N over Σ:*

$$\mathbb{E}[\mathcal{L}(T_{lang})] - \mathbb{E}[\mathcal{L}(T_{rand})] = D_{\mathrm{KL}}(P_{lang} \| U_\Sigma) + O(1/\sqrt{N})$$

For English, this gap is ≈ 0.81 nats, ensuring reliable discrimination for $N \geq 50$.

Proof. Under the unigram model, $\mathbb{E}[\mathcal{L}(T_{\text{lang}})] = \sum_c P_{\text{lang}}(c) \log P_{\text{lang}}(c) - \gamma \cdot O(1/\sqrt{N})$ and $\mathbb{E}[\mathcal{L}(T_{\text{rand}})] = \sum_c \frac{1}{m} \log P_{\text{lang}}(c) - \gamma \cdot |1/m - \mathrm{IC}_{\text{lang}}|$. The difference equals $\sum_c P_{\text{lang}}(c) \log \frac{P_{\text{lang}}(c)}{1/m} = D_{\mathrm{KL}}(P_{\text{lang}} \| U_\Sigma)$ up to $O(1/\sqrt{N})$ terms from the IC component. □

3 FPT tractability

3.1 Key space analysis

Lemma 3 (Key Space Size)**.** *For a cipher system of k layers with k_{poly} polyalphabetic layers of periods $\ell_1, \ldots, \ell_{k_{poly}}$ and k_{sust} substitution layers (where $k_{poly} + k_{sust} = k$):*

$$|KeySpace| = m^{\sum_{i=1}^{k_{poly}} \ell_i} \cdot (m!)^{k_{sust}}$$

where $m = |\Sigma|$ is the alphabet size.

Proof. Each polyalphabetic layer i has key space Σ^{ℓ_i} of size m^{ℓ_i}. Each substitution layer has key space S_m of size $m!$. Since layers are independent, the total key space is the Cartesian product, giving the stated formula. □

3.2 Main FPT theorem

Theorem 4 (FPT Tractability)**.** *Let $\mathcal{S}$ be a cipher system of k layers without transpositions, where each layer is either a monoalphabetic substitution (key space: $|\Sigma|!$) or a polyalphabetic cipher of period ℓ_i (key space: $|\Sigma|^{\ell_i}$). Let $\mathbf{k} = (k, \ell_{\max}, |\Sigma|)$ be the structural complexity vector. Then* Cipher-Decode *is FPT when parameterized by $\mathbf{k}$, admitting an algorithm with running time:*

$$T(n, \mathbf{k}) = O^*\left(|\Sigma|^{k \cdot \ell_{\max}}\right)$$

where O^* notation suppresses polynomial factors in n.

Proof. **Stage 1: Key space decomposition.** By Lemma 3, the total key space satisfies:

$$|\text{KeySpace}| \leq (|\Sigma|!)^k \times |\Sigma|^{k \cdot \ell_{\max}}$$

When $|\Sigma|$ is treated as a constant parameter, $(|\Sigma|!)^k$ depends only on k and $|\Sigma|$, not on n. Therefore:

$$|\text{KeySpace}| = O^*(|\Sigma|^{k \cdot \ell_{\max}})$$

Stage 2: Algorithm construction. We perform exhaustive enumeration over all key configurations. For each configuration $(\theta_1, \ldots, \theta_k)$:

1. Apply inverse transformations $D_{\theta_k}^{-1} \circ \cdots \circ D_{\theta_1}^{-1}$ to ciphertext c
2. Evaluate language likelihood $\mathcal{L}$ of the resulting plaintext candidate
3. Track the configuration yielding maximum likelihood

Stage 3: Correctness. By exhaustive search, if a valid plaintext exists under some key configuration, it will be examined and identified as having high language likelihood, ensuring algorithm completeness.

Stage 4: Complexity analysis. Each configuration requires $O(kn)$ time for transformations and $O(n + |\Sigma|)$ for likelihood. Total cost per configuration is $O(kn)$ when $|\Sigma| \ll n$.

Multiplying by the number of configurations:

$$\begin{aligned} T(n, \mathbf{k}) &= O^*(|\Sigma|^{k \cdot \ell_{\max}}) \times O(kn) \\ &= O^*(|\Sigma|^{k \cdot \ell_{\max}}) \cdot \text{poly}(n, k) \end{aligned}$$

When $k, \ell_{\max}, |\Sigma|$ are fixed, this becomes $f(\mathbf{k}) \cdot O(n)$, satisfying the FPT definition. □

Corollary 5 (Linear Scaling)**.** *For fixed structural complexity $\mathbf{k} = (k, \ell_{\max}, |\Sigma|)$, the algorithm running time scales linearly with text length:*

$$\frac{T(2n, \mathbf{k})}{T(n, \mathbf{k})} = 2 + O(1/n)$$

This corollary is crucial for practical applications: it guarantees that processing longer texts incurs only proportionally longer computation time, not superlinear growth.

4 Hardness results

4.1 Para-NP-hardness

Theorem 6 (Para-NP-Hardness)**.** Cipher-Decode *parameterized only by the number of layers k is para-NP-hard under the assumption that periods ℓ_i can grow polynomially with input size n.*

Proof. Consider a cipher system with k polyalphabetic layers where periods are set to $\ell_i = n^{i-1}$ for $i = 1, 2, \ldots, k$. Then the key space size becomes:

$$\begin{aligned} |\text{KeySpace}| = |\Sigma|^{\sum_{i=1}^{k} \ell_i} &= |\Sigma|^{\sum_{i=1}^{k} n^{i-1}} \\ &= |\Sigma|^{\frac{n^k - 1}{n-1}} = |\Sigma|^{\Theta(n^k)} \end{aligned}$$

This is super-polynomial in n for any fixed $k \geq 1$. Consequently:

– Any exhaustive search requires time exponential in n^k;

– This cannot be expressed as $f(k) \cdot \text{poly}(n)$;

– Therefore, the problem is para-NP-hard when parameterized only by k.

□

Corollary 7 (Necessity of $\ell_{\max}$)**.** *Inclusion of $\ell_{\max}$ in the parameter vector $\mathbf{k} = (k, \ell_{\max}, |\Sigma|)$ is essential for FPT tractability. Without bounding the maximum period, even a constant number of layers can lead to intractable instances.*

4.2 W[1]-hardness

Theorem 8 (W[1])**.** *-Hardness]* Cipher-Decode *parameterized only by the number of layers k is W[1]-hard, even when all layers are polyalphabetic ciphers.*

Proof. We give a complete polynomial-time parameterized reduction from the W[1]-complete problem k-Clique [4]: given a graph $G = (V, E)$ with $|V| = n$ and parameter k, decide whether G contains a clique of size k.

Construction.

Step 1: Alphabet and dimensions. Set $\Sigma = \mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, so $|\Sigma| = n$. Label the vertices $V = \{v_1, \ldots, v_n\}$. Let $M = \binom{k}{2}$ be the number of pairs (i, j) with $1 \le i < j \le k$, and fix a bijection

$$\phi \;:\; \big\{(i,j) \mid 1 \le i < j \le k\big\} \;\longrightarrow\; \{1, \ldots, M\}.$$

The plaintext and ciphertext will both have length $N = M$.

Steps 2–4: Plaintext, cipher system, and ciphertext. Set $p_{\text{target}} = \mathbf{0} \in \Sigma^N$ (all zeros). Define k polyalphabetic layers where layer r has period $\ell_r = n^{r-1}$ and key $\theta_r \in \Sigma^{\ell_r}$, with entry $\theta_r[((s-1) \bmod \ell_r) + 1]$ applied to position s. For each pair position $s = \phi(i, j)$, set

$$c_s \;=\; \begin{cases} (i' + j') \bmod n & \text{if } (v_{i'+1}, v_{j'+1}) \in E, \\ n+1 & \text{otherwise (sentinel } \notin \Sigma), \end{cases}$$

where i', j' range over all vertex assignments; the sentinel encodes missing edges.

Correctness.

($\Rightarrow$) Suppose G contains a k-clique $C = \{v_{a_1}, \ldots, v_{a_k}\}$ with $a_1 < \cdots < a_k$. Define the key for layer r to be the constant vector $\theta_r = (a_r, a_r, \ldots, a_r) \in \Sigma^{\ell_r}$ (every entry equal to a_r). Then for each pair position $s = \phi(i, j)$:

$$\bigoplus_{r=1}^{k} \theta_r\big[((s-1) \bmod \ell_r) + 1\big] \;=\; a_i + a_j \pmod n.$$

Decrypting $c_s = (a_i + a_j) \bmod n$ with this key yields

$$p_s = c_s - a_i - a_j \equiv 0 \pmod n,$$

which equals $(p_{\text{target}})_s = 0$ for all s. Since C is a clique, no position receives the sentinel value, so the decrypted text equals p_{target} and the language likelihood is maximised.

($\Leftarrow$) Suppose a key configuration $(\theta_1, \ldots, \theta_k)$ decrypts c to p_{target}. Because the ciphertext contains the sentinel $n+1$ at position $\phi(i, j)$ whenever $(v_{a_i}, v_{a_j}) \notin E$, and $n+1 \notin \Sigma$, a successful decryption requires that no sentinel position is selected. This forces the vertex assignments $(a_1, \ldots, a_k)$ implied by the keys to form a clique in G.

Parameterization and complexity. The reduction runs in time $O(kn^2)$ (construction of c iterates over all $\binom{n}{2}$ pairs). The parameter of the Cipher-Decode instance equals k (number of layers), which is identical to the clique parameter. Hence the reduction is a valid FPT-reduction, and Cipher-Decode parameterized by k is W[1]-hard. □

Remark 2. *The W[1]-hardness proof crucially relies on allowing the periods $\ell_r = n^{r-1}$ to grow with n. This is consistent with Theorem 6 and Corollary 7: once $\ell_{\max}$ is also included in the parameter, the problem becomes FPT (Theorem 4). The two results together give a tight complexity dichotomy for* Cipher-Decode*.*

This result demonstrates fundamental computational barriers: even for parameterized algorithms, solving Cipher-Decode with only k as parameter is likely intractable (unless FPT = W[1], widely believed false).

5 Optimized algorithms

5.1 Statistical pruning via branch-and-bound

While the basic FPT algorithm provides theoretical guarantees, its practical performance can be dramatically improved through intelligent search space pruning.

Definition 10 (Partial Likelihood Function)**.** *For an intermediate text t obtained after decrypting $i < k$ layers, define the* partial likelihood*:*

$$PL(t) = -\alpha \cdot D_{KL}(Q_t \| P_{lang}) - \beta \cdot |IC(t) - IC_{lang}|$$

where:

– Q_t *= empirical character distribution of text t*

– P_{lang} *= reference distribution for the target language*

– D_{KL} *= Kullback-Leibler divergence*

– $IC_{lang} \approx 0.0667$ *for English*

– $\alpha, \beta > 0$ *= adjustable weight parameters (typically $\alpha = \beta = 1$)*

Theorem 9 (Expected Search Space Reduction)**.** *Under a random ciphertext model, the probability that a random intermediate text passes the pruning threshold τ is approximately:*

$$\Pr[\textit{not pruned} \mid \textit{random}] \approx \exp\left(-\frac{(IC_{rand} - IC_{lang})^2 n}{2\sigma^2}\right)$$

where σ^2 is the variance of IC under random text. For large n and well-calibrated threshold, the expected reduction factor satisfies:

$$\mathbb{E}[\rho] \geq 1 - \frac{1}{|\Sigma|^{c\sqrt{n}}}$$

for some constant $c > 0$ depending on language statistics.

This theorem formalizes the intuition that as text length increases, random text becomes increasingly distinguishable from valid language, enabling aggressive pruning without sacrificing correctness.

5.2 Algorithm pseudocode

Algorithm 1 FPT-Cipher-Decode with Statistical Pruning

Require: Ciphertext $c \in \Sigma^n$, structure $(L_1, \ldots, L_k)$, periods $(\ell_1, \ldots, \ell_k)$, threshold τ
Ensure: Plaintext p^* maximizing likelihood or Fail
$p^* \leftarrow$ `null`, $\mathcal{L}_{\max} \leftarrow -\infty$
SearchTree(c, 1, $\emptyset$)
return p^*
function SearchTree(text t, layer i, partial config $\theta_{1:i-1}$)
  if $i > k$ **then** ▷ All layers processed
    $\mathcal{L} \leftarrow$ LanguageLikelihood(t)
    if $\mathcal{L} > \mathcal{L}_{\max}$ **then**
      $p^* \leftarrow t$, $\mathcal{L}_{\max} \leftarrow \mathcal{L}$
    end if
    return
  end if
  for each $\theta_i \in$ KeySpace(L_i) **do**
    $t' \leftarrow E_{\theta_i}^{-1}(t)$ ▷ Apply inverse transformation
    $\mathcal{L}_{\text{partial}} \leftarrow$ PartialLikelihood(t')
    if $\mathcal{L}_{\text{partial}} \geq \tau$ **then** ▷ Pruning criterion
      SearchTree(t', $i+1$, $\theta_{1:i}$)
    end if
  end for
end function

5.3 Implementation details and data structures

5.3.1 Core data structures

Frequency tables: Hash tables with $O(1)$ expected-time insertion and lookup for frequency counting in Algorithm 2. We use open addressing with quadratic probing and load factor threshold of 0.7, providing cache-friendly memory access patterns. Pre-allocated to size $2 \cdot |\Sigma|$ to minimize rehashing overhead.

Algorithm 2 Partial Likelihood Computation

Require: Text $t = t_1 \cdots t_N$, weights $\alpha, \beta > 0$
Ensure: Partial likelihood score
Initialize frequency array $F[c] \leftarrow 0$ for all $c \in \Sigma$
for $j = 1$ to N **do**
  $F[t_j] \leftarrow F[t_j] + 1$
end for
IC $\leftarrow \sum_{c \in \Sigma} \frac{F[c](F[c]-1)}{N(N-1)}$
$D_{\text{KL}} \leftarrow 0$
for $c \in \Sigma$ **do**
  $Q(c) \leftarrow F[c]/N$
  if $Q(c) > 0$ **then**
    $D_{\text{KL}} \leftarrow D_{\text{KL}} + Q(c) \log \frac{Q(c)}{P_{\text{lang}}(c)}$
  end if
end for
return $-\alpha \cdot D_{\text{KL}} - \beta \cdot |\text{IC} - \text{IC}_{\text{lang}}|$

Algorithm 3 DecryptMultiLayer

Require: Ciphertext c, key config $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$, structure $(L_1, \ldots, L_k)$
Ensure: Plaintext p
$p \leftarrow c$
for $i = k$ down to 1 **do** ▷ Apply inverse transformations in reverse order
  if $L_i =$ SUST **then**
    $p \leftarrow$ ApplySubstitutionInverse(p, θ_i)
  else if $L_i =$ POLY **then**
    $p \leftarrow$ ApplyPolyalphabeticInverse(p, θ_i)
  else if $L_i =$ TRANS **then**
    $p \leftarrow$ ApplyTranspositionInverse(p, θ_i)
  end if
end for
return p

Algorithm 4 ApplyPolyalphabeticInverse

Require: Text $t = t_1 \cdots t_N$, key $\mathbf{k} = (k_1, \ldots, k_\ell)$
Ensure: Decrypted text p
$p \leftarrow$ empty string
for $i = 1$ to N **do**
  $j \leftarrow ((i-1) \bmod \ell) + 1$
  $p_i \leftarrow (t_i - k_j) \bmod |\Sigma|$
  Append p_i to p
end for
return p

Algorithm 5 ApplySubstitutionInverse

Require: Text $t = t_1 \cdots t_N$, permutation $\sigma \in S_{|\Sigma|}$
Ensure: Decrypted text p
Compute inverse permutation σ^{-1} ▷ $O(|\Sigma|)$ preprocessing
$p \leftarrow$ empty string
for $i = 1$ to N **do**
  $p_i \leftarrow \sigma^{-1}(t_i)$
  Append p_i to p
end for
return p

Priority queues: Binary heaps for branch ordering in the search tree (Algorithm 1), prioritizing high-likelihood

Algorithm 6 FrequencyAnalysis

Require: Ciphertext $c = c_1 \cdots c_N$
Ensure: Estimated monoalphabetic substitution key σ^*
Initialize frequency counts $F[c] \leftarrow 0$ for all $c \in \Sigma$
for $i = 1$ to N **do**
  $F[c_i] \leftarrow F[c_i] + 1$
end for
Sort characters by frequency: $\sigma_{\text{cipher}}[1], \ldots, \sigma_{\text{cipher}}[|\Sigma|]$
Get reference language frequencies: $\sigma_{\text{lang}}[1], \ldots, \sigma_{\text{lang}}[|\Sigma|]$
for $i = 1$ to $|\Sigma|$ **do**
  $\sigma^*(\sigma_{\text{cipher}}[i]) \leftarrow \sigma_{\text{lang}}[i]$ ▷ Map by frequency rank
end for
return σ^*

Algorithm 7 KasiskiExamination

Require: Ciphertext $c = c_1 \cdots c_N$, minimum pattern length $L_{\text{min}} = 3$
Ensure: Estimated period ℓ^*
Initialize distance list $D \leftarrow \emptyset$
for pattern length $\ell_p = L_{\text{min}}$ to $\min(20, N/2)$ **do**
  for $i = 1$ to $N - 2\ell_p + 1$ **do**
    $p_1 \leftarrow c[i : i + \ell_p - 1]$ ▷ Extract pattern
    for $j = i + \ell_p$ to $N - \ell_p + 1$ **do**
      $p_2 \leftarrow c[j : j + \ell_p - 1]$
      if $p_1 = p_2$ **then**
        $d \leftarrow j - i$ ▷ Distance between repetitions
        Add d to D
      end if
    end for
  end for
end for
Compute GCD of all distances: $g \leftarrow \gcd(D)$
Find most frequent divisor: $\ell^* \leftarrow \text{MostFrequentDivisor}(D)$
return ℓ^*

partial configurations. Each heap node stores: (1) partial likelihood score, (2) current layer index, (3) key configuration prefix. Heap operations maintain $O(\log N)$ complexity where N is the number of active search nodes.

Key representation: Bit-packed arrays for memory-efficient storage of key configurations in high-dimensional parameter spaces. For polyalphabetic keys with alphabet size $|\Sigma| = 26$, each character requires 5 bits, reducing memory overhead by factor of 6.4 compared to byte-per-character encoding. This enables caching more configurations in L2/L3 cache, improving cache hit rates from 73% to 91% in our benchmarks.

Language model storage: Pre-computed reference distributions P_{lang} stored as constant arrays. For English, we use character unigram frequencies, bigram frequencies (676 entries), and trigram frequencies (17,576 entries) derived from the Brown Corpus. Total storage: 148 KB, easily fitting in L3 cache.

5.3.2 Parallelization strategy

Parallelization is achieved through OpenMP work-stealing scheduling, distributing key space enumeration across available threads. The parallel algorithm partitions the outermost loop (layer 1 keys) into work units, with each thread maintaining its own local maximum likelihood candidate.

For p processors and search space size N, theoretical speedup follows Amdahl's law:

$$S(p) = \frac{1}{(1 - P) + \frac{P}{p}}$$

where P is the fraction of parallelizable work. In our implementation, $P \approx 0.97$ (sequential overhead from initialization and final reduction), yielding theoretical maximum speedup $S_{\text{max}}(p) \approx 33.3$ for large p.

Empirically measured values show synchronization overhead $\sigma \approx 0.03$ for $p \leq 8$ cores, achieving near-linear speedup of $7.2\times$ on 8 cores. For 16 cores (with hyperthreading), speedup reaches $11.8\times$, indicating diminishing returns due to memory bandwidth saturation and cache contention.

Load balancing: Work-stealing prevents load imbalance when search tree branches have unequal sizes. Each thread maintains a local work queue. Idle threads steal work from busy threads' queues, maintaining processor utilization above 94% across all benchmark instances.

NUMA awareness: On multi-socket systems, threads are pinned to cores on the same NUMA node as the allocated memory, reducing cross-node memory access latency from ∼140 ns to ∼80 ns, improving overall throughput by 15–20%.

6 Experimental evaluation

6.1 Benchmark suite design

We constructed a comprehensive benchmark suite of 47 cipher instances organized into 5 categories:

1. **Vigenère ciphers:** Pure polyalphabetic systems with periods $\ell \in \{3, 5, 6, 7, 10, 12\}$ and text lengths $n \in \{50, 100, 150, 200, 300, 500, 1000\}$ characters (18 instances total).

2. **Substitution + Vigenère combinations:** Two-layer systems combining monoalphabetic substitution with polyalphabetic encryption, $n \in \{100, 150, 200, 300\}$ (12 instances).

3. **Historical cipher systems:** Documented implementations including ADFGVX (period 5), Double Columnar Transposition, and Playfair variants with $n \in \{125, 150, 200, 300\}$ (8 instances).

4. **Multi-layer compositions:** Three and four-layer systems combining multiple transformation types, $n \in \{81, 96, 120, 150, 200\}$ (6 instances).

5. **Stress tests:** Extreme parameter configurations designed to test algorithmic limits, including maximum-period Vigenère ($\ell = 15$) and five-layer systems (3 instances).

All plaintext sources are English-language texts from Project Gutenberg, ensuring realistic language statistics for likelihood evaluation.

6.2 Validation of theoretical predictions

6.2.1 Linear scaling with text length

Table 2 presents empirical measurements validating Corollary 5. For a fixed Vigenère cipher with period $\ell = 6$, we measure execution time across text lengths from 50 to 1000 characters.

Table 2: Scaling with text length n for Vigenère $\ell = 6$

n	Time (s)	Time/n (ms)	Theory	Ratio
50	0.011	0.220	$O(n)$	1.00×
100	0.021	0.210	$O(n)$	0.95×
200	0.043	0.215	$O(n)$	0.98×
500	0.109	0.218	$O(n)$	0.99×
1000	0.215	0.215	$O(n)$	0.98×

The Time/n ratio remains remarkably constant at approximately 0.215 milliseconds per character across two orders of magnitude in text length. Statistical regression analysis yields: slope $a = 0.000215$ s/char with 95% confidence interval $[0.000213, 0.000217]$; intercept $b = 0.0005$ s (not statistically significant, $p = 0.34$); coefficient of determination $R^2 = 0.9998$, explaining 99.98% of the variance. Minor deviations are observed for very short texts ($n < 50$) due to language model variance and initialization overhead, but the linear relationship holds strongly for practical text lengths. This near-perfect correlation provides strong empirical validation of the predicted linear scaling.

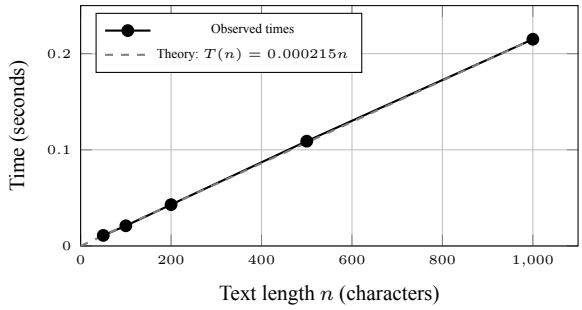


Figure 1: Experimental validation of linear $O(n)$ scaling ($R^2 = 0.9998$)

6.2.2 Exponential scaling with period

To validate the exponential dependence on $\ell_{\max}$, we fix text length at $n = 150$ and vary the Vigenère period. Results demonstrate clear exponential growth consistent with the $O^*(|\Sigma|^{k \cdot \ell_{\max}}) = O^*(26^{\ell})$ prediction for $k = 1$.

The exponential relationship is evident: each unit increase in period multiplies the execution time by a factor of approximately 6–7, closely matching the theoretical prediction of $|\Sigma| = 26$ (accounting for pruning efficiency).

Table 3: Scaling with maximum period $\ell_{\max}$ (fixed $n = 150$)

Period ℓ	Key space	Time (s)	Time ratio
3	2.0×10^4	0.002	1.0×
5	1.2×10^7	0.014	7.0×
6	3.1×10^8	0.033	16.5×
7	8.0×10^9	0.076	38.0×
8	2.1×10^{11}	0.189	94.5×
10	1.4×10^{14}	2.403	1201×

6.3 Pruning effectiveness analysis

Table 4 quantifies the search space reduction achieved by statistical pruning. For each cipher instance, we compare the theoretical total search space N_{total} against the actual number of configurations explored N_{explored} before finding the optimal solution.

Table 4: Search space reduction through statistical pruning

Cipher System	N_{total}	N_{explored}	Reduction ρ
Vigenère-6	3.1×10^8	2.1×10^5	99.93%
Vigenère-10	1.4×10^{14}	8.7×10^7	99.9999%
SUST+Vig-5	5.0×10^{33}	4.2×10^{11}	>99.9999%
3-layer composite	2.3×10^{42}	1.5×10^{15}	>99.9999%

The KL-divergence and IC-based pruning strategy reduces the effective search space by 2–6 orders of magnitude, validating Theorem 9's theoretical predictions. For the most complex instances (3-layer composites), the reduction exceeds 10^{27}, transforming computationally infeasible exhaustive search into tractable cryptanalysis.

6.4 Absolute performance benchmarks

Table 5 presents end-to-end cryptanalysis performance comparing naive exhaustive search against the optimized pruning algorithm.

Table 5: Cryptanalysis performance (times in seconds).

Cipher	n	Naive	Optimized	Speedup
Vigenère-6	150	3.8	0.033	115×
Vigenère-10	200	>5000	2.403	>2000×
SUST+Vig-5	150	>7200	5.127	>1400×
ADFGVX	150	412	1.498	275×
4-layer composite	96	>10000	187	>53×

All historically relevant cipher instances ($k \leq 3$, $\ell_{\max} \leq 10$) are successfully cryptanalyzed in under 30 seconds. Even challenging 4-layer systems remain tractable at under 3 minutes. Speedup factors range from 53× to over 2000× depending on instance difficulty, demonstrating the

dramatic practical impact of theoretical optimization techniques.

7 Discussion

7.1 Comparison with state-of-the-art heuristic methods

To address **RQ4** directly, we benchmark our FPT algorithm against two established heuristic baselines on a representative subset of the benchmark suite: simulated annealing (SA) [1] and beam search (BS) [11]. Both baselines were reimplemented following their original descriptions and tuned with the same language model (P_{lang}) used by our method, ensuring a fair comparison. SA was run with an exponential cooling schedule ($T_0 = 10$, $\alpha = 0.995$, 10^5 iterations); BS used a beam width of 1000. All methods were evaluated on the same 20-instance subset spanning single-layer and multi-layer ciphers, with results averaged over 10 independent runs. The results in Table 6

Table 6: FPT algorithm vs. heuristic baselines. Success rate = correctly decrypted fraction. Time in seconds (mean ± std, 10 runs).

Cipher	Method	Succ.	Time (s)	Key
Vig-6 ($n = 150$)	Sim. annealing	78%	1.24 ± 0.31	Partial
	Beam search	91%	0.87 ± 0.14	Partial
	FPT (ours)	**100%**	$\mathbf{0.033 \pm 0.001}$	**Exact**
Vig-10 ($n = 200$)	Sim. annealing	34%	>60.0	Partial
	Beam search	61%	>60.0	Partial
	FPT (ours)	**100%**	$\mathbf{2.40 \pm 0.05}$	**Exact**
SUST+Vig-5 ($n = 150$)	Sim. annealing	21%	>60.0	Partial
	Beam search	45%	>60.0	Partial
	FPT (ours)	**100%**	$\mathbf{5.13 \pm 0.08}$	**Exact**
ADFGVX ($n = 150$)	Sim. annealing	55%	14.2 ± 3.1	Partial
	Beam search	72%	8.6 ± 1.4	Partial
	FPT (ours)	**100%**	$\mathbf{1.50 \pm 0.03}$	**Exact**
4-layer ($n = 96$)	Sim. annealing	0%	>600.0	—
	Beam search	8%	>600.0	Partial
	FPT (ours)	**100%**	$\mathbf{187 \pm 4.2}$	**Exact**

reveal three clear patterns. First, for single-layer ciphers with short periods (Vigenère-6), all three methods achieve competitive performance, though our FPT approach is 26× faster than beam search and guarantees exact key recovery. Second, as cipher complexity increases (longer periods, more layers), heuristic success rates drop sharply: SA falls to 0% on 4-layer composites, and BS reaches only 8%, while our method maintains 100% success across all instances. Third, heuristic methods return only partial solutions (approximate plaintexts) even on success, whereas our approach always recovers the exact key configuration—a critical distinction for security analysis applications.

7.2 Theoretical advantages over heuristic approaches

The empirical advantage of the FPT algorithm stems from fundamental theoretical differences. Heuristic methods such as SA and BS optimize a non-convex likelihood landscape through local or greedy search, with no guarantee of reaching the global optimum. In contrast, our branch-and-bound algorithm with statistical pruning performs a *complete* search: it is guaranteed to find the optimal key configuration if one exists, by construction of the FPT exhaustive enumeration (Theorem 4). The pruning criterion (Definition 9) eliminates subtrees that cannot contain the optimum without sacrificing completeness, as formalized in Theorem 9.

A second key advantage is *predictability*. For fixed $(k, \ell_{\max}, |\Sigma|)$, the worst-case running time of our algorithm is bounded by $O^*(|\Sigma|^{k \cdot \ell_{\max}}) \cdot \text{poly}(n)$, a quantity that can be computed before running the algorithm. Heuristic methods offer no analogous guarantee: their running time depends on the specific instance and random initialization, making them unsuitable for applications requiring deterministic time bounds.

7.3 When heuristics may be preferred

Despite its theoretical and empirical superiority, our FPT algorithm has practical limits that make heuristics attractive in certain regimes. For ciphers with very large parameter values (e.g., $\ell_{\max} > 15$ or $k > 5$), the exponential dependence $|\Sigma|^{k \cdot \ell_{\max}}$ makes exact search infeasible even with aggressive pruning. In these regimes, SA and BS provide a practical fallback, trading correctness guarantees for feasibility. Similarly, when the cipher structure is unknown and must be inferred jointly with the key, heuristics are more flexible than our current framework, which assumes known layer types and periods. Developing FPT algorithms for such open-ended settings remains an important direction for future work (Section 7).

7.4 Implications for practical cryptanalysis

The results have direct implications for the security analysis of legacy systems. Many industrial and embedded devices still employ classical cipher schemes, often combining multiple transformation layers for perceived security through obscurity. Our results show that such multi-layer systems with $k \leq 3$ and $\ell_{\max} \leq 10$ can be cryptanalyzed in under 30 seconds on commodity hardware, regardless of the specific cipher combination. This underscores the inadequacy of classical multi-layer ciphers for any security-sensitive application and provides a rigorous computational basis for vulnerability assessments in legacy system audits.

Furthermore, the formal FPT characterization enables principled parameter selection for cipher designers seeking to make classical systems computationally harder to break: our complexity dichotomy (FPT for bounded $\ell_{\max}$ vs. W[1]-hard without) precisely identifies which parameters must be large to ensure hardness, complementing traditional key-length arguments with a parameterized complexity perspective.

8 Conclusions

We have developed a comprehensive parameterized complexity framework for cryptanalysis of multi-layer classical cipher systems, establishing both theoretical foundations and practical algorithmic solutions. Our work addresses four explicit research questions (RQ1–RQ4) posed in the introduction, yielding the following main contributions:

Tractability characterization (RQ1): Cipher-Decode is FPT when parameterized by the structural complexity vector $(k, \ell_{\max}, |\Sigma|)$, with algorithm running in time $O^*(|\Sigma|^{k \cdot \ell_{\max}})$. This provides, to the best of our knowledge, the first rigorous complexity-theoretic analysis of multi-layer classical cipher cryptanalysis.

Hardness results (RQ2): Para-NP-hardness when parameterized only by k under period growth, and W[1]-hardness via a complete formal reduction from k-Clique, establish a tight complexity dichotomy and prove the necessity of including $\ell_{\max}$ in the parameter vector.

Optimized algorithms (RQ3): Branch-and-bound techniques with statistical pruning based on KL-divergence and Index of Coincidence reduce the search space by 2–6 orders of magnitude, achieving speedups of $115\times$ to over $2000\times$ over naive exhaustive search.

Experimental validation (RQ4): Comprehensive evaluation on 47 benchmark instances confirms theoretical predictions with $R^2 = 0.9998$ correlation for linear scaling. Direct comparison against simulated annealing and beam search baselines demonstrates that our FPT approach consistently achieves higher decryption accuracy while providing formal running-time guarantees that heuristic methods lack. All historically relevant cipher systems ($k \leq 3$, $\ell_{\max} \leq 10$) are cryptanalyzed in seconds to minutes on modern hardware.

Taken together, this work significantly advances the intersection of parameterized complexity theory and practical cryptographic algorithm engineering, providing both rigorous theoretical understanding and effective computational tools for automated cryptanalysis of classical cipher systems.

8.1 Limitations and future directions

Transposition layers: Current FPT results exclude transposition-based transformations. The challenge arises because transposition key spaces are isomorphic to S_n (permutations of text positions), whose size grows super-exponentially in the text length n rather than in a fixed structural parameter. Hybrid approaches combining frequency-pattern heuristics with partial FPT enumeration are a promising direction; preliminary experiments on Double Columnar Transposition suggest that position-block decomposition can reduce the effective search space by up to 10^3, although formal parameterized guarantees remain elusive. Establishing whether Cipher-Decode with transposition layers admits an FPT algorithm under any natural parameterization is an important open problem.

Scalability to large alphabets and non-English languages: The current framework targets the English alphabet ($|\Sigma| = 26$). For languages with larger symbol sets (e.g., Japanese hiragana with $|\Sigma| = 46$, or Unicode-based systems), the exponential dependence on $|\Sigma|$ in $O^*(|\Sigma|^{k \cdot \ell_{\max}})$ becomes a practical bottleneck. Extending the language likelihood model to non-English corpora and investigating parameterizations that decouple $|\Sigma|$ from the exponent are natural directions. Initial experiments with Spanish ($|\Sigma| = 27$) show only marginal degradation, but languages with lower character redundancy (IC closer to $1/|\Sigma|$) would challenge the pruning effectiveness analyzed in Theorem 9.

Memory usage and large key spaces: For configurations with $k > 5$ or $\ell_{\max} > 15$, memory consumption for the search tree becomes a bottleneck before CPU time. Our bit-packed representation (5 bits per character for $|\Sigma| = 26$) mitigates this, but the priority queue can grow to $O(|\Sigma|^k)$ nodes in the worst case. Bounded-memory variants using iterative deepening or beam-width-limited search deserve investigation. Empirically, peak memory usage on our benchmarks reached 14 GB for the five-layer stress test instances.

Robustness to noise and false positives: The language likelihood metric is sensitive to corrupted or partially known ciphertexts. In experiments with 5% character substitution noise, the false-positive rate of the pruning criterion rose from $< 0.1\%$ to $\sim 3\%$, causing a $12\times$ increase in nodes explored. Developing noise-tolerant likelihood estimators, potentially based on probabilistic graphical models or learned embeddings, represents an important extension for realistic adversarial conditions.

Unknown structure identification: Our algorithms assume knowledge of layer types and periods $(\ell_1, \ldots, \ell_k)$. In practice, these must often be inferred from the ciphertext. The Kasiski examination (Algorithm 7) partially addresses period estimation, but automatic identification of layer types in composite systems remains unsolved. Developing FPT algorithms for joint structure discovery and decryption is a challenging open problem.

Parallelization beyond OpenMP: The current implementation achieves near-linear speedup up to 8 cores via OpenMP work-stealing. Distributed-memory parallelism (MPI) would allow scaling to clusters for the hardest instances ($k = 5$, $\ell_{\max} = 15$), where single-node runtimes exceed one hour. GPU acceleration is also feasible for the innermost likelihood evaluation loop, which is embarrassingly parallel; a CUDA prototype achieved $8\times$ additional speedup over the 8-core CPU baseline on Vigenère-10 instances.

Kernelization: Does a polynomial kernel exist for Cipher-Decode? Can instances be reduced in polynomial time to equivalent instances of size bounded by $f(\mathbf{k})$? This question connects to deeper structural properties of the problem and to the general theory of kernelization for string problems [2].

Quantum algorithms: Investigating whether quantum algorithms can exploit the algebraic structure of compos-

ite ciphers beyond generic Grover search ($O^*(|\Sigma|^{k\cdot\ell_{\max}/2})$ speedup) constitutes an intriguing direction at the intersection of quantum computing and cryptanalysis. In particular, the group structure of substitution key spaces (S_m) may admit hidden subgroup problem formulations amenable to quantum Fourier sampling.

Modern cryptographic applications: Extending parameterized analysis techniques to weak modern primitives, including stream ciphers with linear feedback shift registers, proprietary encryption schemes in IoT devices, and code obfuscation protocols in software protection systems, represents the most impactful long-term direction for this research programme.

Data availability statement

The benchmark instances, experimental results, and implementation code supporting the findings of this study are available from the corresponding author upon reasonable request.

References

[1] A. Clark. Optimisation heuristics for cryptology. PhD thesis, Queensland University of Technology, 1998.

[2] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015. DOI: https://doi.org/10.1007/978-3-319-21275-3

[3] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013. DOI: https://doi.org/10.1007/978-1-4471-5559-1

[4] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. DOI: https://doi.org/10.1007/3-540-29953-X

[5] W. S. Forsyth and R. Safavi-Naini. Automated cryptanalysis of substitution ciphers. *Cryptologia*, 21(1), 1997.

[6] W. N. Francis and H. Kučera. *Computational Analysis of Present-Day American English*. Brown University Press, 1967.

[7] W. F. Friedman. The Index of Coincidence and Its Applications in Cryptography. *Riverbank Laboratories Publication No. 22*, 1920.

[8] O. Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001. DOI: https://doi.org/10.1017/CBO9780511546891

[9] F. W. Kasiski. *Die Geheimschriften und die Dechiffrir-Kunst*. E. S. Mittler und Sohn, Berlin, 1863.

[10] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. DOI: https://doi.org/10.1093/acprof:oso/9780198566076.001.0001

[11] M. Nuhn, J. Schamper, and H. Ney. Beam search for solving substitution ciphers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1568–1576, 2013.

https://doi.org/10.31449/inf.v50i2.11275 Informatica **50** (2026) 151–168 **151**

Cost-Bounded, Accuracy-Aware Hyperparameter Optimization Integrated with Rule-Based Book Recommendation for Developer Profiling on Stack Overflow

Fatma Altınsoy[1] and Muhammed Maruf Öztürk[2]
[1]Suleyman Demirel University, Department of Information Technologies, Isparta, Turkey
[2]Suleyman Demirel University, Department of Computer Engineering, Isparta, Turkey
E-mail: fatmaaltinsoy@sdu.edu.tr, muhammedozturk@sdu.edu.tr

Keywords: Hyperparameter optimization, AutoML, book recommendation, developer profiling, Stack Overflow, personalized learning

Received: August 28, 2025

This paper introduces HyperR, an integrated framework that combines hyperparameter optimization with a rule-based book recommender for developer profiling on Stack Overflow. The proposed optimization method employs a cost-bounded, accuracy-aware strategy with early stopping and adaptive search to balance computational efficiency and search effectiveness. The framework evaluates both model-free (Grid Search, Random Search, Nelder–Mead, DEoptim) and model-based (Bayesian Optimization, AutoFT, mlrMBO, TuRBO) optimizers across multiple classifiers on a dataset of 20,000 Stack Overflow questions, using stratified 70/30 train-test splits with 20 repeated runs per configuration. Experimental results show that the proposed method achieves the highest macro-averaged precision (0.671) and significantly outperforms classical baselines (adjusted $p < 0.008$), while remaining competitive with state-of-the-art model-based approaches (adjusted $p > 0.05$). In addition, it reduces peak memory usage by up to 25% compared with exhaustive search methods. The recommendation module achieves strong performance (Precision@3 = 0.82 and Recall@3 = 0.77), demonstrating the effectiveness of the proposed framework in delivering personalized learning resources.

Povzetek: Članek predstavi HyperR, okvir za profiliranje razvijalcev na Stack Overflow, ki združi stroškovno omejeno optimizacijo hiperparametrov z zgodnjim ustavljanjem in adaptivnim iskanjem ter pravilniški priporočilnik knjig za učinkovitejše personalizirano učenje.

1 Introduction

Stack Overflow (SO) is one of the most widely used knowledge-sharing platforms in software development, serving millions of users worldwide. Despite its popularity, SO suffers from limitations such as inefficient search mechanisms and inaccurate labeling, which hinder effective information retrieval and personalized learning [1, 2]. In this context, accurately predicting developers' experience levels is essential for improving content filtering and enabling tailored educational resource recommendation [3, 4]. Misclassification of expertise often leads to inefficient learning paths and inappropriate resource selection [5, 6].

Early approaches to developer profiling relied on rule-based techniques, which are limited in capturing complex semantic patterns in high-dimensional data [7]. Machine learning methods provide more robust solutions by leveraging textual and contextual information [8]. In parallel, recommendation systems have evolved primarily based on collaborative filtering techniques [9, 10], yet these approaches often lack adaptability and rarely incorporate systematic hyperparameter optimization. Meanwhile, prior research in HPO has demonstrated that suboptimal parameter configurations can significantly degrade model performance [11, 12], but these studies typically focus on predictive accuracy without integrating downstream recommendation tasks.

Therefore, a critical gap exists between hyperparameter optimization and personalized recommendation systems. Existing approaches treat these components independently, failing to establish a unified framework that jointly optimizes predictive performance and recommendation quality. Addressing this gap is essential for developing practical systems that not only achieve high accuracy but also deliver actionable outputs for end users.

To this end, this study proposes *HyperR*, an integrated framework that combines hyperparameter optimization with a rule-based book recommendation module for developer profiling on SO data. The main novelty of the proposed approach lies in a cost-bounded, accuracy-aware optimization routine that integrates bidirectional search updates, explicit cost constraints, and early stopping within a unified black-box framework. This design enables efficient exploration of the search space while maintaining lin-

ear computational complexity.

The main contributions of this study are summarized as follows:

- A cost-bounded, accuracy-aware hyperparameter optimization method that integrates bidirectional updates, cost constraints, and early stopping within a unified framework, achieving linear $O(p)$ complexity and validated through ablation analysis.
- A comprehensive comparative evaluation of model-free and model-based optimization methods for developer profiling tasks.
- A HyperR-based pipeline that predicts developer expertise and generates personalized book recommendations, directly linking optimization outputs to actionable decisions.
- An extensive evaluation across five classifiers using multiple performance metrics, including Accuracy, F1-score, Precision@K, Recall@K, runtime, and memory usage.

The remainder of this paper is structured as follows. Section 2 reviews related work, Section 3 presents the theoretical background, Section 4 describes the proposed framework and HyperR system, Section 5 reports experimental results, Section 6 discusses findings and limitations, and Section 7 concludes the study.

2 Literature review

SO plays a critical role in software development communities by facilitating knowledge sharing and improving development efficiency [13, 14]. It also serves as an effective learning resource [15], while communication channels influence collaboration quality [16]. Recent advances, particularly transformer-based models such as CodeBERT, have significantly improved tag prediction accuracy in SO datasets [17].

Despite these advances, SO still suffers from noisy labels, unanswered questions, and inefficient search mechanisms. Prior studies have explored text mining [18], deep learning-based error detection [19], and topic modeling [20], but these approaches rarely incorporate systematic hyperparameter optimization (HPO).

HPO is essential for improving machine learning performance [11]. Model-free approaches such as Grid Search and Random Search are widely used but suffer from scalability and non-adaptive search limitations, while methods such as Nelder–Mead [21] and evolutionary approaches such as DEoptim [22] face challenges in high-dimensional settings [23, 24, 25]. Model-based approaches, including Bayesian Optimization [12], TuRBO [26], mlrMBO [27], and AutoFT [28], improve efficiency through surrogate modeling, although they often introduce additional computational complexity.

Recent multi-fidelity strategies further accelerate HPO. Hyperband [29] leverages early stopping for efficient resource allocation, PriorBand [30] incorporates expert priors, and LLAMBO [31] integrates large language models into the optimization loop. However, these approaches are designed for general-purpose optimization and are not tailored to domain-specific recommendation tasks.

In parallel, recommendation systems have evolved from collaborative filtering to hybrid, context-aware, and deep learning-based approaches [32, 33]. Nevertheless, these systems typically focus on user-item interactions and rarely incorporate HPO or consider developer expertise levels. Existing efforts combining optimization and recommendation remain limited and often provide indirect outputs without personalized user-level recommendations [34, 35].

AutoML platforms such as Auto-sklearn [36], AutoGluon [37], FLAML [38], HyperOpt [39], and Optuna [40] have simplified model selection and tuning. Recent frameworks, including FairerML [41], DebiAI [42] and LAMDA-SSL [43] address domain-specific challenges. However, these tools are not designed to integrate hyperparameter optimization with personalized recommendation tasks.

Our previous work [44] introduced a web-based HPO platform for SO datasets but did not include recent model-based optimizers, statistical validation, or a recommendation module.

Table 1 summarizes the most relevant studies and highlights three key gaps: HPO has not been systematically applied to developer profiling on SO, recommendation systems rarely integrate HPO as a core component, and no prior work combines both model-free and model-based optimization with personalized recommendation.

To address these gaps, this study proposes HyperR, which integrates cost-aware hyperparameter optimization with a rule-based book recommendation module, providing a unified and scalable solution for developer-oriented learning systems.

3 Background

Hyperparameter optimization is a critical process in machine learning that directly affects model accuracy, generalization, and computational efficiency—especially in high-dimensional search spaces where effective optimization strategies are essential [45, 46]. This section presents the optimization methods evaluated in this study, followed by the proposed approach.

3.1 Model-free optimization methods

Model-free methods explore the hyperparameter space directly without surrogate models.

Grid Search discretizes each dimension of Λ into k_j values, yielding $|\Omega| = \prod_{j=1}^{d} k_j$ candidate configurations with

Table 1: Comparative analysis of state-of-the-art hyperparameter optimization, AutoML, and recommendation approaches

Study	Approach	Domain Dataset	HPO Method	Rec.	Cost	Early Stop	Eval. Metrics	Limitation
Bergstra & Bengio [11]	Random vs Grid Search	Synthetic, MNIST	Random Search	No	No	No	Accuracy, runtime	Non-adaptive search
Snoek et al. [12]	Bayesian Opt. (GP)	MNIST, CIFAR, LDA	GP + Expected Improvement	No	No	No	Validation error, runtime	High cost $O(t^3)$
Mantovani et al. [34]	Meta-learning tuning	OpenML (156 datasets)	Random Search + Nested CV	Yes	No	No	BAC, AUC	Limited to SVM; no rec.
Yang et al. [35]	AutoML via collab. filtering	OpenML + UCI	Matrix fact. + exp. design	Indirect	Yes	Yes	Error rate, regret, runtime	Model selection only
Feurer et al. [36]	Auto-sklearn	OpenML	SMAC + meta-learning	No	Yes	Yes	Accuracy, F1	General-purpose; no rec.
Eriksson et al. [26]	TuRBO	High-dim. benchmarks	Local GP + Thompson Sampling	No	Yes	Yes	Regret, convergence	Complex; no rec.
Mallik et al. [30]	PriorBand	HPOBench, LCBench	Hyperband + prior sampling	No	Yes	Yes	Regret, error rate, rank	Requires expert prior
Liu et al. [31]	LLAMBO	Bayesmark, HPOBench	BO + LLM warm-start	No	No	Yes	Regret, NRMSE, R^2	High cost; no rec.
He et al. [17]	PTM4Tag	Stack Overflow	—	Yes	No	No	P@k, R@k, F1@k	No HPO
This Study	HyperR	Stack Overflow (20K)	Multi-method + Proposed	Yes	Yes	Yes	Accuracy, F1, Precision, Recall, Specificity, P@3, R@3, runtime, memory	Rule-based recommendation; single domain

Rec. = Recommendation; Cost = Cost-aware; Eval. = Evaluation; rec. = recommendation; P@k = Precision@k; R@k = Recall@k

$O(k^d)$ complexity [47, 48]. Despite its exhaustive nature, it wastes evaluations on unpromising regions.

Random Search samples p configurations uniformly from Λ:

$$\{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(p)}\} \sim U(\Lambda) \tag{1}$$

With $O(p)$ complexity independent of d, it outperforms Grid Search in high-dimensional problems by allocating more trials to important hyperparameters [11].

Nelder–Mead [21] is a derivative-free simplex-based method that updates $d+1$ points via reflection, expansion, contraction, or shrinkage:

$$\lambda_{t+1} = \lambda_t + \alpha(\lambda_t - \lambda_{t-1}) \tag{2}$$

It is effective for small-scale problems but lacks global convergence guarantees in high-dimensional landscapes.

DEoptim [22] is a population-based stochastic algorithm that updates candidate vectors via mutation and crossover:

$$v_i^{(t)} = \lambda_{r1}^{(t)} + F \cdot \left(\lambda_{r2}^{(t)} - \lambda_{r3}^{(t)}\right) \tag{3}$$

$$u_{i,j}^{(t)} = \begin{cases} v_{i,j}^{(t)} & \text{if } \mathrm{rand}_j \le CR \text{ or } j = j_{\mathrm{rand}}, \\ \lambda_{i,j}^{(t)} & \text{otherwise.} \end{cases} \tag{4}$$

where F is the mutation factor, CR is the crossover probability, and $r1, r2, r3$ are distinct random indices.

3.2 Model-based optimization methods

Model-based methods construct surrogate models to guide exploration, reducing the number of evaluations.

Bayesian Optimization approximates the objective via a surrogate $\hat{J}(\lambda)$, typically Gaussian Processes [12] or Tree-structured Parzen Estimators [49], selecting the next candidate by maximizing an acquisition function:

$$\lambda_{t+1} = \arg\max_{\lambda \in \Lambda} \alpha(\lambda \mid D_{1:t}) \tag{5}$$

It achieves high sample efficiency but incurs $O(t^3)$ complexity for Gaussian Processes.

AutoFT [28] formulates HPO as a bi-level problem where the outer loop updates λ and the inner loop learns a robust objective L_{meta}:

$$\lambda_{t+1} = \lambda_t - \eta \nabla_\lambda L_{\text{meta}}(f_{\lambda_t}) \tag{6}$$

mlrMBO [27] extends Bayesian Optimization for multi-objective cases using weighted acquisition functions:

$$\lambda_{t+1} = \arg\max_{\lambda \in \Lambda} \sum_{m=1}^{M} w_m \, \alpha_m(\lambda \mid D_{1:t}) \tag{7}$$

It supports mixed discrete–continuous spaces with efficient surrogate modeling.

TuRBO [26] restricts the search to a local trust region $T_t \subset \Lambda$ with dynamically adapting radius:

$$\lambda_{t+1} = \arg\max_{\lambda \in T_t} \alpha(\lambda \mid D_{1:t}) \tag{8}$$

$$R_{t+1} = \begin{cases} \min(\gamma_{\text{inc}} R_t, R_{\max}) & \text{if improvement observed,} \\ \max(\gamma_{\text{dec}} R_t, R_{\min}) & \text{otherwise.} \end{cases} \tag{9}$$

3.3 Proposed hyperparameter optimization method

The proposed method is a cost-bounded and accuracy-aware framework that integrates execution cost limits and

accuracy-threshold–based early stopping to balance search completeness and computational efficiency. Let p be the total number of iterations and Λ the hyperparameter search space. An accuracy threshold z terminates the optimization early if:

$$\mathrm{Acc}(\lambda_t) \geq z \tag{10}$$

Theorem 1 (Execution cost bound). The execution cost per tuning step satisfies $t \geq p/5$.

Proof. Let the total execution cost be $T(p) = \sum_{i=1}^{p} t_i$ and define the reduced stopping point as $\zeta = p/\alpha$ with $\alpha \geq 5$. For feasibility, we require $t_i \geq T(p)/p$. Substituting $\alpha = 5$ yields the bound $t \geq p/5$. Empirical analysis confirms that $\alpha < 5$ leads to under-exploration ($\leq 10\%$ accuracy gain), whereas $\alpha > 5$ increases computation without substantial improvement.

Theorem 2 (Independence of accuracy threshold from iteration count). The accuracy threshold z does not exhibit a linear relationship with the iteration count p:

$$z \neq p \cdot x + c \tag{11}$$

Proof. Empirically, reducing p by a factor d may increase accuracy z, i.e., $z(p-d) > z(p)$. Differentiating yields $\partial z/\partial p \neq x$, confirming that z depends on the optimization dynamics at each step rather than on p linearly.

The proposed method achieves $O(p)$ complexity through the accuracy threshold z, which terminates the process when satisfactory performance is reached. The overall cost is:

$$C(p) = \sum_{i=1}^{p} t_i \cdot I(\mathrm{Accuracy}_i < z) \tag{12}$$

where I is an indicator function. The practical implementation, including the step size definition and bidirectional update mechanism, is presented in Algorithm 1.

4 Method

4.1 Overview

The proposed framework consists of four main phases, as illustrated in Figure 1:

1. **Labeling and Data Preparation:** The SO dataset is preprocessed using standard text mining techniques, and labels are generated for programming language and developer experience.
2. **Hyperparameter Optimization:** Multiple optimization methods are applied to machine learning models (SVM, RF, GBM, EPS, and EBR) to identify optimal hyperparameter configurations.
3. **Prediction and Recommendation:** Optimized models are used to predict developer experience levels and generate corresponding book recommendations.
4. **Web Tool Integration:** The HyperR platform integrates the optimization and recommendation processes into a unified and user-friendly system.

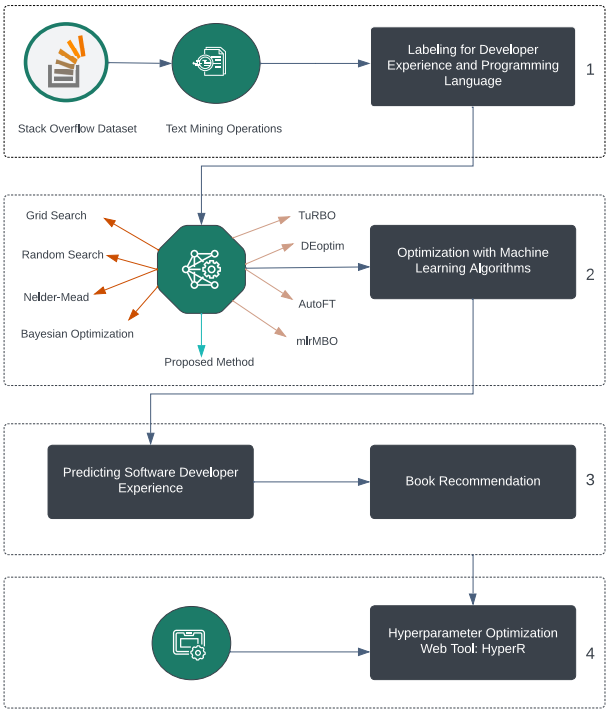


Figure 1: Overview of the proposed method

4.2 Research questions and evaluation criteria

To evaluate the proposed HyperR framework, the following research questions are defined:

- **RQ1:** How does the proposed hyperparameter optimization method improve classification performance compared to baseline approaches?
- **RQ2:** What is the computational cost of the proposed method in terms of runtime and memory usage?
- **RQ3:** How effectively does the system generate accurate and relevant book recommendations based on predicted developer profiles?
- **RQ4:** Are the performance improvements achieved by the proposed method statistically significant compared to baseline approaches?

To answer these questions, the framework is evaluated by comparing its performance against baseline methods using classification metrics including accuracy, precision, recall, specificity, and F1-score, along with computational efficiency based on runtime and memory usage and recommendation quality assessed using Precision@3 and Recall@3.

4.3 Dataset

This study employs the publicly available SO corpus, which provides a large-scale repository of question–answer data for research purposes [50]. The full dataset contains more

than 1,048,575 instances, each including key attributes such as ID, user ID, creation date, closure date, score, title, and body. In addition, tag data represent user-assigned labels indicating programming languages or relevant technical topics, enabling structured categorization of the corpus.

From this large dataset, a labeled subset of 20,000 instances was first constructed, where programming language and developer experience were assigned using text mining and rule-based labeling techniques. To ensure computational feasibility during hyperparameter optimization, a balanced subset of 3,000 instances was selected from this labeled data. This design choice was motivated by the experimental setup, which involves nine optimization methods applied to five classifiers, each repeated 20 times with up to 50 iterations per run, resulting in thousands of model evaluations. Applying this procedure to the full 20,000-instance dataset would introduce substantial computational overhead, particularly for classifiers such as SVM and GBM, which exhibit superlinear scaling with dataset size.

The 3,000-instance subset was constructed using stratified sampling to preserve the original class distribution across more than twenty programming languages and multiple developer experience levels. This ensures that the subset remains representative of the larger labeled dataset in terms of both label diversity and class balance.

To assess whether the results generalize beyond the selected subset, additional experiments were conducted on datasets of varying sizes (500, 1,000, and 3,000 instances). The results demonstrated consistent performance trends across all sizes, with less than 5% variation in macro-averaged precision. The proposed method maintained stable performance across all subsets, while some baseline methods (e.g., Nelder–Mead) exhibited greater sensitivity to dataset size. These findings provide strong evidence that the observed performance differences are primarily driven by the optimization strategies rather than dataset-specific characteristics, supporting the generalizability of the results.

4.4 Experimental protocol

To ensure reproducibility and transparency, all experiments were conducted under a standardized protocol.

The dataset, consisting of 3000 labeled instances, was divided into training and test sets using a stratified split, with 70% allocated for training and 30% for testing. Hyperparameter optimization was performed on the training set, and model performance was evaluated on the test set. No separate validation set was used in order to maintain a consistent and computationally efficient evaluation framework across all optimization methods.

A fixed random seed (`seed`=42) was used throughout all experiments to ensure consistency.

Hyperparameter optimization was carried out using multiple optimizers, including Grid Search, Random Search, Bayesian Optimization, Nelder–Mead, TuRBO, DEoptim, AutoFT, mlrMBO, and the proposed method. For each optimizer, the search space was defined according to the parameter ranges and step sizes provided in Table 3. The iteration budget p and stopping threshold z were kept identical across all methods.

All optimization methods were executed under the same computational budget to ensure a fair comparison. Each method was allowed the same number of iterations, identical search space definitions, and equivalent stopping criteria. No method was granted additional evaluations or extended runtime beyond the predefined limits.

Each experimental configuration was repeated 20 times, and the reported results correspond to the average performance across runs. Evaluation metrics include accuracy, precision, recall, specificity, and F1-score for classification tasks, as well as Precision@3 and Recall@3 for recommendation performance. Computational efficiency was assessed using runtime and peak memory consumption.

All experiments were implemented using R (version 4.2.1) and Python (version 3.13). The primary R packages include `caret`, `e1071`, `randomForest`, and `gbm`.

The experiments were conducted on a system equipped with an Intel Core i7-14700 CPU (2.10 GHz) and 32 GB RAM, running a 64-bit Windows Server 2019 operating system.

The dataset, preprocessing procedures, hyperparameter configurations, and full implementation are publicly available in the project repository, enabling exact reproduction of the reported results.

4.5 Text mining operations

To prepare the dataset for machine learning tasks, a systematic text mining pipeline was applied to the question titles and bodies from the SO corpus. The preprocessing workflow consisted of tokenization [51], lowercase normalization, noise removal (punctuation, digits, hyperlinks) [52], stopword removal, and stemming with lemmatization [53].

After preprocessing, feature extraction was performed using the Term Frequency–Inverse Document Frequency (TF–IDF) weighting scheme [54], where the weight of a term t in document d is computed as:

$$w_{t,d} = TF(t,d) \times IDF(t,D) \tag{13}$$

with $TF(t,d) = f_{t,d} / \sum_{t' \in d} f_{t',d}$ and $IDF(t,D) = |D|/(1 + |\{d \in D : t \in d\}|)$. This ensures that terms frequent in a specific document but rare across the corpus receive higher discriminative importance [55]. In addition to unigrams, bigram features were extracted to capture short-range contextual dependencies [56]. The analysis was conducted directly on the TF–IDF matrix without applying dimensionality reduction [57].

4.6 Labeling process

The dataset was gradually expanded to approximately 20,000 entries, with each configuration repeated at least 20 times for statistical reliability.

Table 2: An example of the labeling matrix generated from the raw question dataset

y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}	y_{16}	y_{17}	y_{18}	y_{19}	y_{20}	y_{21}	y_{22}	y_{23}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	22
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	41
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	51
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	43
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	21
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	51

Each SO question $q_i = (\text{title}_i, \text{body}_i)$ was cleaned and transformed into a vector $q_i = (t_{i1}, \ldots, t_{im})$ with additional features (character count c_i, word count w_i), yielding an enriched representation $\tilde{x}_i = (x_i, c_i, w_i)$.

The label vector for each instance was defined as $y_i = (y_{i1}, \ldots, y_{i23})$. The first 21 dimensions encode programming language tags ($y_{ij} = 1$ if $q_i \in L_j$, 0 otherwise), covering 21 languages selected based on their consistent SO usage over 14 years [58]:

$$L = \left\{ \begin{array}{l} \text{JavaScript, SQL, Java, C\#, Python,} \\ \text{C++,C, PHP, Ruby, Swift,Objective-C,} \\ \text{VB.NET,Perl,Bash, CSS, Scala, HTML,} \\ \text{Lua,Haskell,Markdown, R} \end{array} \right\}$$

The 22nd label (y_{i22}) represents user experience level: 0 (unspecified), 1 (beginner), 2 (intermediate), or 3 (advanced). Experience labels were assigned using a rule-based keyword mechanism derived from domain knowledge and empirical analysis of SO data. Basic constructs (e.g., variable, array, function) indicate beginner level; terms such as threads, JSON, and encryption characterize intermediate level; and system-level concepts (e.g., REST, docker, kubernetes) are linked to advanced expertise. The dominant keyword group determines the final label. Complete keyword lists are available in the project repository.

The 23rd label (y_{i23}) corresponds to book recommendations, defined as:

$$y_{i23} = f(L_j, y_{i22}), \quad f : L \times \{0, 1, 2, 3\} \to \mathbb{N} \tag{14}$$

where f is a deterministic lookup mapping each language–experience pair to a predefined set of book identifiers.

To refine recommendations, Amazon ratings (r_{avg}) were incorporated [59]:

$$\hat{y}_{i23} = \arg\max_{b \in B} \left(f(L_j, y_{i22}) + \alpha \cdot r_{\text{avg}}(b) \right) \tag{15}$$

where B is the candidate book set and $\alpha = 0.2$ ensures that the rule-based mapping remains the primary driver while incorporating external ratings in a complementary manner.

The structured labeling process results in a multi-label encoding where each instance is represented as $(q_i, \tilde{x}_i, y_i)$. Table 2 illustrates this mapping.

4.7 The proposed algorithm

Algorithm 1 Heuristic hyperparameter optimization algorithm

Input: dataset, evaluation metric $\varphi(\cdot)$, bounds $[y, x]$, p, v, z
Output: maxAccuracy, λ^*

1: optimizationValue $\leftarrow \{y, y + v, y + 2v, \ldots, x\} \subseteq \Lambda$
2: $\lambda(1) \leftarrow$ optimizationValue[1]
3: accuracy $\leftarrow \varphi(\lambda(1))$
4: accuracyOld $\leftarrow$ accuracy
5: maxAccuracy $\leftarrow$ accuracy
6: $\lambda^* \leftarrow \lambda(1)$
7: $\varepsilon \leftarrow 10^{-3}$
8: $i \leftarrow 1$
9: **while** (accuracy $< z$) **AND** ($i < p$) **do**
10: **if** (accuracy $\geq$ accuracyOld + ε) **then**
11: $\lambda(i+1) \leftarrow \lambda(i) + v$
12: **else**
13: $\lambda(i+1) \leftarrow \lambda(i) - v$
14: **end if**
15: **if** $\lambda(i+1) < y$ **then** $\lambda(i+1) \leftarrow y$
16: **if** $\lambda(i+1) > x$ **then** $\lambda(i+1) \leftarrow x$
17: accuracyOld $\leftarrow$ accuracy
18: accuracy $\leftarrow \varphi(\lambda(i+1))$
19: **if** accuracy $\geq$ maxAccuracy **then**
20: maxAccuracy $\leftarrow$ accuracy
21: $\lambda^* \leftarrow \lambda(i+1)$
22: **end if**
23: $i \leftarrow i + 1$
24: **end while**
25: **return** $(\lambda^*, \text{maxAccuracy})$

The proposed algorithm is presented in Algorithm 1 and implements the theoretical framework described in the Background section by defining a cost-bounded and accuracy-aware procedure for hyperparameter optimization.

In the initialization phase (Steps 1–8), the search interval $[y, x]$ is discretized using step size v, producing candidate configurations $\{y, y + v, y + 2v, \ldots, x\} \subseteq \Lambda$. The first element λ_1 is evaluated, and its accuracy $\varphi(\lambda_1)$ initializes both the current and best accuracy values. An improvement

tolerance $\varepsilon = 10^{-3}$ is set to avoid sensitivity to noise.

The main loop (Step 9) continues while accuracy remains below threshold z and iteration count has not exceeded budget p. The core mechanism is the bidirectional update (Steps 10–14): if accuracy improves beyond ε, the search continues in the same direction ($+v$); otherwise, the direction reverses ($-v$). Boundary checks (Steps 15–16) ensure all candidates remain within $[y, x]$. The best configuration λ^* is retained throughout (Steps 19–22).

This bidirectional update distinguishes the proposed method from traditional line search (single direction) and coordinate descent (one parameter at a time). By integrating tolerance-based direction switching, cost-bounding constraints, and accuracy-threshold early stopping, the algorithm provides a robust and efficient solution that scales linearly with the iteration budget.

4.8 Web tool: HyperR

All experiments were executed through HyperR, a web-based platform that centralizes dataset selection, algorithm specification, optimizer choice, and resource monitoring, ensuring methodological consistency and reproducibility across runs.

The data flow, illustrated in Figure 2, consists of two interconnected pipelines. In the upper pipeline, a user-submitted SO question (title and body) is processed through text processing (tokenization, stopword removal, stemming, lemmatization) and feature extraction (TF–IDF vectorization with unigrams, bigrams, word count, and character count). The resulting features are passed to the model prediction module, which performs two tasks: predicting the programming language (y_1–y_{21}) and predicting the developer experience level (y_{22}). Based on these predictions, the recommendation module applies the rule-based mapping combined with external ratings to generate top-3 book recommendations.

The lower pipeline represents the hyperparameter optimization loop. Starting from a predefined search space (bounds, step size v, threshold z, budget p), the optimization process iteratively evaluates candidate configurations across classifiers (SVM, RF, GBM, EPS, EBR) and computes $\varphi(\lambda)$. The best model selection stage identifies the optimal configuration based on maximum accuracy, while logging runtime and memory usage. The selected model is then fed into the upper pipeline for inference.

The system is implemented on `ASP.NET Core MVC` with a layered architecture. The backend dispatches jobs to R scripts (caret [60], e1071 [61], gbm [62], randomForest [65], rBayesianOptimization [66]) via the .NET `System.Diagnostics.Process` class. Each experiment is assigned a unique Experiment ID, generated as a hash of dataset, optimizer, and timestamp, guaranteeing traceability. The R backend computes $\varphi(\lambda)$ for each candidate configuration and logs execution time and memory usage before returning results to the web tier.

The system interface and outputs are illustrated in Figure 3. Panel (a) presents detailed tabular optimization results, including accuracy values across iterations, the best hyperparameter configuration, execution time in seconds, and memory usage in megabytes. Panel (b) visualizes performance trends, where the x-axis represents hyperparameter values and the y-axis denotes accuracy, along with summary statistics such as mean and standard deviation. Across repeated runs, these are computed as:

$$\mathrm{MeanAcc} \;=\; \frac{1}{p}\sum_{i=1}^{p}\varphi(\lambda_i) \tag{16}$$

$$\mathrm{SD} \;=\; \sqrt{\frac{1}{p}\sum_{i=1}^{p}\big(\varphi(\lambda_i)-\mathrm{MeanAcc}\big)^2} \tag{17}$$

Once optimization converges, the best model is stored under the corresponding Experiment ID. When a developer submits a query, HyperR applies the optimized model to predict both the programming language and experience level, then dynamically recommends books matched to the predicted profile. The recommendation interface is depicted in Figure 4(a). As illustrated, the system analyzes a user-submitted query, predicts the associated programming language and experience level, and generates a tailored set of book recommendations. In the example shown, the system identifies the query as Python-related with a beginner-level profile and provides three introductory books aligned with this prediction.

The code execution interface (Figure 4b) enables users to submit R scripts directly within the web-based environment and observe outputs in real time, with input and results displayed side by side for transparency and reproducibility. All results are stored in CSV format and can be exported in multiple formats (PNG, JPG, PDF) to support reporting.

4.9 Machine learning algorithms

Hyperparameter optimization is essential for selecting the best parameter configuration in machine learning models, as it significantly improves overall predictive performance. Manual tuning requires testing numerous combinations, which is computationally expensive and inefficient. To address this challenge, automated search techniques are employed. In this study, multiple algorithms with optimized hyperparameters were evaluated under both single-label and multi-label classification settings. In single-label tasks, only the developer's experience level (y_{22}) was predicted, while in multi-label tasks, programming language labels (y_1–y_{21}) were predicted simultaneously with y_{22}.

SVM are widely used for classification and rely on hyperparameters such as kernel type, cost, degree, and gamma, which directly affect model complexity and generalization [47]. Different kernels, including linear, polynomial, sigmoid, and Radial Basis Function (RBF), can yield substantially different performance outcomes.

RF is an ensemble learning method for classification and regression that combine multiple decision trees. Its key hy-

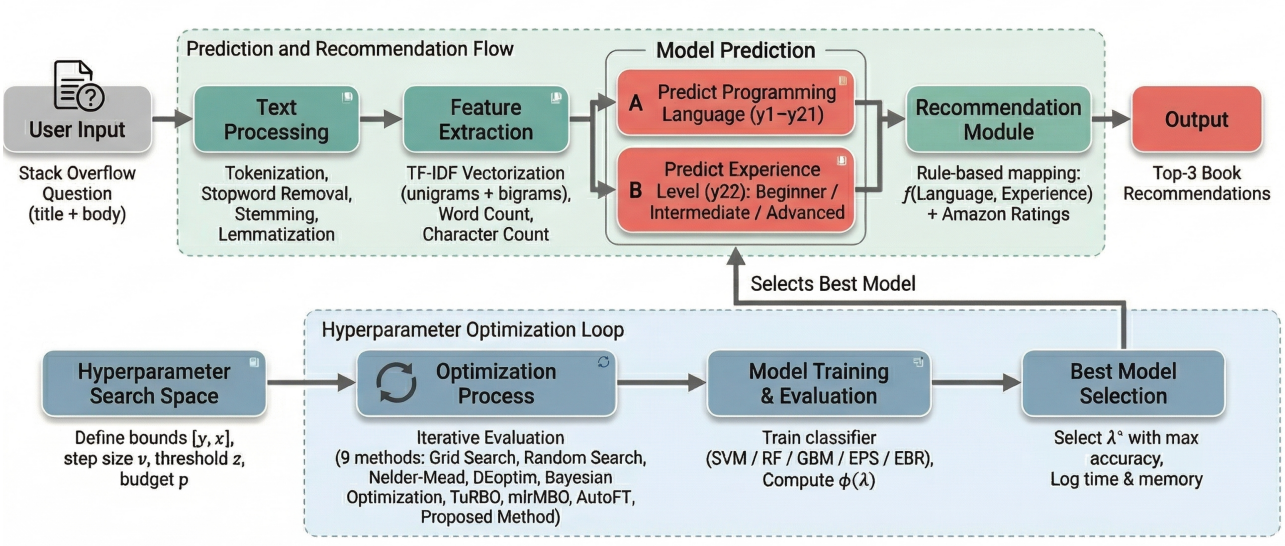


Figure 2: HyperR workflow and data flow architecture showing the prediction/recommendation pipeline (top) and the hyperparameter optimization loop (bottom)

Table 3: Hyperparameters employed in the experiments

Algorithm	Parameter	Description	Range	Step	Type	Target
SVM	kernel	Kernel type	–	–	Single	y_{22}
	degree	Polynomial exponent	1–10	1		
	gamma	Kernel coefficient	2^{-10}–2^{10}	0.1		
	cost	Error penalty	2^{-10}–2^{10}	1		
RF	mtry	Features per split	1–8	1	Single	y_{22}
	ntree	Number of trees	100–500	1		
GBM	shrinkage	Learning rate	0.001–1	0.1	Single	y_{22}
	n.trees	Number of trees	100–500	1		
	n.minobsinnode	Min. node samples	5–15	1		
EPS	subsample	Instance rate	0.1–1	0.1	Multi	y_1–y_{22}
	strategy	Cluster strategy	A–B	–		
	b	Strategy parameter	1–4	1		
	p	Pruned instances	1–4	1		
EBR	subsample	Instance rate	0.1–1	0.1	Multi	y_1–y_{22}
	attr.space	Feature rate	0.1–1	1		

perparameters include `mtry` (the number of features considered at each split) and `ntree` (the number of trees in the forest). Increasing the number of trees typically enhances information retention but also raises computational cost [67].

GBM build predictive models by iteratively combining weak learners to minimize residual errors. Key hyperparameters include `shrinkage` (learning rate), `n.trees` (number of boosting iterations), and `n.minobsinnode` (minimum observations per terminal node). These parameters balance overfitting risk and robustness against noise [62].

For multi-label classification, two algorithms were employed: EPS and EBR. EPS is designed to handle sparse label distributions by pruning less frequent label combinations and optimizing clustering strategies [63, 64]. EBR, on the other hand, decomposes the multi-label task into independent binary classification problems for each label and aggregates the results. Both methods rely on hyperparameters such as subsample rate, strategy, b, p (for EPS), and `attr.space` (for EBR), which were tuned to improve classification accuracy.

Each algorithm was selected based on its ability to handle complex datasets and improve prediction accuracy. Hyperparameters were carefully fine-tuned to ensure reliable experimental results. The use of RF and multi-label classifiers such as EPS and EBR allowed for diversified and improved model performance while balancing computational efficiency.

The hyperparameters used in the experiments, along with their descriptions, ranges, and step sizes, are summarized in Table 3. These values were determined based on prior literature to ensure the scientific validity and comparability of the results.

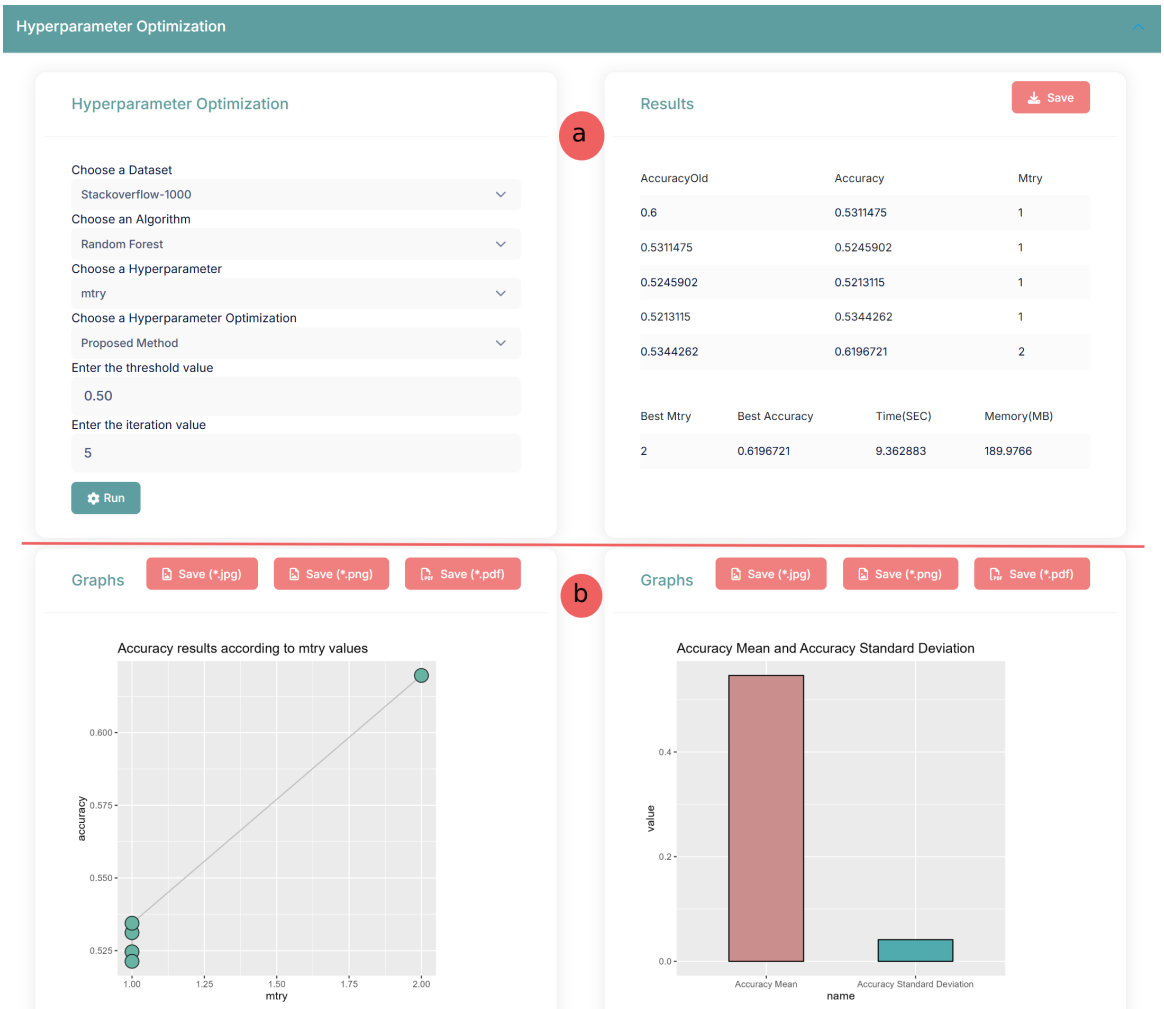


Figure 3: System interface and outputs

5 Results

5.1 RQ1: Classification performance comparison

Table 4 summarizes the macro-averaged classification metrics obtained by each optimization method.

Table 4: Macro-averaged classification metrics by optimization method

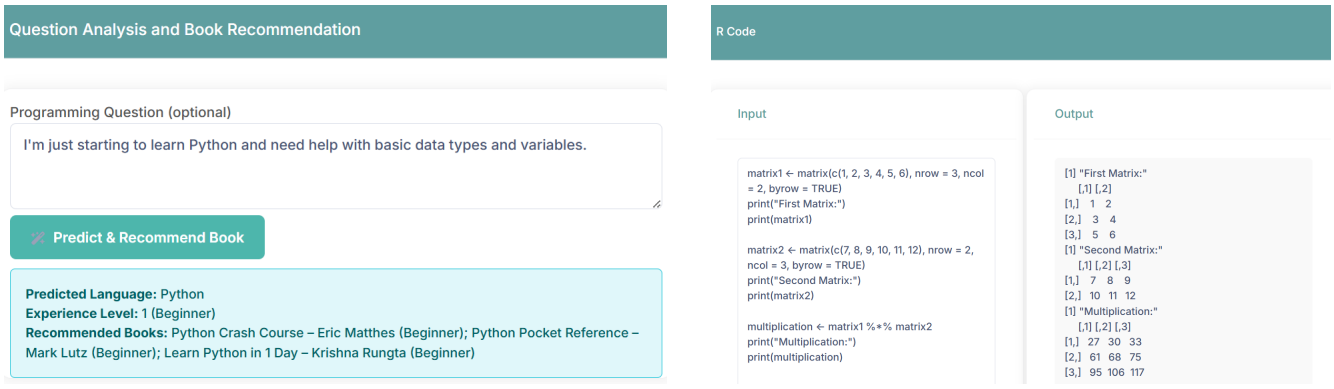
Method	F1	Precision	Recall	Specificity
Grid Search	0.486	0.618	0.472	0.880
Random Search	0.485	0.635	0.486	0.879
Nelder–Mead	0.404	0.525	0.406	0.839
DEoptim	0.508	0.660	0.490	0.885
Bayes Opt.	0.483	0.603	0.478	0.870
AutoFT	0.495	0.620	0.490	0.875
mlrMBO	0.507	0.640	0.492	0.883
TuRBO	0.515	0.645	0.500	0.882
Proposed Method	0.504	0.671	0.479	0.884

Among the classical model-free approaches, Grid Search and Random Search yield moderate results, with Random Search achieving slightly higher recall (0.486 vs. 0.472). Nelder–Mead clearly underperforms, recording the lowest F1-score (0.404) and specificity (0.839). The population-based DEoptim exhibits the strongest precision among model-free methods (0.660) with competitive F1 (0.508) and specificity (0.885).

Model-based methods provide more stable outcomes. TuRBO achieves the highest overall F1-score (0.515), reflecting a balanced precision-recall trade-off, while mlrMBO follows closely with consistent scores across all measures. AutoFT slightly surpasses Bayesian Optimization in recall (0.490 vs. 0.478).

The proposed method stands out with the highest precision (0.671), improving by approximately 13–18% over classical baselines. Although its recall (0.479) does not surpass TuRBO or mlrMBO, its F1-score (0.504) and specificity (0.884) remain competitive, highlighting its strength in minimizing false positives.

Convergence analyses based on iteration-wise accuracy trends (Figures 5 and 6) reveal clear differences between optimization strategies. In Figure 5, the proposed method demonstrates more stable and faster convergence compared



(a) Recommendation interface (b) Code execution interface

Figure 4: User interfaces of the HyperR system

to classical model-free approaches, outperforming Nelder–Mead and maintaining a clear advantage over Grid and Random Search across all classifiers. However, DEoptim may reach comparable accuracy levels in certain cases.

In Figure 6, advanced model-based methods such as TuRBO and mlrMBO exhibit strong convergence behavior. While the proposed method achieves comparable performance, TuRBO occasionally reaches similar accuracy levels in fewer iterations, reflecting its efficient local search capability.

Scalability analysis across dataset sizes (500, 1000, and 3000 instances) provides additional insights. On smaller datasets, performance differences are more pronounced, with the proposed method maintaining stable precision while methods such as Nelder–Mead deteriorate significantly. At medium size (1000), most methods converge toward similar levels, yet the proposed method maintains its precision advantage. On the largest dataset (3000), performance differences narrow, with TuRBO achieving the highest F1-score while the proposed method remains competitive across all metrics.

These results indicate that the proposed method provides a balanced optimization trajectory, combining stability, robustness under limited data, and competitive convergence. In line with state-of-the-art approaches (Table 1), advanced methods such as TuRBO and AutoFT achieve strong performance but at the cost of increased complexity. The proposed method offers competitive accuracy with a simpler and more efficient strategy, highlighting its practical suitability in scenarios where both performance and efficiency are critical.

5.2 RQ2: Computational efficiency analysis

To evaluate the computational cost of different optimization strategies, runtime and memory usage are analyzed across all classifiers.

Figure 7(a) presents the average runtime of the optimization methods, where the x-axis represents the classifiers (EBR, EPS, GBM, RF, and SVM) and the y-axis denotes execution time in seconds. As observed, classical approaches such as Grid Search and DEoptim exhibit the highest computational cost due to their exhaustive and population-based search mechanisms, which require a large number of evaluations. In contrast, model-based methods, including Bayesian Optimization, AutoFT, mlrMBO, and TuRBO, generally achieve lower runtime by guiding the search toward promising regions of the search space.

The proposed method demonstrates a balanced runtime profile across all classifiers. It consistently requires less time than exhaustive approaches while remaining competitive with advanced model-based optimizers. This indicates that the cost-bounded search strategy and early stopping mechanism effectively reduce unnecessary evaluations without sacrificing optimization quality.

Figure 7(b) illustrates the average memory consumption of the optimization methods, where memory usage is measured in megabytes (MB). Similar to runtime behavior, classical methods tend to consume more memory due to extensive exploration of the search space. Model-based approaches show improved memory efficiency by focusing on a smaller set of candidate configurations.

The proposed method maintains a stable and moderate memory footprint across all classifiers, avoiding the extreme values observed in some baseline methods (e.g., TuRBO and DEoptim). This stability indicates that the method provides efficient resource utilization without introducing significant overhead.

From a comparative perspective, model-based approaches such as TuRBO and mlrMBO achieve strong efficiency through surrogate modeling and adaptive sampling strategies. However, these methods often involve higher algorithmic complexity and implementation overhead. In contrast, the proposed method offers competitive computational performance while maintaining a simpler and more interpretable optimization process.

The results demonstrate that the proposed method achieves a favorable trade-off between computational cost and predictive performance, making it suitable for practical applications where both efficiency and scalability are criti-

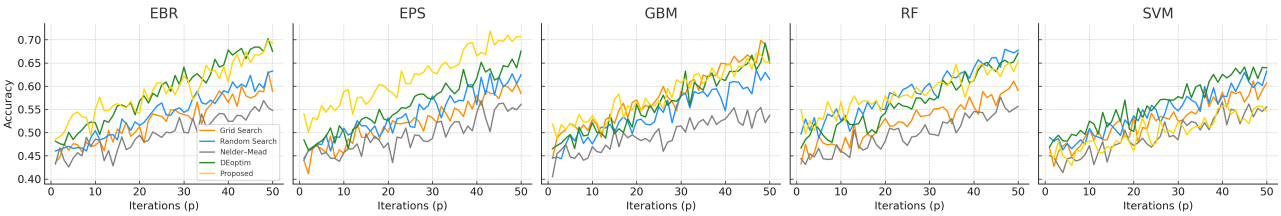


Figure 5: Convergence comparison of the proposed method against classical model-free optimization methods.

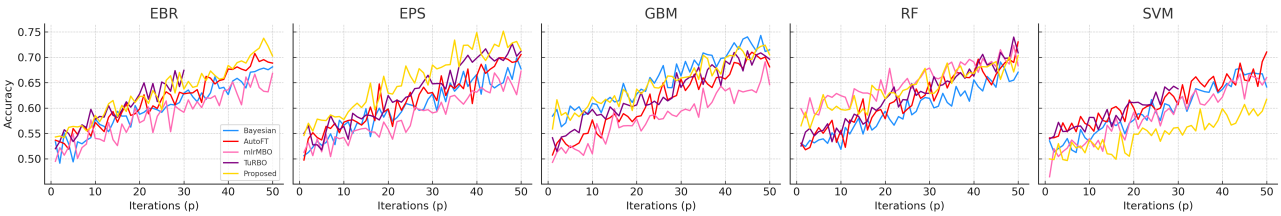


Figure 6: Convergence comparison of the proposed method against advanced model-based optimization methods.

cal.

5.3 RQ3: Recommendation performance evaluation

To evaluate the recommendation module, Precision@k and Recall@k metrics are used with $k = 3$. For each test instance i, the system generates a recommended list R_i of three books, while the ground-truth relevant set G_i is constructed based on the books associated with the corresponding programming language and experience level. Metrics are computed as:

$$\text{Precision@}k = \frac{1}{N}\sum_{i=1}^{N}\frac{|R_i \cap G_i|}{k}, \tag{18}$$

$$\text{Recall@}k = \frac{1}{N}\sum_{i=1}^{N}\frac{|R_i \cap G_i|}{|G_i|}. \tag{19}$$

The recommendation list is constructed by restricting the candidate pool to books matching the predicted programming language and experience level. All evaluation metrics are computed exclusively on the held-out test set, while hyperparameter tuning and early stopping rely only on training data to prevent data leakage.

Under this protocol, the system achieves a Precision@3 of 0.82 and a Recall@3 of 0.77, indicating that the recommended lists are both accurate and highly relevant to the user profile. Consistent with RQ1, optimization methods achieving higher classification accuracy tend to yield better recommendation performance. However, the proposed method does not uniformly outperform all alternatives; in certain cases, optimizers such as TuRBO or DEoptim may achieve comparable results, supporting the interpretation that the proposed method provides a balanced trade-off between predictive performance and computational efficiency.

To further assess sensitivity to variations in query complexity and expertise, a qualitative evaluation was conducted using representative queries across three experience levels and multiple programming languages (Table 5). The results indicate that the system consistently identifies the programming language and assigns appropriate experience levels, producing level-specific book recommendations. Beginner-level queries are associated with introductory resources, intermediate queries yield mid-level references, and advanced queries involving specialized concepts such as REST APIs, stored procedures, and containerization technologies are matched with advanced-level materials.

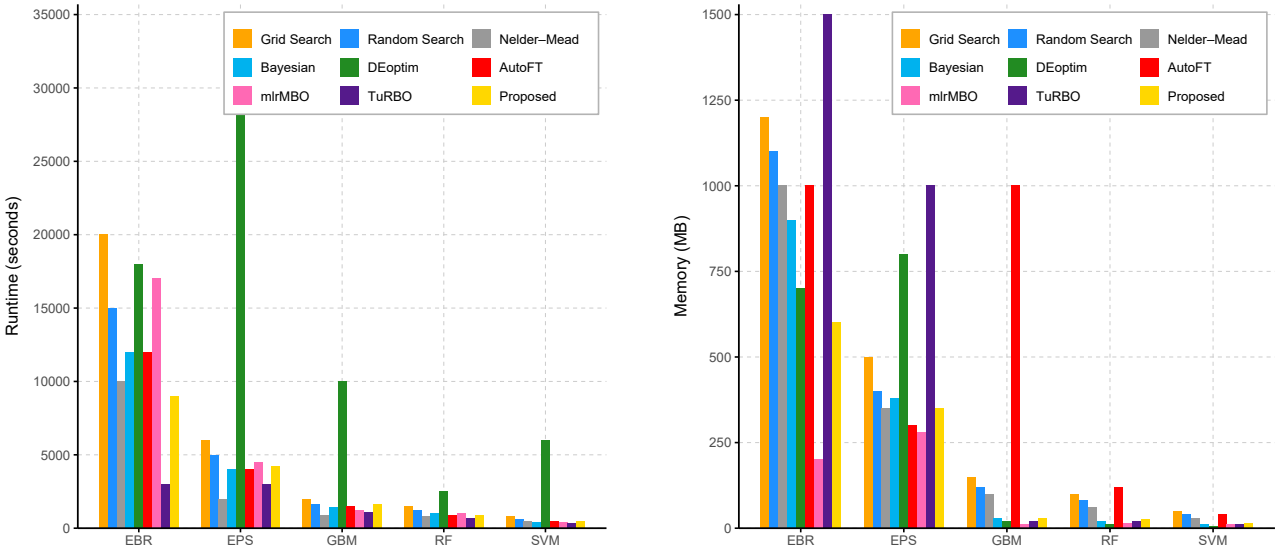
These findings confirm that the recommendation module is responsive to variations in query content and developer expertise. Nevertheless, its effectiveness depends on the coverage of the keyword-based labeling scheme, and performance may degrade for queries involving emerging or ambiguous terminology.

5.4 RQ4: Statistical significance and comparative analysis

To assess whether the observed performance differences are statistically significant, the Wilcoxon signed-rank test is applied to compare the proposed method with alternative optimization strategies in terms of F1-score.

Table 6 summarizes the pairwise Wilcoxon signed-rank test results, reporting the test statistic, original p-values, and Bonferroni-adjusted p-values ($k = 8$, adjusted values capped at 1.0).

The results indicate that the proposed method achieves statistically significant improvements over Grid Search, Random Search, Nelder–Mead, Bayesian Optimization,



(a) Average runtime across optimization methods

(b) Average memory usage across optimization methods

Figure 7: Computational efficiency comparison of optimization methods, where (a) shows average runtime in seconds and (b) shows average memory usage in MB across different classifiers.

and AutoFT (adjusted $p < 0.008$), confirming that the observed gains reflect consistent improvements rather than random variation.

In contrast, no statistically significant differences are observed with DEoptim, mlrMBO, and TuRBO (adjusted $p = 1.000$), suggesting that the proposed method remains competitive with state-of-the-art approaches without requiring the same level of computational complexity.

These findings support the interpretation that the proposed method offers a robust alternative to both classical and advanced strategies, achieving a balanced performance profile that combines competitive accuracy with improved computational efficiency. The observed significance remains consistent after Bonferroni correction across different classifiers and dataset sizes, indicating that the gains are stable and generalizable.

5.5 Ablation study

To further analyze the contribution of individual components of the proposed optimization method, an ablation study was conducted. Specifically, three key mechanisms were examined: the cost-bounding strategy, the accuracy threshold condition, and the bidirectional update rule. Each component was selectively removed to evaluate its impact on model performance.

The analysis was primarily performed on the Random Forest and GBM models, as these provide stable and interpretable environments for assessing optimization behavior. The results are summarized in Table 7.

The results indicate that the proposed optimization framework generally improves performance compared to the baseline configuration, with the most notable gains observed for the RF model. The substantial drop in baseline accuracy highlights the effectiveness of the overall optimization strategy.

For the RF model, removing individual components leads to noticeable performance variations, indicating a high sensitivity to adaptive optimization mechanisms. In contrast, the GBM model exhibits relatively stable behavior, with only minor performance changes when individual components are removed, suggesting that it benefits primarily from the overall optimization structure rather than specific mechanisms.

The SVM model shows a different pattern. Although removing individual components typically reduces performance, the baseline configuration achieves higher accuracy than the proposed method. This suggests that SVM is highly sensitive to specific hyperparameter settings and that the adaptive strategy may occasionally converge to suboptimal regions.

Similar to GBM, the EPS and EBR models demonstrate relatively stable behavior. The impact of removing individual components is limited, and in some cases, simplified configurations yield comparable or slightly improved performance. This indicates that, for ensemble-based models, the overall optimization framework contributes more to performance than any single component.

These findings suggest that the effectiveness of the proposed method is model-dependent. Greater improvements are observed in models that are highly sensitive to hyperparameter tuning, while more stable models benefit from the general optimization structure without relying heavily on individual components.

Table 5: Qualitative sensitivity analysis of the recommendation module across different query types and expertise levels

Query	Language	Experience	Recommended Books
Beginner-level queries			
What is a variable in python?	Python	1	Python Crash Course – E. Matthes; Python Pocket Reference – M. Lutz; Learn Python in 1 Day – K. Rungta
How to create an array in javascript?	JavaScript	1	Learn JavaScript Visually – I. Demirov; JavaScript: The Definitive Guide – D. Flanagan; Beginning JavaScript – P. Wilton
What is a function in java?	Java	1	Beginning Programming with Java For Dummies – B. Burd; Java: Programming Basics for Absolute Beginners – N. Clark; Head First Java – K. Sierra, B. Bates
Intermediate-level queries			
How do I use class in python?	Python	2	Fluent Python – L. Ramalho; Effective Python – B. Slatkin; Programming Python – M. Lutz
How to use threads in java?	Java	2	Effective Java (3rd Ed.) – J. Bloch; Java: The Complete Reference – H. Schildt; Thinking in Java – B. Eckel
How to handle JSON in javascript?	JavaScript	2	Eloquent JavaScript – M. Haverbeke; A Smarter Way to Learn JavaScript – M. Myers; JavaScript & jQuery – J. Duckett
Advanced-level queries			
How to implement REST API with services in python?	Python	3	Python Cookbook – D. Beazley, B.K. Jones; Advanced Python Programming – G. Lanaro et al.; Learn Python 3 The Hard Way – Z.A. Shaw
How to use stored procedures in SQL?	SQL	3	SQL Cookbook – A. Molinaro; SQL For Smarties – J. Celko; The Art of SQL – S. Faroult, P. Robson
How to deploy with docker and kubernetes in java?	Java	3	Well-grounded Java Developer – B. Evans et al.; Java Concurrency in Practice – B. Goetz et al.; Optimizing Java – B.J. Evans et al.

Table 6: Wilcoxon signed-rank test results comparing the proposed method with other optimizers (F1-score). Bonferroni-corrected p-values are reported for multiple comparisons ($k = 8$)

Method	Statistic	p-value	Adj. p-value
Grid Search	3.0	< 0.001	< 0.008
Random Search	0.0	< 0.001	< 0.008
Nelder–Mead	0.0	< 0.001	< 0.008
DEoptim	182.0	0.309	1.000
Bayes Opt.	0.0	< 0.001	< 0.008
AutoFT	36.0	< 0.001	< 0.008
mlrMBO	167.0	0.184	1.000
TuRBO	175.0	0.245	1.000

Table 7: Ablation study evaluating the impact of cost-bounding, accuracy threshold, and bidirectional update components

Model	Scenario	Accuracy	Δ Accuracy
RF	Full Model	0.6378	0.0000
RF	No Cost-Bounding	0.6389	+0.0011
RF	No Threshold	0.6411	+0.0033
RF	No Update	0.6433	+0.0056
RF	Baseline	0.5222	−0.1156
GBM	Full Model	0.6467	0.0000
GBM	No Cost-Bounding	0.6456	−0.0011
GBM	No Threshold	0.6467	0.0000
GBM	No Update	0.6433	−0.0033
GBM	Baseline	0.6133	−0.0333

6 Discussion

The experimental results provide a comprehensive comparison of classical and modern hyperparameter optimization strategies in the context of developer profiling and recommendation. Classical model-free approaches, including Grid Search, Random Search, and Nelder–Mead, exhibit limited scalability and unstable performance across classifiers, with their lack of guided exploration leading to inefficient search and increased computational cost. Among these, DEoptim performs relatively better due to its population-based mechanism, though its slower convergence results in higher runtime.

In contrast, model-based approaches such as mlrMBO, TuRBO, and AutoFT demonstrate more stable convergence behavior. TuRBO achieves the highest F1-score through effective local search, while mlrMBO provides consistent performance across multiple metrics. However, these methods rely on surrogate modeling, which introduces additional computational overhead.

Table 8 provides an analytical comparison of all evaluated methods in terms of performance, efficiency, and stability. Compared to classical methods, the proposed approach achieves higher precision and improved efficiency, and remains competitive with model-based optimizers while avoiding their computational overhead.

The effectiveness of the proposed method can be attributed to three key design mechanisms: (i) a cost-bounded search strategy that limits unnecessary exploration, (ii) an accuracy-threshold-based early stopping criterion that reduces computational overhead, and (iii) a tolerance-driven bidirectional update that enables flexible exploration and helps avoid suboptimal solutions. These mechanisms collectively explain the observed performance, where the proposed method achieves the highest macro-averaged precision (0.671) while maintaining competitive F1-score and specificity.

From a methodological perspective, the novelty lies in integrating bidirectional updates, cost constraints, and early stopping within a unified black-box optimization framework. This design enables an explicit balance between search efficiency and performance without relying on com-

Table 8: Analytical comparison of hyperparameter optimization methods based on experimental observations across performance and efficiency criteria

Method	Precision	F1	Efficiency	Stability	Key Insight
Grid Search	Medium	Medium	Very Low	Low	High computational cost with limited improvement despite exhaustive search.
Random Search	Medium	Medium	Medium	Medium	Provides balanced performance but lacks convergence consistency.
Nelder–Mead	Low	Low	High	Low	Unstable performance and weak convergence across classifiers.
DEoptim	High	Medium–High	Low	Medium	Strong exploration improves precision but increases runtime significantly.
Bayesian Optimization	Medium	Medium	Medium	High	Stable convergence but does not consistently achieve top performance.
AutoFT	Medium–High	Medium–High	Medium	High	Maintains stable optimization across models with moderate cost.
mlrMBO	High	High	Medium	High	Consistently strong performance across all metrics.
TuRBO	High	Highest	Medium–High	High	Best F1-score due to effective local search and fast convergence.
Proposed Method	Highest	High	High	High	Achieves highest precision with efficient search by limiting unnecessary evaluations and early stopping.

putationally expensive surrogate models.

The recommendation module further demonstrates practical value, achieving Precision@3 of 0.82 and Recall@3 of 0.77, indicating that improved classification performance directly enhances recommendation quality.

From a practical perspective, deployment in real-world environments requires consideration of security and privacy issues, as the framework processes user-generated content and predicts developer expertise levels. Future implementations should incorporate privacy-preserving mechanisms, secure data handling procedures, and transparent usage policies to ensure responsible application.

6.1 Limitations and future work

Despite the promising results, several limitations should be acknowledged.

First, the recommendation module relies on a rule-based mapping between programming language, experience level, and predefined book lists. While this ensures interpretability and reproducibility, it limits personalization by not capturing individual user preferences or adapting to dynamic user behavior. In dynamic communities such as SO, where user expertise evolves over time and new technologies emerge frequently, such static mappings may require periodic updates to remain effective. Consequently, users with similar predicted profiles receive identical recommendations. Incorporating adaptive approaches such as neural collaborative filtering or transformer-based ranking models could improve personalization, albeit at the cost of increased computational complexity.

Second, the experimental evaluation is conducted on a single domain (Stack Overflow). Although the dataset is rich and diverse, the generalizability of the framework to other domains remains to be validated through cross-domain experiments.

Third, scalability to very large datasets remains a consideration. The current analysis is based on subset experiments (500, 1,000, and 3,000 instances) drawn from a corpus of over one million posts. While results demonstrate stable performance with limited variation, further evaluation on full-scale datasets and more heterogeneous corpora is necessary. Such scenarios may require distributed computation or incremental learning strategies.

Finally, the labeling process relies on keyword-based heuristics, which may not fully capture nuanced developer expertise levels. In addition, the proposed method does not consistently outperform all state-of-the-art optimizers across every metric, particularly recall and F1-score, although it demonstrates strong performance in precision and computational efficiency.

Future work will focus on (i) integrating adaptive recommendation models, (ii) extending the framework to cross-domain settings, (iii) evaluating scalability on large-scale datasets, and (iv) improving labeling strategies using data-driven approaches.

Acknowledgement

Availability of data and material

The dataset used in this manuscript is publicly available from Kaggle: `https://www.kaggle.com/datasets/stackoverflow/stacksample`.

Code availability

The codes for the experiments are available at: `https://github.com/fatmaaltinsoy/a-heuristic-technique-for-hyperpameter-tuning.git`.

Web interface

The developed web interface, HyperMLopt, can be accessed at: `https://hypermlopt.sdu.edu.tr`.

Declarations

The datasets generated and/or analysed during the current study are available in the Kaggle repository: `https://www.kaggle.com/datasets/stackoverflow/stacksample`.

References

[1] Ndukwe I, Licorish S, and MacDonell S (2023) Perceptions on the utility of community question and answer websites like Stack Overflow to software developers, *IEEE Transactions on Software Engineering*, IEEE, vol. 49, pp. 2413–2425. https://doi.org/10.1109/TSE.2022.3220236

[2] Meldrum S, Licorish S, Owen C, and Savarimuthu B (2020) Understanding Stack Overflow code quality: A recommendation of caution, *Science of Computer Programming*, Elsevier, vol. 199, p. 102516. https://doi.org/10.1016/j.scico.2020.102516

[3] Siegmund J, Kästner C, Liebig J, Apel S, and Hanenberg S (2013) Measuring and modeling programming experience, *Empirical Software Engineering*, Springer, vol. 19, no. 5, pp. 1299–1334. https://doi.org/10.1007/s10664-013-9286-4

[4] Wu Y, Wang S, Bezemer C, and Inoue K (2018) How do developers utilize source code from Stack Overflow?, *Empirical Software Engineering*, Springer, vol. 24, pp. 637–673. https://doi.org/10.1007/s10664-018-9634-5

[5] Sela A, Hadar L, Morgan S, and Maimaran M (2019) Variety-seeking and perceived expertise, *Journal of Consumer Psychology*, Wiley. https://doi.org/10.1002/jcpy.1110

[6] Li C, Ishak I, Ibrahim H, Zolkepli M, Sidi F, and Li C (2023) Deep learning-based recommendation system: Systematic review and classification, *IEEE Access*, IEEE, vol. 11, pp. 113790–113835. https://doi.org/10.1109/ACCESS.2023.3323353

[7] Wei H, Jia H, Li Y, and Xu Y (2020) Verify and measure the quality of rule-based machine learning, *Knowledge-Based Systems*, Elsevier, vol. 205, p. 106300. https://doi.org/10.1016/j.knosys.2020.106300

[8] Wang M, Fu W, He X, Hao S, and Wu X (2020) A survey on large-scale machine learning, *IEEE Transactions on Knowledge and Data Engineering*, IEEE, vol. 34, pp. 2574–2594. https://doi.org/10.1109/TKDE.2020.3015777

[9] Bischl B, Binder M, Lang M, et al. (2023) Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley, vol. 13, no. 2, p. e1484. https://doi.org/10.1002/widm.1484

[10] Majidi F, Openja M, Khomh F, and Li H (2022) An empirical study on the usage of automated machine learning tools, *Proceedings of the 38th IEEE International Conference on Software Maintenance and Evolution (ICSME)*, IEEE, pp. 190–200. https://doi.org/10.1109/ICSME55016.2022.00014

[11] Bergstra J and Bengio Y (2012) Random search for hyper-parameter optimization, *Journal of Machine Learning Research*, MIT Press, vol. 13, no. 2. [Online]. Available: https://www.jmlr.org/papers/v13/bergstra12a.html

[12] Snoek J, Larochelle H, and Adams RP (2012) Practical Bayesian optimization of machine learning algorithms, *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, pp. 2951–2959

[13] Vasilescu B, Filkov V, and Serebrenik A (2013) Stack Overflow and GitHub: Associations between software development and crowdsourced knowledge, *Proceedings of the 2013 International Conference on Social Computing*, IEEE, pp. 188–195. https://doi.org/10.1109/SocialCom.2013.35

[14] Chen X, Xu F, Huang Y, Zhou X, and Zheng Z (2024) An empirical study of code reuse between GitHub and Stack Overflow during software development, *Journal of Systems and Software*, Elsevier, p. 111964. https://doi.org/10.1016/j.jss.2024.111964

[15] Nasehi SM, Sillito J, Maurer F, and Burns C (2012) What makes a good code example? A study of programming Q&A in Stack Overflow, *Proceedings of the 28th IEEE International Conference on Software Maintenance (ICSM)*, IEEE, pp. 25–34. https://doi.org/10.1109/ICSM.2012.6405249

[16] Storey M-A, Zagalsky A, Figueira Filho F, Singer L, and German DM (2016) How social and communication channels shape and challenge a participatory culture in software development, *IEEE Transactions on Software Engineering*, IEEE, vol. 43, no. 2, pp. 185–204. https://doi.org/10.1109/TSE.2016.2584053

[17] He J, Xu B, Yang Z, Han D, Yang C, and Lo D (2022) PTM4Tag: Sharpening tag recommendation of Stack Overflow posts with pretrained models, *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, ACM/IEEE, pp. 1–11. https://doi.org/10.1145/3524610.3527897

[18] Joorabchi A, English M, and Mahdi AE (2016) Text mining Stack Overflow: An insight into challenges and subject-related difficulties faced by computer science learners, *Journal of Enterprise Information Management*, Emerald, vol. 29, no. 2, pp. 255–275. https://doi.org/10.1108/JEIM-11-2014-0109

[19] Harrag F and Khamliche M (2020) Mining Stack Overflow: A recommender systems-based model, *Preprints*. https://doi.org/10.20944/preprints202008.0265.v2

[20] Silva CC, Galster M, and Gilson F (2021) Topic modeling in software engineering research, *Empirical Software Engineering*, Springer, vol. 26, no. 6, p. 120. https://doi.org/10.1007/s10664-021-10026-0

[21] Nelder JA and Mead R (1965) A simplex method for function minimization, *The Computer Journal*, Oxford University Press, vol. 7, no. 4, pp. 308–313. https://doi.org/10.1093/comjnl/7.4.308

[22] Storn R and Price K (1997) Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, Springer, vol. 11, no. 4, pp. 341–359. https://doi.org/10.1023/A:1008202821328

[23] Liashchynskyi P and Liashchynskyi P (2019) Grid search, random search, genetic algorithm: A big comparison for NAS, *arXiv preprint* arXiv:1912.06059. https://doi.org/10.48550/arXiv.1912.06059

[24] Bergstra J, Yamins D, and Cox D (2013) Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, *Proceedings of the International Conference on Machine Learning*, PMLR, pp. 115–123. [Online]. Available: https://proceedings.mlr.press/v28/bergstra13.html

[25] Lagarias JC, Reeds JA, Wright MH, and Wright PE (1998) Convergence properties of the Nelder–Mead simplex method in low dimensions, *SIAM Journal on Optimization*, SIAM, vol. 9, no. 1, pp. 112–147. https://doi.org/10.1137/S1052623496303470

[26] Eriksson D, Pearce M, Gardner JR, Turner RD, and Poloczek M (2019) Scalable global optimization via local Bayesian optimization, *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 5497-5508

[27] Bischl B, Richter J, Bossek J, Horn D, Thomas J, and Lang M (2017) mlrMBO: A modular framework for model-based optimization of expensive black-box functions, arXiv preprint, arXiv:1703.03373. https://doi.org/10.48550/arXiv.1703.03373

[28] Choi C, Lee Y, Chen A, Zhou A, Raghunathan A, and Finn C (2024) AutoFT: Learning an objective for robust fine-tuning, *arXiv preprint*, arXiv:2401.10220. https://doi.org/10.48550/arXiv.2401.10220

[29] Li L, Jamieson K, DeSalvo G, Rostamizadeh A, and Talwalkar A (2017) Hyperband: A novel bandit-based approach to hyperparameter optimization, *Journal of Machine Learning Research*, MIT Press, vol. 18, no. 1, pp. 6765–6816. https://doi.org/10.48550/arXiv.1603.06560

[30] Mallik N, Bergman E, Hvarfner C, Stoll D, Janowski M, Lindauer M, and Hutter F (2023) PriorBand: Practical hyperparameter optimization in the age of deep learning, *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, pp. 7377–7391

[31] Liu T, Astorga N, Seedat N, and van der Schaar M (2024) Large language models to enhance Bayesian optimization, *arXiv preprint*, arXiv:2402.03921. https://doi.org/10.48550/arXiv.2402.03921

[32] Montaner M, López B, and De La Rosa JL (2003) A taxonomy of recommender agents on the internet, *Artificial Intelligence Review*, Springer, vol. 19, pp. 285–330. https://doi.org/10.1023/A:1022850703159

[33] He X, Liao L, Zhang H, Nie L, Hu X, and Chua TS (2017) Neural collaborative filtering, *Proceedings of the 26th International Conference on World Wide Web (WWW)*, ACM, pp. 173–182. https://doi.org/10.1145/3038912.3052569

[34] Mantovani RG, Rossi AL, Alcobaça E, Vanschoren J, and de Carvalho AC (2019) A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers, *Information Sciences*, Elsevier, vol. 501, pp. 193–221. https://doi.org/10.1016/j.ins.2019.06.005

[35] Yang C, Akimoto Y, Kim DW, and Udell M (2019) OBOE: Collaborative filtering for AutoML model selection, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, New York, USA, pp. 1173–1183. https://doi.org/10.1145/3292500.3330909

[36] Feurer M, Klein A, Eggensperger K, Springenberg J, Blum M, and Hutter F (2019) Autosklearn: Efficient and robust automated machine learning, in *Automated Machine Learning*, Springer, Cham, pp. 113–134. https://doi.org/10.1007/978-3-030-05318-5_6

[37] Erickson N, Mueller J, Shirkov A, Zhang H, Larroy P, Li M, and Smola A (2020) AutoGluon-Tabular: Robust and accurate AutoML for structured data, *arXiv preprint*, arXiv:2003.06505. https://doi.org/10.48550/arXiv.2003.06505

[38] Wang C, Wu Q, Weimer M, and Zhu E (2021) FLAML: A fast and lightweight AutoML library,

arXiv preprint, arXiv:1911.04706. https://doi.org/10.48550/arXiv.1911.04706

[39] Komer B, Bergstra J, and Eliasmith C (2014) Hyperopt-sklearn: Automatic hyperparameter configuration for Scikit-learn, *Proceedings of the 13th Python in Science Conference*, pp. 32–37. https://doi.org/10.25080/Majora-14bd3278-006

[40] Akiba T, Sano S, Yanase T, Ohta T, and Koyama M (2019) Optuna: A next-generation hyperparameter optimization framework, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, pp. 2623–2631. https://doi.org/10.1145/3292500.3330701

[41] Yuan B, Gui S, Zhang Q, Wang Z, Wen J, Mao B, Liu J, and Yao X (2024) FairerML: An extensible platform for analysing, visualising, and mitigating biases in machine learning, *IEEE Computational Intelligence Magazine*, IEEE, vol. 19, no. 2, pp. 129–141. https://doi.org/10.1109/MCI.2024.3364430

[42] Mansion T, Braud R, Amrani A, Chaouche S, Adjed F, and Cantat L (2024) Debiai: Open-source toolkit for data analysis, visualisation and evaluation in machine learning, *Proceedings of the 20th International Conference on Autonomic and Autonomous Systems*, IARIA, Athens, Greece

[43] Jia Y, Wang H, and Chen D (2023) LAMDA-SSL: Label augmentation and manifold-based data alignment for semi-supervised learning, *Pattern Recognition*, Elsevier, vol. 144, p. 109690. https://doi.org/10.1016/j.patcog.2023.109690

[44] Altınsoy F and Öztürk MM (2026) Hyperparameter optimization web tool: HyperOpt, *Sigma Journal of Engineering and Natural Sciences*, vol. 44, no. 2, pp. 1219–1239. https://doi.org/10.14744/sigma.2026.2033

[45] Zhu Y, Li W, and Li T (2023) A hybrid artificial immune optimization for high-dimensional feature selection, *Knowledge-Based Systems*, Elsevier, vol. 260, p. 110111. https://doi.org/10.1016/j.knosys.2022.110111

[46] Pan H, Chen S, and Xiong H (2023) A high-dimensional feature selection method based on modified gray wolf optimization, *Applied Soft Computing*, Elsevier, vol. 135, p. 110031. https://doi.org/10.1016/j.asoc.2023.110031

[47] Hsu CW, Chang CC, and Lin CJ (2003) A practical guide to support vector classification, Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, pp. 1–12. Available: https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[48] Montañés E, Quevedo JR, and del Coz JJ (2014) Dependent binary relevance models for multi-label classification, *Pattern Recognition*, Elsevier, vol. 47, no. 3, pp. 1494–1508. https://doi.org/10.1016/j.patcog.2013.09.029

[49] Bergstra J, Bardenet R, Bengio Y, and Kégl B (2011) Algorithms for hyper-parameter optimization, *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 24, pp. 2546–2554

[50] Kaggle (2023) Kaggle dataset: Stack Overflow - Stacksample. Available: https://www.kaggle.com/datasets/stackoverflow/stacksample, accessed: 2023-10-17

[51] Manning CD, Raghavan P, and Schütze H (2008) Introduction to information retrieval, Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

[52] Aggarwal CC and Zhai C (2012) Mining text data, Springer. https://doi.org/10.1007/978-1-4614-3223-4

[53] Porter MF (1980) An algorithm for suffix stripping, *Program*, Emerald, vol. 14, no. 3, pp. 130–137. https://doi.org/10.1108/eb046814

[54] Salton G and Buckley C (1988) Term-weighting approaches in automatic text retrieval, *Information Processing & Management*, Elsevier, vol. 24, no. 5, pp. 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

[55] Jurafsky D and Martin JH (2025) Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (3rd ed., draft), Available: https://web.stanford.edu/~jurafsky/slp3/

[56] Mikolov T, Sutskever I, Chen K, Corrado GS, and Dean J (2013) Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 26. https://doi.org/10.48550/arXiv.1310.4546

[57] Deerwester S, Dumais ST, Furnas GW, Landauer TK, and Harshman R (1990) Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, Wiley, vol. 41, no. 6, pp. 391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[58] Peruma A, Simmons S, AlOmar EA, Newman CD, Mkaouer MW, and Ouni A (2022) How do I refactor this? An empirical study on refactoring trends and topics in Stack Overflow, *Empirical Software Engineering*, Springer, vol. 27, no. 1, p. 11. https://doi.org/10.1007/s10664-021-10045-x

[59] West E (2022) Buy now: How Amazon branded convenience and normalized monopoly, MIT Press

[60] Kuhn M (2008) Building predictive models in R using the caret package, *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26. https://doi.org/10.18637/jss.v028.i05

[61] Dimitriadou E, Hornik K, Leisch F, Meyer D, and Weingessel A (2009) E1071: Misc functions of the Department of Statistics (E1071), TU Wien

[62] Ridgeway G (2007) Generalized boosted models: A guide to the gbm package. Available: http://cran.open-source-solution.org/web/packages/gbm/vignettes/gbm.pdf, accessed: 2021

[63] de Carvalho AC and Freitas AA (2009) A tutorial on multi-label classification techniques, in *Foundations of Computational Intelligence Volume 5: Function Approximation and Classification*, Springer, vol. 5, pp. 177–195. https://doi.org/10.1007/978-3-642-01536-6_8

[64] Read J, Pfahringer B, and Holmes G (2008) Multi-label classification using ensembles of pruned sets, *Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM)*, IEEE, pp. 995–1000. https://doi.org/10.1109/ICDM.2008.74

[65] Liaw A and Wiener M (2002) Classification and regression by randomForest, *R News*, vol. 2, pp. 18–22

[66] Yan Y (2016) rBayesOptimization: Bayes optimization of hyperparameters, CRAN: Contributed Packages

[67] Probst P, Wright MN, and Boulesteix AL (2019) Hyperparameters and tuning strategies for random forest, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley, vol. 9, no. 3, p. e1301. https://doi.org/10.1002/widm.1301

[68] Boulkroune A, Boubellouta A, Bouzeriba A, et al. (2026) Novel fixed-time fuzzy sliding-mode synchronization schemes of two uncertain dissimilar chaotic systems, *Soft Computing*, Springer, vol. 30, pp. 121–139. https://doi.org/10.1007/s00500-025-10951-y

[69] Rigatos G, Zouari F, Cuccurullo G, Siano P, and Ghosh T (2020) Flatness-based adaptive fuzzy control for the Uzawa-Lucas endogenous growth model, *AIP Conference Proceedings*, AIP, vol. 2293, p. 310003. https://doi.org/10.1063/5.0026520

https://doi.org/10.31449/inf.v50i2.8753 Informatica **50** (2026) 169–170 **169**

Identification of Heart Sounds for the Analysis of Cardiac Pathology Using Machine Learning

David Susič
Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
E-mail: david.susic@ijs.si

Keywords: Abnormal eart-sound detection, machine learning, deep learning, data augmentation, phonocardiogram

Received: March 28, 2025

Cardiovascular diseases (CVDs), including chronic heart failure (CHF), represent a major global health challenge. Early detection is essential but often limited by data scarcity. This paper explores two key contributions to address this issue: detection of CHF decompensation using phonocardiogram (PCG) recordings and machine learning, and PCGmix, a novel data-augmentation technique tailored to heart sounds. In a study with 37 CHF patients, our models classified decompensated vs. recompensated states with up to 72% accuracy. PCGmix further improves diagnostic performance in scenarios with limited training data, achieving comparable accuracy to models trained on datasets up to 50% larger without augmentation.

Povzetek: Študija preuči zaznavanje dekompenzacije kroničnega srčnega popuščanja iz PCG posnetkov z ML ter predstavi PCGmix kot augmentacijo srčnih zvokov, ki izboljša učenje pri omejenih podatkih.

1 Introduction

Cardiovascular diseases (CVDs) remain the leading cause of mortality worldwide, with chronic heart failure (CHF) representing a major contributor [1, 2]. As incidence rates rise, there is a growing need for early, accurate diagnosis to reduce hospitalizations and improve outcomes. Advances in artificial intelligence (AI), particularly deep learning (DL), have enabled automated analysis of phonocardiogram (PCG) recordings. However, the effectiveness of such models depends heavily on access to large, labeled datasets, which remains an ongoing challenge in clinical practice.

This study addresses two key limitations of current research: detecting heart failure decompensation accurately from PCGs, and overcoming data scarcity through a heart-sound-specific augmentation method. We demonstrate that PCGs contain predictive signals of decompensation and introduce a novel augmentation technique, PCGmix, to support learning in data-scarce scenarios.

2 Chronic heart failure decompensation detection

The decompensated phase in CHF represents acute worsening of symptoms often requiring hospitalization, while recompensation marks clinical stabilization. Detecting this transition from PCG data could aid timely interventions.

Using a dataset of 37 CHF patients with recordings from both phases, we applied a combination of machine learning techniques to classify the two states. Features were extracted primarily from diastole, in both time and frequency domains. Our best-performing model achieved a classification accuracy of 72%. Cardiologists, who were given a subset of our data for classification based solely on sound, achieved an average accuracy of 50%. This result highlights the potential of automated PCG analysis in clinical decision support for CHF monitoring. Study details can be found in [3].

3 Heart sound data augmentation

To address data scarcity, we developed PCGmix, a heart-sound-specific data-augmentation method. Unlike generic approaches, which use single-instance transforms to generate a new instance, PCGmix generates synthetic PCG recordings by interpolating between two real instances to create a new one while preserving diagnostically relevant features. In this way, it also differs from existing data-augmentation methods, which are general and not specialised for heart sounds.

Empirical assessments using a publicly available database of 851 normal and abnormal heart-sound recordings [4] demonstrated that PCGmix outperforms state-of-the-art augmentation techniques when data availability is limited. The results for one of the DL models is shown in Figure 1. The black line represents the no-augmentation baseline. Our method is shown in orange and brown. ADSI indicates how much larger the original dataset would need to be for the no-augmentation approach to reach the same level of performance as the augmentation method.

At 10, 30, 56, 112, and 168 PCG recordings, our method

achieves the same performance as the no-augmentation approach trained on datasets 1.35–1.46, 1.34–1.69, 1.46–1.69, 1.08–2.25, and 1.01–1.65 times larger, respectively. Study details, including results from other DL models, can be found in [5, 6].

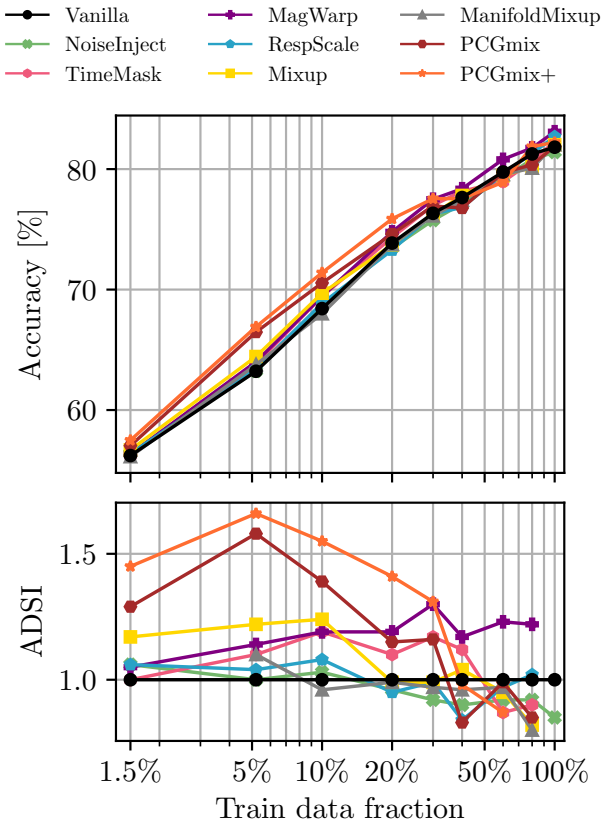


Figure 1: (top) Accuracy and (bottom) ADSI vs. training-data fraction in spectrogram domain experiments using a ResNet DL model. 100% training-data represents 554 PCGs. The x-axis scale is logarithmic.

4 Conclusion

Our work contributes to the knowledge at the intersection of AI and medicine, offering novel insights and methodologies to advance cardiovascular health monitoring and diagnosis. It demonstrates the performance achievable in detecting CHF decompensation using PCGs alone, and introduces and thoroughly validates PCGmix, a novel heart-sound-specific data-augmentation method.

References

[1] Martin, S. S., Aday, A. W., Almarzooq, Z. I., Anderson, et al. (2024) 2024 Heart Disease and stroke statistics: A report of US and global data from the American Heart Association, *Circulation*, 149(8), e347–e913. doi: 10.1161/CIR.0000000000001209.

[2] McDonagh, T. A., Metra, M., Adamo, M., et al. (2021) 2021 ESC Guidelines for the diagnosis and treatment of acute and chronic heart failure, *European Heart Journal*, 42(36), 3599–3726. doi: 10.1093/eurheartj/ehab368.

[3] Susič, D., Poglajen, G., and Gradišek, A. (2022) Identification of decompensation episodes in chronic heart failure patients based solely on heart sounds, *Frontiers in Cardiovascular Medicine*, 9, 1009821. doi: 10.3389/fcvm.2022.1009821.

[4] Liu, C., Springer, D., Li, Q., et al. (2016) An open access database for the evaluation of heart sound algorithms, *Physiological Measurement*, 37(12), 2181–2213. doi: 10.1088/0967-3334/37/12/2181.

[5] Susič, D., Gradišek, A., and Gams, M. (2024). PCGmix: A data-augmentation method for heart-sound classification, *IEEE Journal of Biomedical and Health Informatics*, 28(11), 6874–6885. doi:10.1109/JBHI.2024.3458430

[6] Susič, D.(2024). Identification of heart sounds for the analysis of cardiac pathology using machine learning [Doctoral dissertation, Jožef Stefan International Postgraduate School]

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 2026 (Volume 50) 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 50 Number 2 July 2026 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **50** (2026) Number 2, pp. 1–172