ML-Based Stroke Detection Model Using Different Feature Selection Algorithms

Hussein Abdel-Jaber[1], Ahmed Abdel-Wahab*[1], Anas Abdualqader Hadi[1], Nesrine Atitallah[1], Ali Wagdy Mohamed[2]
Email: habdeljaber@arabou.edu.sa, a.rakha@arabou.edu.sa, a.hadi@ arabou.edu.sa, n.atitallah@ arabou.edu.sa, aliwagdy@gmail.com
[1]Faculty of Computer Studies, Arab Open University (AOU), Riyadh, Kingdom of Saudi Arabia.
[2] Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza, Egypt.
*Corresponding author

Keywords: decision tree, k-nearest neighbours, multilayer perceptron, naïve bayes, random forest, support vector machine

Received: April 20, 2024

Stroke occurs in the brain due to the blockage of blood flow carrying oxygen and nutrients or due to sudden bleeding within the brain. Delaying stroke treatment can lead to serious consequences, including death. This paper proposes a model based on classification algorithms in machine learning to detect whether a stroke has occurred. The classification algorithms used in this study are k-nearest neighbours, decision tree, random forest, naïve Bayes, multilayer perceptron and support vector machine. These algorithms were applied to the classification task using different feature selection methods, namely: all features, select K best (SelectKBest), select percentile (SelectPercentile), select false-discovery rate (SelectFdr), select false-positive rate (SelectFpr) and select family-wise error (SelectFwe). This paper compares the performance of the above algorithms using the different feature selection methods to determine which algorithm provides the best classification results in terms of accuracy, recall, precision and F1-score. The decision tree algorithm shows the highest performance in accuracy, precision and F1-score, regardless of the feature selection method used. Both decision tree and random forest yield the highest and identical recall results when the 'all features' selection method is applied. For the other feature selection methods, decision tree consistently provides the highest recall results. Performance evaluation was conducted by comparing the proposed model to the most relevant works using different machine learning algorithms. The results indicate that the proposed model outperforms other approaches, particularly with the decision tree algorithm. Statistical results, including means, standard deviations and 95% confidence intervals for all features and the target variable in the stroke dataset, were obtained. Trade-offs between precision and recall results for the compared algorithms are also presented.

Povzetek: Razvita je nova metoda za zaznavanje možganske kapi, ki temelji na algoritmih strojnega učenja in različnih metodah izbire značilnosti. Algoritem odločitvenega drevesa je dosegel najboljše rezultate.

1 Introduction

A. *Background*

Stroke is a prevalent global health issue with substantial consequences for individuals, families and healthcare systems. According to the World Health Organization, stroke is the second most common cause of mortality and a major contributor to long-term disability worldwide [1]. Stroke occurs when the flow of blood to the brain is disrupted, either due to a blockage (ischemic stroke) or bleeding (haemorrhagic stroke). The effects of stroke can be profound, leading to physical disability, cognitive impairments and emotional difficulties. Furthermore, stroke places a significant financial strain on healthcare institutions and society as a whole [1].

Machine learning (ML) algorithms have shown significant promise in aiding stroke detection and diagnosis, providing healthcare workers with vital tools in this critical area. These algorithms can analyse vast amounts of medical data, including imaging scans, clinical records and patient demographics, to identify patterns and generate precise predictions. By utilising advanced computational approaches, ML algorithms can assist in the early detection of stroke cases, enabling timely interventions and improving patient outcomes. Recent research has produced encouraging results in this field. For example, Luo et al. (2021) developed a convolutional neural network (CNN) to autonomously identify acute ischemic stroke in brain magnetic resonance imaging (MRI) data [2]. The CNN demonstrated exceptional accuracy and sensitivity (SEN), indicating its potential as a reliable method for stroke diagnosis. In another study by Wu et al. (2020), ML algorithms were used to analyse electrocardiogram (ECG) signals to identify atrial fibrillation, a common risk factor for stroke [3]. Their findings highlighted the feasibility and effectiveness of ML methods in detecting individuals at risk of stroke through ECG data analysis. These recent developments underscore the potential of ML algorithms to enhance stroke detection and diagnosis, providing critical support to healthcare providers and improving patient care.

Stroke detection methods currently consist of both conventional approaches and emerging research that employs ML techniques. Conventional approaches to identifying stroke typically depend on clinical evaluation, which involves analysing the patient's medical history, conducting a physical examination and utilising neuroimaging techniques such as computed tomography (CT) or MRI. Although these approaches have their benefits, they can be influenced by subjective interpretation and may take a significant amount of time, potentially resulting in delays in diagnosis and the initiation of therapy [4].

There has been increasing interest in using ML techniques to improve stroke detection in recent years. ML methods can evaluate extensive datasets and identify significant patterns and characteristics that assist in precise and timely diagnosis. Various studies have explored the use of ML algorithms for stroke identification, demonstrating promising results. Researchers have developed algorithms that use image processing and pattern recognition to automatically detect and categorise abnormalities related to stroke in brain imaging studies, such as CT or MRI. These algorithms have shown exceptional precision in differentiating between various stroke types and can help radiologists achieve more streamlined and accurate diagnoses [5].

In addition, ML algorithms have been used to analyse diverse data sources beyond imaging, including ECGs and electronic health records (EHRs), with the aim of enhancing stroke detection. By examining ECG signals, researchers have developed algorithms to detect distinct patterns associated with atrial fibrillation, a common risk factor for stroke. The integration of ML with EHR data has shown potential in accurately predicting the likelihood of stroke, thereby facilitating timely treatments and preventive measures [6].

While traditional approaches to stroke detection remain valuable, the integration of ML algorithms offers exciting opportunities to enhance accuracy, efficiency and accessibility in stroke diagnosis. Further investigation and advancements in this field may lead to the development of improved and reliable tools for identifying stroke, thereby enhancing patient outcomes and reducing the strain on healthcare systems [7].

The problems addressed in this research work are introduced as follows:

- Failure to recognise stroke can lead to severe consequences.
- Enhancing ML methods for detecting stroke occurrences, such as the proposed model using decision trees (DT) and random forest (RF).
- Evaluating the performance of the proposed model's classification metrics compared to three previous ML algorithms.

B. *Research objectives*

- Assess accuracy: Analyse and evaluate the precision of various ML methods in detecting stroke.
- Analyse and contrast different methods for selecting features: Examine and compare several feature selection methods used by ML algorithms for stroke identification. Evaluate the impact of feature selection on algorithm performance.
- Determine the optimal algorithm: Identify the algorithm that provides the highest level of accuracy and efficiency for early stroke detection.
- Evaluate algorithm performance: Assess the effectiveness of accuracy, recall and F1-score for the algorithms being compared, using various feature selection strategies.
- Identify the most effective feature selection strategy: Determine which feature selection strategy offers the best outcomes in terms of accuracy, precision, recall and F1-score for each algorithm being compared.

C. *Research questions*

This section presents the research questions of this paper:

- Among the compared ML algorithms using different feature selection strategies, which algorithm yields the best results in terms of accuracy, precision, recall and F1-score for detecting stroke?
- Which feature selection strategy is most suitable for use with the comparative ML algorithms in detecting stroke?
- Which model exhibits superior performance when comparing the proposed model with three related models?

D. *Contribution and expected outcomes*

The objective of this paper is to advance the field of stroke detection through a systematic assessment and comparison of the effectiveness of several ML methods. The anticipated results include a comprehensive understanding of the accuracy, precision, recall and F1-score of various ML algorithms. Some ML algorithms are expected to demonstrate higher accuracy in stroke detection, potentially leading to more prompt and precise diagnoses. The paper will conduct a comparative analysis between the proposed model and three state-of-the-art models utilising different ML methods to determine which model offers superior performance.

The findings of this study could assist researchers and clinicians in selecting the most appropriate ML algorithms for stroke detection. This may lead to improved patient outcomes, reduced diagnostic delays and enhanced efficiency in stroke management within the healthcare industry.

E. *Organisation of the paper*

This paper is organised as follows: It begins with an introduction (Section 1) that presents an overview of the significance of stroke detection, the drawbacks of traditional methods and the potential of ML algorithms in enhancing diagnostic precision and efficiency. This section also outlines the research objectives. Section 2 follows with a thorough literature review, covering relevant research on stroke detection methodologies, including both conventional approaches and ML-driven strategies. Section 3 details the methodology, including the dataset used, the ML methods selected for comparison, the features analysed and the criteria for assessment. Section 4 presents the findings, comparing the efficacy of various ML algorithms in stroke detection. This comparison encompasses metrics such as accuracy, precision, recall, F1-score and efficiency. Section 5 examines and evaluates the outcomes, highlighting the advantages and potential limitations of the algorithms. Finally, Section 6 concludes by succinctly recapitulating the key findings, reaffirming the significance of the study and proposing suggestions for future research endeavours.

2 Literature review

Stroke detection is a critical area of medical research, where timely and accurate diagnosis significantly impacts patient outcomes. In the field of ML, numerous studies have explored its application in stroke identification using diverse patient data, including age, gender, blood pressure and heart condition.

For instance, a study conducted by Shoily et al. [8] employed four distinct ML algorithms – naïve Bayes (NB), J48, k-nearest neighbours (KNN) and RF – to determine the type of stroke, whether prospective or existing, based on an individual's physical state and medical records. The researchers collected a substantial amount of data from hospitals for their analysis. The classification results were promising, suggesting that real-time deployment in medical settings is feasible. The study indicates that ML algorithms could significantly enhance disease understanding and act as valuable tools in healthcare. Performance analysis revealed that NB outperformed the other methods. Despite some issues with dataset symmetry, which did not noticeably affect the accuracy of the other algorithms, the NB algorithm did not achieve the expected outcomes. In response, Yang et al. [9] developed a stroke risk prediction model specifically tailored for hypertensive patients. This model leverages historical electronic medical records and ML techniques. Out of 250,788 individuals diagnosed with hypertension, a subset of 57,671 patients was examined, with 9,421 experiencing a stroke within a 3-year follow-up period. The researchers used stratified sampling to account for sex ratio and age categories, creating a balanced sample of both positive and negative cases. Ultimately, 19,953 samples were randomly divided into training and test sets, with a 70% and 30% split, respectively. Four ML algorithms – logistic regression (LR), support vector machine (SVM), RF and gradient boosted trees (XGBoost) – were used for modelling and their efficacy in predicting stroke risk was compared to traditional risk indicators. The tree-based integration approach demonstrated exceptional performance, achieving an area under the receiver operating characteristic curve of 92.20%, surpassing the performance of the other three conventional ML techniques.

Several ML models have been developed to predict the likelihood of a stroke occurring in the brain. The study by Tazin et al. [10] utilised a variety of physiological parameters and ML algorithms, including LR, DT classification, RF classification and voting classifier. These algorithms were used to train four separate models to generate reliable predictions. Among these, RF demonstrated the highest performance, achieving an accuracy rate of approximately 96%. The researchers employed the open-access Stroke Prediction dataset to develop their methodology. Their findings indicate that the RF approach outperforms other methods in predicting brain stroke based on cross-validation metrics. Akter et al. [11] proposed a model aimed at accurately predicting brain stroke. They used a dataset specifically focused on brain stroke and employed RF, SVM and DT classifiers for both training and testing. To evaluate each classifier's effectiveness, the study used various performance metrics, including accuracy, SEN, error rate, false-positive rate (FPR), false-negative rate, root mean square error and log loss. The dataset comprised observations from 5,110 patients and included 12 features pertinent to brain stroke. The RF classifier demonstrated outstanding performance, capturing influential information effectively and achieving a significantly higher accuracy rate of 95.30% compared to the other classifiers.

Extensive research has been conducted on the application of ML algorithms for stroke prediction. A multitude of laboratory tests have been associated with stroke, and developing a predictive model that can assess stroke likelihood based on laboratory test data could have critical implications for saving lives. The study by Alanazi et al. [12] aimed to use computational methodologies and ML techniques to forecast stroke occurrence based on laboratory test data. The researchers used datasets from the National Health and Nutrition Examination Survey and applied three distinct data selection approaches (without data resampling, data imputation and data resampling) to create predictive models. They evaluated the models using four ML classifiers (NB, Bayes Net, DT and RF) and six performance indicators (accuracy, SEN, specificity, positive predictive value, negative predictive value and area under the curve (AUC)). The results demonstrated that accurate and responsive ML models could be created to forecast stroke occurrence using laboratory test data. Notably, the data resampling approach outperformed the other two data selection techniques. When all attributes were used, the RF algorithm produced the most precise predictions, achieving an accuracy of up to 96% with the data resampling approach. The prediction model, constructed using laboratory test data, exhibited exceptional accuracy and user-friendliness.

In the current era of rapid advancements in artificial intelligence and ML, clinical practitioners, medical specialists and decision-makers can leverage developed models to identify key characteristics or factors that increase the likelihood of stroke occurrence. Additionally, they can assess associated risk levels. The study conducted by Elias and Maria [13] aims to apply ML techniques to develop and evaluate several models for creating a reliable framework for predicting long-term stroke risk. The study's main contribution is the development of a stacking technique that demonstrates outstanding performance, achieving an AUC of 98.9%, with F-measure, precision and recall all at 97.4%, and an accuracy of 98%. These metrics – AUC, precision, recall, F-measure and accuracy – confirm the technique's effectiveness. The stacking approach proves its efficacy in identifying individuals at high risk of experiencing a stroke over an extended period. The high AUC values highlight the model's exceptional predictive capability and its ability to distinguish between categories accurately.

Ahammad's research [14] focused on an enhanced method for identifying risk variables and detecting stroke in clinical datasets by employing ML models. The study began with examining the dataset to identify discrepancies and reveal underlying patterns. Subsequently, several subsets of features were selected to identify and prioritise stroke risk factors for classification. Ten ML classification models –RF, XGB, DT, LightGBM, CatBoost, AdaBoost, SVM, multilayer perceptron (MLP), KNN and LR – were used to predict stroke occurrence using a train-test splitting strategy. The performance of these classifiers was evaluated using five metrics: accuracy, precision, F1-score, recall and area under the ROC curve. The researchers compared the outcomes of these ten models, analysing both the full set of available features and the top seven feature subsets. They observed notable differences in classifier performance based on the feature selection used. Generally, gradient boosting and ensemble tree-based classifiers demonstrated superior accuracy with the stroke dataset.

An essential task for physicians is the methodical analysis of various attributes within EHRs to effectively manage these records. Instead of retaining all attributes, the data management team can selectively archive only those crucial for stroke prediction. The study by Dev et al. [15] systematically examines several characteristics found in EHRs to enhance stroke prediction accuracy. By using statistical methods and principal component analysis, they identify the key attributes critical for predicting stroke. Their research highlights that age, heart disease, average glucose level and hypertension are the primary determinants for stroke identification. Three ML techniques were applied to evaluate different feature sets and principal component configurations. The researchers found that the neural network method showed superior performance. Notably, they achieved promising results with only four features. However, they recognised that the lack of additional distinguishing features and the limited size of the dataset prevented the accuracy from exceeding 77%.

Stroke prediction is influenced by lifestyle factors such as smoking status and employment type. The study by Sharma et al. [16] examines the traits of individuals who are more prone to experience a stroke compared to others. The study utilised a dataset from a publicly accessible source and employed various classification algorithms to forecast the likelihood of a stroke occurring in the near future. The RF method achieved a high level of accuracy. Additionally, the study suggests preventive strategies such as quitting smoking, abstaining from drinking and addressing other contributing factors to reduce stroke risk. The results demonstrate the practicality of using historical data mining methods for stroke prediction. The study analysed the accuracy rates of five prominent classification algorithms: DT, RF, NB, MLP and JRip.

Shobayo et al. [17] emphasised the importance of health-related data such as BMI and age in ML models for stroke detection. This study aimed to predict stroke occurrence by combining demographic and behavioural data with the RF algorithm. The experimental results showed that RF outperformed the DT and LR algorithms. Additionally, the research highlighted that age and body mass index (BMI) were the most significant factors in predicting stroke occurrence.

Table 1 compares state-of-the-art stroke detection models based on the performance metrics of various ML algorithms, including KNN, DT, RF, NB, MLP and SVM. The evaluated metrics are accuracy, precision, recall and F1-score.

Table 1 indicates that all studies used the same Kaggle stroke prediction dataset. This presents an opportunity to explore other relevant datasets or combine multiple datasets to improve the generalisability of the models. The studies did not provide a comprehensive analysis of the trade-offs between different performance metrics (e.g. accuracy, precision, recall, F1-score) and their practical implications in a clinical setting. Additionally, none of the studies reported any explicit feature selection methods used. Employing feature selection techniques could enhance the performance of ML models by identifying the most relevant features for stroke prediction.

Table 1: A comparison between the state of art stroke detection models

ML Algorithm	**Accuracy**		
	[18]	***[19]***	***[20]***
KNN	NA	87%	98.82%
DT	99.46%	91%	96.90%
RF	99.98%	NA	99.87%
NB	NA	78%	74.77%
MLP	NA	79%	79.94%
SVM	NA	NA	99.99%
ML Algorithm	**Precision**		
	[25]	***[26]***	***[27]***
KNN	NA	77%	98.66%
DT	99.00%	87%	96.63%
RF	99.00%	NA	99.85%
NB	NA	86%	74.26%
MLP	NA	71%	79.58%
SVM	NA	NA	99.99%
	Recall		

ML Algorithm	***[25]***	***[26]***	***[27]***
KNN	NA	83%	98.97%
DT	99%	90%	97.27%
RF	99%	NA	99.88%
NB	NA	77%	74.39%
MLP	NA	79%	80.05%
SVM	NA	NA	99.99%
ML Algorithm	**F1-score**		
	[25]	***[26]***	***[27]***
KNN	NA	89%	98.81%
DT	99%	89%	96.86%
RF	99%	NA	99.86%
NB	NA	72%	74.32%
MLP	NA	78%	79.71%
SVM	NA	NA	99.99%

3 Methodology

The dataset used in this study for training and evaluating ML algorithms for stroke detection was acquired from Kaggle [21], a well-known online platform for data science and ML. The dataset comprises 40,910 cases with a wide range of clinical and demographic characteristics. It includes ten features and one target variable (11 columns in total) and 40,910 instances (rows). The sex feature had three missing values, which were removed. The features are sex, age, hypertension, heart_disease, ever_married, work_type, residence_type, avg_glucose_level, BMI and smoking_status. The target variable is stroke. The dataset has been augmented, cleaned and balanced to ensure the classes are equally represented.

This study utilised six ML methods to identify instances of stroke: KNN, DT, RF, NB, MLP and SVM. KNN is a nonparametric algorithm that classifies instances based on their proximity to nearby data points. It assigns a class label to a data point by determining the majority class among its k nearest neighbours. DT is a hierarchical model that segments the feature space by making a series of binary decisions, creating a tree-like structure to categorise instances based on feature thresholds. RF is an ensemble technique that improves accuracy and robustness by combining multiple DTs. NB is a probabilistic classifier that applies Bayes' theorem, assuming feature independence, to compute the probability of a class based on feature values. MLP is a type of neural network with multiple layers, including hidden layers, trained using the backpropagation algorithm. It captures complex relationships between features and has been successful in various fields. SVM is a robust method that classifies data by finding an optimal hyperplane in a multidimensional feature space, using kernel functions to handle nonlinear classification. This study aims to evaluate the effectiveness of these ML algorithms in detecting stroke and to analyse their strengths and limitations in accurate stroke diagnosis.

Four evaluation measures were used to compare the ML algorithms KNN, DT, RF, NB, MLP and SVM in detecting stroke. Accuracy, a commonly used metric, quantifies the overall correctness of the algorithm's predictions by calculating the proportion of correctly classified cases out of the total number of instances. Precision measures the algorithm's ability to correctly identify actual stroke cases among all the cases it predicts as positive. It is the ratio of true positive (TP) instances to the sum of TP and false positive (FP) cases. Recall, or SEN, assesses the algorithm's ability to accurately identify all actual positive cases. It is calculated by dividing the number of TP cases by the sum of TP and false negative (FN) cases. The F1-score is a composite metric that combines precision and recall, providing a balanced evaluation of the algorithm's performance. It is the harmonic mean of precision and recall, offering a comprehensive assessment by accounting for both FPs and FNs. These evaluation measures collectively offer a detailed comparison of the ML algorithms based on their accuracy, precision, recall and F1-score, reflecting their overall effectiveness in detecting stroke cases.

A. Dataset description

This study's dataset for stroke prediction comprises approximately 40,910 records, each containing 11 attributes: ten features and one target variable. The sex feature had three missing values, which were removed. All attributes are detailed in Table 2.

Table 2: The attributes in the stroke dataset

Attribute	**Description**
Age	This attribute means a person's age. It's numerical data.
Sex	This attribute means a person's gender. It's categorical data (Male, Female).
Hypertension	This attribute means that this person is hypertensive or not. It's binary data (0,1).
Work_Type	This attribute represents the person's work type. It's categorical data (private, self-employed, other).
Residence_Type	This attribute represents the person's living type. It's categorical data (urban, rural).
Heart_Disease	This attribute means whether this person has a heart disease or not. It's binary data (0,1).
Avg_Glucose_Level	This attribute means what was the level of a person's glucose condition. It's numerical data.
BMI	This attribute means the body mass index of a person. It's numerical data.
Ever_Married	This attribute represents a person's married status. It's Boolean data (true, false).
Smoking_Status	This attribute means a person's smoking condition. It's categorical data (never smoked, formerly smoked, smokes, unknown)
Stroke	This attribute means a person previously had a stroke or not. It's numerical data.

Table 3 presents the number of instances per class both before and after removing the missing values.

Table 3: The number of instances per class before and after deleting the missing values.

Class	Number of Instances before deleting the missing values	Number of Instances after deleting the missing values
0 (not stroke)	20450	20447
1 (stroke)	20460	20460

Tables 4–10 display the distribution of the following features: sex, hypertension, heart_disease, ever_married, work_type, residence_type and smoking_status. Due to the extensive range of values for age, avg_glucose_level and BMI, their distributions are not presented in tables.

Table 4: The distribution of sex feature before and after deleting the missing values.

Sex feature value	Number of instances before deleting the missing values	Number of instances after deleting the missing values
0 (female)	18197	18197
1 (male)	22710	22710

Table 5: The distribution of hypertension feature before and after deleting the missing values.

Hypertension feature value	Number of instances before deleting the missing values	Number of instances after deleting the missing values
0 (patient has not ever had hypertension)	32162	32159
1 (patient has ever had hypertension)	8748	8748

Table 6: The distribution of heart_disease feature before and after deleting the missing values.

Heart_disease feature value	Number of instances before deleting the missing values	Number of instances after deleting the missing values
0 (patient has not ever had heart_disease)	35685	35682
1 (patient has ever had heart_disease)	5225	5225

Table 7: The distribution of ever_married feature before and after deleting the missing values.

ever_married feature value	Number of instances before deleting the missing values	Number of instances after deleting the missing values
0 (patient not married)	7309	7309
1 (patient married)	33601	33598

Table 8: The distribution of work_type feature before and after deleting the missing values.

work_type feature value	Number of instances before deleting the missing values	Number of instances after deleting the missing values
0 (Never_worked)	85	85
1 (children)	431	431
2 (Govt_job)	5588	5588
3 (Self-employed)	9236	9236
4 (Private)	25570	25567

Table 9: The distribution of Residence_type feature before and after deleting the missing values.

Residence_type feature value	Number of instances before deleting the missing values	Number of instances after deleting the missing values
0 (Rural)	19846	19846
1 (Urban)	21064	21061

Table 10: The distribution of smoking_status feature before and after deleting the missing values.

smoking_status feature value	Number of instances before deleting the missing values	Number of instances after deleting the missing values
0 (never smoked)	20921	20921
1 (smokes)	19989	19986

B. Classification algorithms

In this section, we discuss six ML algorithms used for stroke prediction: KNN, DT, RF, NB, MLP and SVM.

KNN is a popular nonparametric ML algorithm used for classification and regression tasks. It operates on the principle that data points with similar attributes are likely to belong to the same class or have similar values. To make a prediction, KNN identifies the K nearest data points in the feature space relative to the given data point. Although Euclidean distance is commonly used for this purpose, other distance metrics can also be applied. For classification, KNN assigns the class label that is most frequent among the K nearest neighbours. In regression tasks, it predicts the value by averaging or weighting the target values of the K closest neighbours. KNN is straightforward and does not assume any specific distribution of the data, allowing it to handle complex decision boundaries. However, calculating distances for each data point can be computationally expensive, particularly with large datasets. Proper selection of K and the distance metric is crucial for accurate predictions. Additionally, data preprocessing, such as normalisation or scaling, may be necessary to ensure reliable feature comparisons [22].

DT is a widely used supervised ML technique for both classification and regression tasks. It constructs a hierarchical model of decisions and outcomes based on input features. The algorithm repeatedly splits the data by feature values to create homogeneous subsets in each branch of the tree. At each node, it selects the most informative feature based on metrics such as information gain or Gini impurity, which improves class separation or reduces subgroup impurity. To make predictions, the algorithm traverses the tree from the root to a leaf node based on the incoming data features. The final classification label or regression value is determined by the leaf node. DTs are transparent and easy to interpret due to their hierarchical structure. They effectively capture complex nonlinear relationships in both numerical and categorical data and are robust against outliers and missing values. However, deep and complex DTs are prone to overfitting. This can be mitigated by pruning the tree and limiting the depth or number of samples in terminal nodes. The simplicity, interpretability and versatility of DTs make them valuable in various fields, including healthcare, finance and customer relationship management [23].

RF is a popular ensemble learning method used for classification and regression tasks. It works by combining the predictions of multiple DTs. RF trains these DTs using random subsets of the training data and input features. This randomness helps to reduce overfitting by introducing variability among the trees. During training, RF uses bootstrap aggregating (bagging) to randomly select and replace segments of the training data, creating slightly different datasets for each tree. Additionally, at each split within the trees, a random subset of features is considered, further enhancing variation. For classification tasks, RF employs majority voting to determine the chosen class based on the most votes from all trees. For regression tasks, RF calculates the average or weighted average of all tree predictions. RF offers several benefits, such as preventing overfitting, effectively managing high-dimensional data and handling missing values and outliers. It also provides estimates of feature importance, indicating the impact of each feature on predictions. Due to its precision and adaptability, RF is well-suited for complex datasets and is widely used in banking, healthcare and image classification [24].

NB is a simple and effective probabilistic classifier based on Bayes' theorem. This model assumes feature independence, meaning the presence or absence of one feature does not affect the presence or absence of another, given the class. NB calculates the probability of a data point belonging to a particular class based on the feature probabilities. During training, it estimates prior and likelihood probabilities for each class and feature, creating a probabilistic model from these values. The model then uses Bayes' theorem to compute the posterior probability for each class and makes predictions based on the attributes of the data point. The class with the highest posterior probability is selected as the prediction. NB offers several advantages, including efficient processing of high-dimensional data and effective performance with small training datasets, with minimal overfitting. It excels when the independence assumption holds. However, NB may struggle with highly correlated features or rare characteristics due to its independence assumption. It also faces challenges with tasks requiring feature interactions or order. NB is widely used for text classification, especially for spam detection and sentiment analysis, due to its speed and simplicity, although performance can vary depending on the dataset [25].

ML is a type of neural network used for classification tasks in ML. It consists of multiple layers of interconnected neurons, forming a feedforward network with input, hidden and output layers. Each neuron in these layers applies an activation function to its inputs and outputs through weighted connections. During training, the MLP algorithm adjusts the weights of these connections to minimise the difference between predicted outputs and true class labels, using optimisation techniques such as gradient descent. The hidden layers of the MLP allow it to model complex relationships and perform extensive data analysis. Activation functions like ReLU or sigmoid introduce nonlinearity, enabling the network to handle nonlinear decision boundaries. The MLP processes input data through the network and predicts class labels based on probability distributions, selecting the most likely class. MLPs offer several advantages, including efficient handling of multidimensional data and the ability to learn complex patterns. Once trained, MLPs can generalise well to new data. They are capable of modelling both linear and nonlinear relationships. However, MLPs are prone to overfitting, especially with large networks. Techniques like dropout and weight decay can mitigate overfitting. Tuning hyperparameters, such as the number of hidden layers, neurons per layer and learning rate, is crucial for optimal performance. MLPs are versatile and widely used in image recognition, natural language processing and medical diagnosis [26].

SVM is a powerful supervised ML algorithm often used for classification tasks, particularly with complex and well-defined datasets. It works by transforming input data into a higher-dimensional space and then finding a hyperplane that best separates the data points into different classes. The goal is to choose a hyperplane that maximises the margin, which is the distance between the hyperplane and the nearest data points from each class. These nearest points, known as support vectors, define the decision boundary of the SVM. SVMs use kernel functions to handle both linearly and nonlinearly separable data. By applying a kernel function, data can be transformed into a higher-dimensional space where a linear hyperplane can be used for separation. Common kernels include linear, polynomial, radial basis function and sigmoid. During training, SVM uses convex optimisation to determine the hyperplane parameters, aiming to minimise classification errors and maximise the margin. Once trained, the SVM model can classify new data points based on their position relative to the hyperplane. The effectiveness of SVM is notable in high-dimensional spaces and small-sample scenarios, as well as its ability to generalise well to new data and resist overfitting. However, SVM can be computationally intensive with large datasets, and the choice of hyperparameters, such as the kernel function and regularisation parameter, can significantly impact performance. SVM is versatile and used in various applications, including image classification, text categorisation and bioinformatics, where precise margin separation and a robust decision boundary are advantageous [27].

C. *Feature selection methods*

Different feature selection methods are employed in this paper as follows:

- Taking all the features of the dataset.
- SelectKBest is used, which means selecting the K best features that have the largest scores for the applied score function, where in this comparison, chi-squared function is used and the best seven features have been chosen (K = 7).
- SelectPercentile is used, which means that the user identified the percentage of features that have the largest scoring for the applied score function. In this comparison, chi-squared function is used as the score function, and 70% of the largest scoring features are chosen.
- SelectFdr is used, where Fdr stands for false discovery rate. The features based on SelectFdr are chosen for the pvalues of the approximated false discovery rate. alpha value is set to 0.05. The score function used is chi-squared.
- SelectFpr is used, where Fpr stands for FP rate. The features based on SelectFpr are chosen when pvalues is smaller than the given alpha depending on the FP rate test, where alpha value is set to 0.05. The score function used is chi-squared.
- SelectFwe is used, where Fwe stands for family wise error. The features based on SelectFwe are chosen for the corresponding to the rate of family wire error, where alpha value is set to 0.05. The score function used is chi-squared.

In this paper, the following univariate feature selection methods are used: SelectKBest, SelectPercentile, false-positive rate (SelectFpr), false-discovery rate (SelectFdr) and family-wise error (SelectFwe). Univariate feature selection methods select features based on univariate statistical tests that assess the relationship between independent and dependent variables. The independent variables with the strongest relationships to the dependent variable are selected. These methods are used to identify the most relevant features for the target variable.

- SelectKBest: Chooses features based on the K highest scores, meaning the selected features have the strongest relationships with the target variable.
- SelectPercentile: Selects features based on the highest scoring percentile, with the chosen features being those most related to the target variable.
- SelectFpr: Chooses features based on the FP rate test.
- SelectFdr: Chooses features using the approximated false-discovery rate.
- SelectFwe: Chooses features based on the family-wise error rate.

In terms of model performance, univariate feature selection methods can either enhance or reduce effectiveness. These methods can improve model performance by removing features with weak relationships to the target variable. However, they may also decrease performance if they eliminate features with strong relationships to the target variable.

Using all features of the dataset can decrease model performance if it includes features that are unrelated to the target variable. Conversely, if all features are strongly related to the target variable, using all of them can enhance model performance.

Univariate feature selection methods reduce dataset dimensionality by selecting the most relevant features [28]. This improves computational efficiency, simplifies models and reduces overfitting [28]. Using all features results in higher computational complexity compared to univariate feature selection methods, as all features are involved in predicting stroke occurrence.

Univariate feature selection methods can use P-values to select features. P-values are related to feature selection in statistical hypothesis testing, particularly in statistics and ML [29]. They help determine whether to select or deselect features from the dataset.

The statistical hypotheses for all features and the target variable are tested as follows.

For sex feature and stroke variable:
Null hypothesis: No significant relationship between sex and stroke.
Alternative hypothesis: A significant relationship between sex and stroke.

For age feature and stroke variable:
Null hypothesis: No significant relationship between age and stroke.
Alternative hypothesis: A significant relationship between age and stroke.

For hypertension feature and stroke variable:
Null hypothesis: No significant relationship between hypertension and stroke.
Alternative hypothesis: A significant relationship between hypertension and stroke.

For heart_disease feature and stroke variable:
Null hypothesis: No significant relationship between heart_disease and stroke.
Alternative hypothesis: A significant relationship between heart_disease and stroke.

For ever_married feature and stroke variable:
Null hypothesis: No significant relationship between ever_married and stroke.

Alternative hypothesis: A significant relationship between ever_married and stroke.

For work_type feature and stroke variable:
Null hypothesis: No significant relationship between work_type and stroke.
Alternative hypothesis: A significant relationship between work_type and stroke.

For residence_type feature and stroke variable:
Null hypothesis: No significant relationship between residence_type and stroke.
Alternative hypothesis: A significant relationship between residence_type and stroke.

For ave_glucose_level feature and stroke variable:
Null hypothesis: No significant relationship between ave_glucose_level and stroke.
Alternative hypothesis: A significant relationship between ave_glucose_level and stroke.

For BMI feature and stroke variable:
Null hypothesis: No significant relationship between BMI and stroke.
Alternative hypothesis: A significant relationship between BMI and stroke.

For smoking_status feature and stroke variable:
Null hypothesis: No significant relationship between smoking_status and stroke.
Alternative hypothesis: A significant relationship between smoking_status and stroke.

The features with P-values less than the given level of significance (e.g., 0.05) are considered to have significant relationships with the target variable. Consequently, the null hypotheses are rejected for these features. Conversely, features with P-values greater than the significance level are considered to have no significant relationship with the target variable, leading to the acceptance of the null hypotheses.

Figure 1 shows the P-values for all features, indicating that BMI, work_type and residence_type have the highest P-values. These P-values are greater than those for the remaining features. In this figure, the P-values of all features, except for BMI, are less than the given level of significance. Consequently, the null hypotheses for all features except BMI are rejected. The null hypothesis for the BMI feature is accepted because its P-value exceeds the significance level. Therefore, the results indicate that all features except BMI have significant relationships with stroke.

Figures 2–6 show the P-values for the selected features using various univariate feature selection methods. The features with the smallest P-values are considered the most relevant. Therefore, the features selected using univariate feature selection methods are those with the smallest P-values. The features with the highest P-values – BMI, work_type and residence_type – were deleted. This is because work_type and residence_type features have less significant relationships with stroke, and the BMI feature has no significant relationship with stroke. The remaining features, which have smaller P-values, are retained. These selected features are sex, age, hypertension, heart_disease, ever_married, ave_glucose_level and smoking_status. In the SelectKBest method, the K parameter is set to 7, selecting seven features. Similarly, in the SelectPercentile method, the percentile parameter is set to 70, also leading to the selection of seven features.

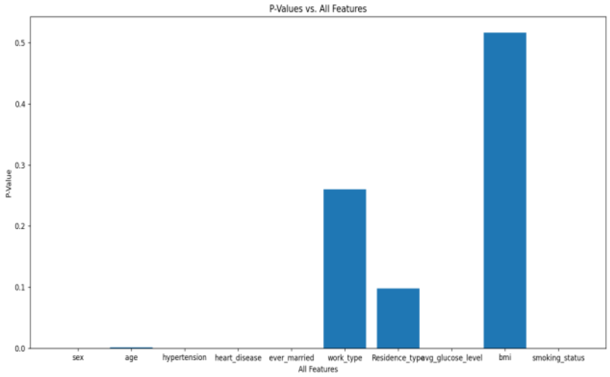

Figure 1: P-Values vs. all features

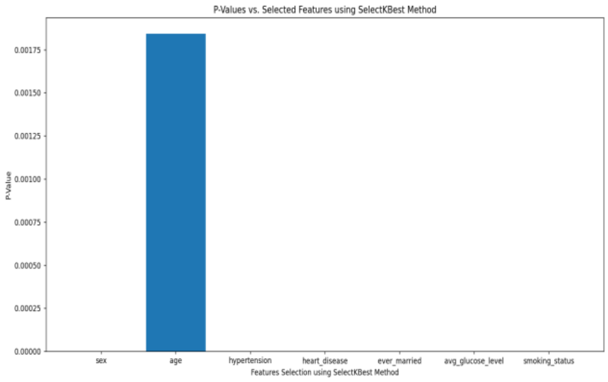

Figure 2: P-Values vs. selected features using SelectKBest method

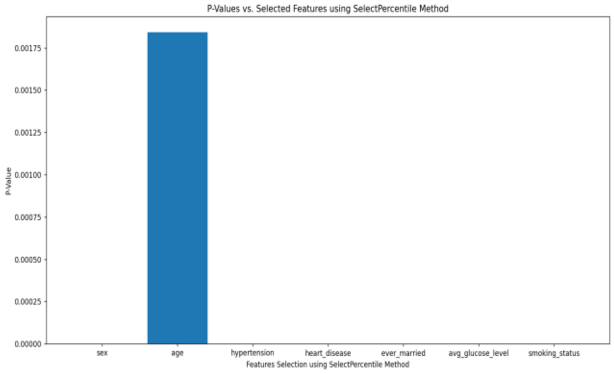

Figure 3: P-Values vs. selected features using SelectPercentile method

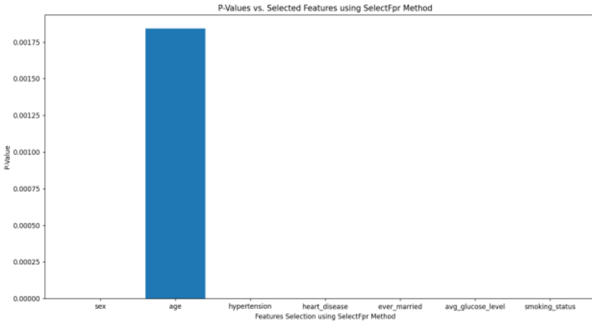

Figure 4: P-Values vs. selected features using SelectFpr method

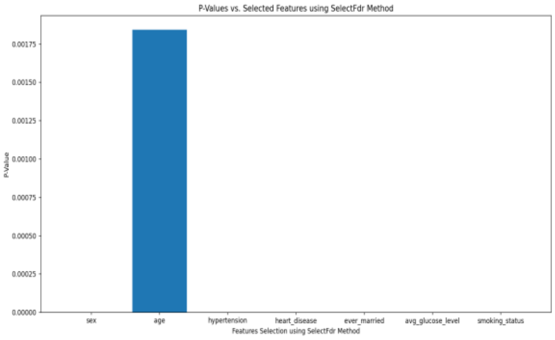

Figure 5: P-Values vs. selected features using SelectFdr method

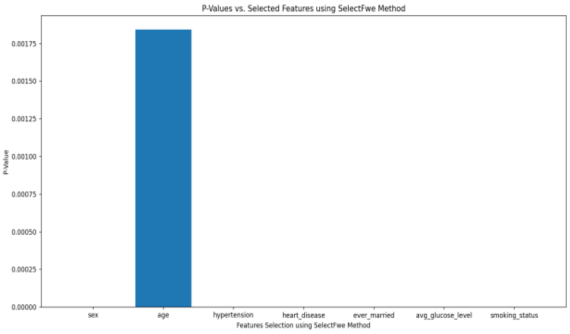

Figure 6: P-Values vs. selected features using SelectFwe method

D. *The proposed models using different machine learning algorithms*

This section presents the proposed models based on the following ML algorithms: KNN, DT, RF, NB, MLP and SVM, used to detect the occurrence of stroke using the stroke dataset mentioned earlier. The proposed models are illustrated in Figure 7:

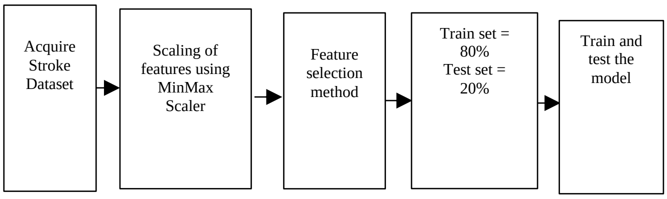

Figure 7: The organization of the proposed model

The pseudocode for the proposed model is shown in Figure 8. This figure provides a detailed view of the model. First, the dataset is obtained from the Kaggle website. Data preprocessing involves deleting missing values and storing the feature data in X and the target values in y. The features are then scaled using MinMaxScaler. You can either keep all features or apply univariate feature selection methods to select the most relevant features for the target variable. The dataset is then split into training and testing sets using the train-test split function, with 80% allocated to the training set and 20% to the testing set. The proposed model, based on different ML algorithms, is trained and tested. Finally, the classification results – accuracy, precision, recall and F1-score – are obtained.

```
1. Acquire stroke dataset
2. Delete missing values
3. Assign features to X and the target
   variable to y
4. Scaling of features using MinMax Scaler
5. Use all features or univariate feature
   selection methods
6. Split the dataset into 80% of train set,
   and 20% for test set
7. Train and test the models based on
   different machine learning algorithms
8. Obtain the classification results for each
   machine learning algorithm
```

Figure 8: The pseudocode of the proposed model

4 Results and discussion

This section is organised into the following subsections. Subsection A presents a comparison of different ML algorithms based on various classification metrics. Subsection B compares the proposed model with state-of-the-art models using different ML algorithms and evaluates their performance with various classification metrics. Subsection C introduces statistical results for all features and the target variable in the stroke dataset. Finally, Subsection D discusses the trade-offs between precision and recall for the different ML algorithms.

A. *Comparison of different machine learning algorithms*

This subsection compares various classification algorithms in ML to determine which algorithm yields the best classification results. The algorithms compared are KNN, DT, RF, NB, MLP and SVM. The classification measures used for comparison are accuracy, precision, recall and F1-score. The dataset is divided into two subsets: the training set, which comprises 80% of the dataset, and the test subset, which constitutes 20% of the dataset.

For the KNN algorithm, the parameter K, representing the number of nearest neighbours, is set to 5. Additionally, the weight function applied in KNN is uniform, meaning that all points in each neighbourhood have equal weights. The parameter p for the Minkowski metric is set to 2, which specifies the use of Euclidean distance. The metric parameter is set to Minkowski, implying that the Euclidean distance is used when p is set to 2.

In DT and RF, the quality of a split is measured using the Gini impurity function. The criterion parameter is set to 'gini', which denotes the use of Gini impurity. To split an internal node, a minimum of two samples is required. The tree's depth is expanded until all leaves are pure or contain fewer samples than the minimum required for a split. The min_weight_fraction_leaf parameter is set to 0.0, representing the minimum weight fraction of the total sample weights needed at the leaf node. The max_features parameter is set to the number of features in DT, which determines how many features are considered when searching for the best split. In DT, the 'best' split is applied at each node, with a minimum of one sample required for leaf nodes. In RF, the forest contains 100 trees. The max_features parameter, which indicates the number of features to consider when searching for the best split, is set to the square root of the total number of features. Each tree in the RF applies the 'best' split.

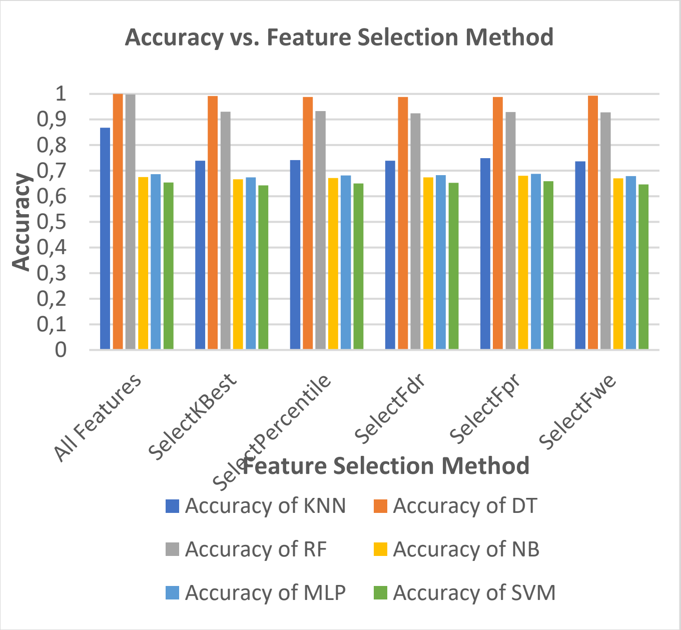

Figure 9: Accuracy results of the compared machine learning algorithms based on different feature selection methods.

The Gaussian NB algorithm is used to perform the classification task. The prior parameter represents the prior probabilities of the classes. If prior probabilities are provided, their values are not adjusted based on the data. The var_smoothing parameter is set to 1e-9. This parameter represents a portion of the largest variance among all features, which is added to the variances to ensure computational stability.

In MLP, the hidden_layer_sizes parameter is set to (100,), indicating that there is one hidden layer with 100 neurons. The activation function used for the hidden layer is the logistic sigmoid function. The solver parameter, which is used for weight optimisation, is set to stochastic gradient descent (sgd). The learning_rate_init parameter represents the initial learning rate, which is set to 0.001. The learning_rate parameter is set to 'constant', meaning the learning rate remains constant and is defined by the learning_rate_init value. The maximum number of iterations is set to 200, and the samples are shuffled at each iteration.

In SVM, the regularisation parameter is set to 1.0. The kernel function used is Linear. Additionally, the probability parameter is set to True.

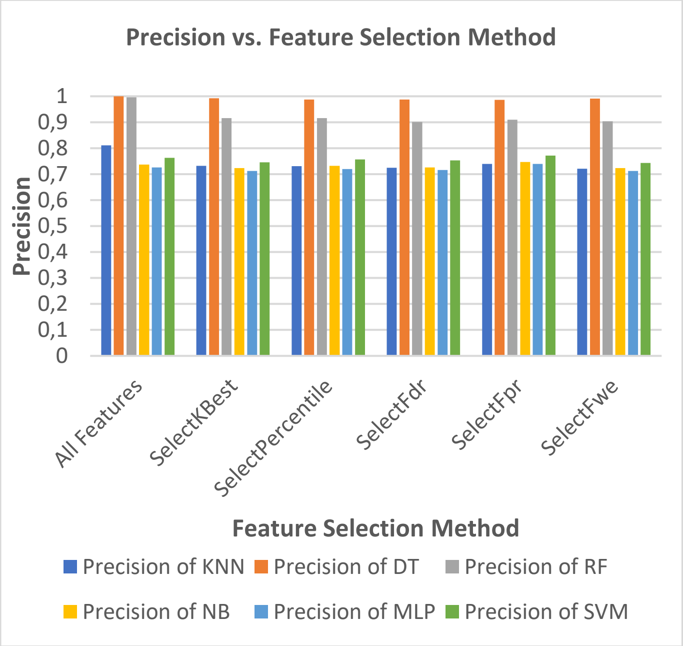

Figure 10: Precision results of the compared machine learning algorithms based on different feature selection methods.

The classification measure results of the compared ML algorithms using different feature selection methods are shown in Figures 9–12. Figure 9 displays the accuracy results. Figures 10 and 11 present the precision and recall results, respectively. Figure 12 illustrates the F1-score results.

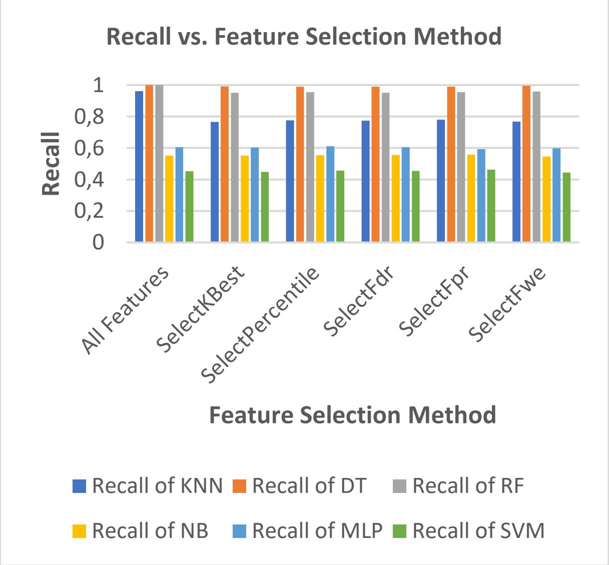

Figure 11: Recall results of the compared machine learning algorithms based on different feature selection methods.

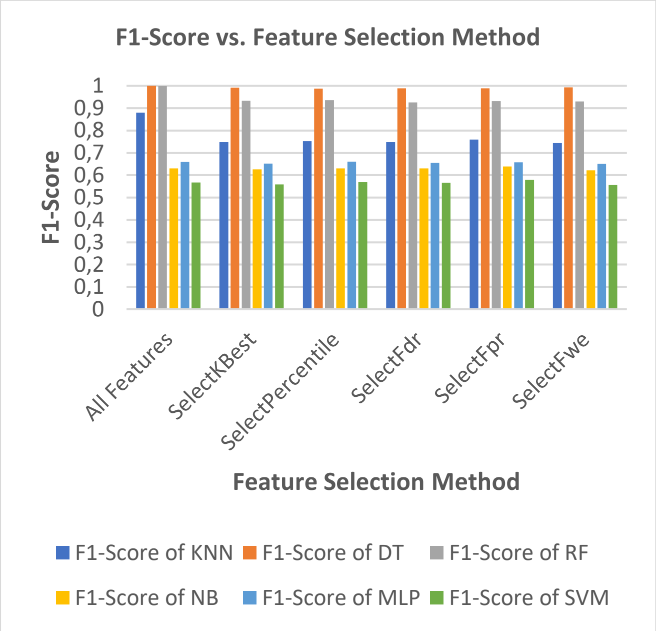

Figure 12: F1-Score results of the compared machine learning algorithms based on different feature selection methods.

Figures 9, 11 and 12 show that DT delivers the most satisfactory results in terms of accuracy, recall and F1-score, regardless of the feature selection method used. The exception is when all features are used, where both DT and RF provide the same recall result. This is because DT generally has the fewest FN cases – instances where the actual class is positive but the predicted class is negative – except when using all features, where DT and RF have the same number of positive cases predicted as negative. RF outperforms KNN, NB, MLP and SVM in accuracy, recall and F1-score across all feature selection methods. This is due to RF having fewer actual positive cases predicted negatively compared to KNN, NB, MLP and SVM. KNN performs better than NB, MLP and SVM in accuracy, recall and F1-score for all feature selection methods because KNN has fewer misclassified positive cases. MLP surpasses NB and SVM in accuracy, recall and F1-score because MLP incorrectly predicts fewer actual positive cases compared to NB and SVM. NB achieves better accuracy, recall and F1-score than SVM due to having fewer incorrectly predicted positive cases.

In Figure 10, when using all features, DT shows the best precision compared to the other algorithms. RF follows with the second-best precision, then KNN, SVM, NB and MLP. DT's high precision results from having the fewest actual negative cases incorrectly predicted as positive. RF has the second-fewest such cases. KNN's precision is better than NB and SVM because it has more TPs. Additionally, SVM's number of actual negative cases incorrectly predicted as positive is lower than NB or MLP, with NB having fewer such cases compared to MLP.

When using the SelectKBest feature selection method, the ordering of algorithms based on precision is similar to that with the all-features method, except for the third and fourth positions. In this case, SVM ranks third and KNN ranks fourth. These results are due to the number of FP cases: DT has the fewest FP cases, followed by RF and then SVM. KNN achieves better precision than NB and MLP with the SelectKBest method because KNN has more TPs compared to NB and MLP. Additionally, NB has fewer FP cases than MLP.

In the remaining selection methods – SelectPercentile, SelectFdr, SelectFpr and SelectFwe – the ordering of algorithms based on precision remains consistent with the SelectKBest method, except for the fourth and fifth positions. In these cases, NB ranks fourth and KNN ranks fifth. These results are attributed to the number of FP cases: DT has the fewest FP cases, followed by RF, SVM and then NB. KNN outperforms MLP in precision across SelectPercentile, SelectFwe, SelectFpr and SelectFdr methods because KNN has more TPs than MLP.

The best results for accuracy, precision, recall and F1-score are achieved by DT, RF and KNN when using the all-features selection method. For NB and SVM, the best results in these metrics are obtained with the SelectFpr method.

For MLP, the highest accuracy and precision are achieved with the SelectFpr method, while the best recall and F1-score are achieved with the SelectPercentile method.

B. *Comparison between the proposed model and state-of-the-art models*

This subsection provides a comparison between the proposed model and state-of-the-art models across various machine learning algorithms, evaluated using different classification metrics.

Table 11 compares the proposed model with state-of-the-art models across various ML algorithms, including KNN, DT, RF, NB, MLP and SVM, using metrics such as accuracy, precision, recall and F1-score.

For accuracy, the proposed model achieves 100% with the DT algorithm. In comparison, references [18] and [20] report accuracies of 99.46% and 96.90%, respectively, while reference [19] reports 91%. This indicates that the proposed model provides slightly higher accuracy than the referenced models.

In terms of precision, the proposed model also achieves 100% with the DT algorithm. Models in [18] and [20] have precision scores of 99% and 96.63%, respectively, with [20] having a lower precision of 87%. Thus, the proposed model with DT offers higher precision than the compared models.

For recall, the proposed model achieves 100% with the DT algorithm. The models in [18], [19] and [20] have recall scores of 99%, 90% and 97.27%, respectively. The proposed model with DT and RF outperforms all three compared models, demonstrating superior ability to identify positive instances.

Finally, the F1-score, which balances precision and recall, is 100% for the proposed model using DT. The F1-scores for models in [18], [19] and [20] are 99%, 89% and 96.86%, respectively. Once again, the proposed model using DT shows a higher F1-score compared to the other models.

Table 11 shows that models based on KNN, NB and MLP from references [19] and [20] achieve better accuracy than the proposed model, with the KNN model in [19] showing slightly superior results. Additionally, RF models in [18] and [20] exhibit marginally better accuracy than the proposed model.

The SVM model in [20] achieves higher accuracy than the proposed model, likely due to its optimised approach.

Overall, based on the metrics provided in the comparison table, the proposed model demonstrates superior performance in accuracy, precision, recall and F1-score with the DT algorithm compared to the models in [18], [19] and [20]. These results indicate that the proposed model offers strong classification capabilities and performs favourably against the referenced models.

The best classification metric results for the proposed model are based on the ML algorithms used in this comparison.

Table 11: A comparison between the proposed model and the state of art models.

ML Algorithm	**Accuracy**			
	[18]	***[19]***	***[20]***	***Proposed Model***
KNN	NA	87%	98.82%	86.81%
DT	99.46%	91%	96.90%	100%
RF	99.98%	NA	99.87%	99.82%
NB	NA	78%	74.77%	68.06%
MLP	NA	79%	79.94%	68.78%
SVM	NA	NA	99.99%	65.87%
ML Algorithm	**Precision**			
	[18]	***[19]***	***[20]***	***Proposed Model***
KNN	NA	77%	98.66%	81.14%
DT	99.00%	87%	96.63%	100%
RF	99.00%	NA	99.85%	99.66%
NB	NA	86%	74.26%	74.74%
MLP	NA	71%	79.58%	73.91%
SVM	NA	NA	99.99%	76.25%
ML Algorithm	**Recall**			
	[18]	***[19]***	***[20]***	***Proposed Model***
KNN	NA	83%	98.97%	96.11%
DT	99%	90%	97.27%	100%
RF	99%	NA	99.88%	100%
NB	NA	77%	74.39%	55.79%
MLP	NA	79%	80.05%	61.15%
SVM	NA	NA	99.99%	46.33%
ML Algorithm	**F1-score**			
	[18]	***[19]***	***[20]***	***Proposed Model***
KNN	NA	89%	98.81%	87.99%
DT	99%	89%	96.86%	100%
RF	99%	NA	99.86%	99.83%
NB	NA	72%	74.32%	63.89%
MLP	NA	78%	79.71%	66.10%
SVM	NA	NA	99.99%	57.90%

C. *Statistical results of all features and the target variable in stroke dataset*

This subsection presents the statistical results for all features and the target variable. The mean, standard deviation and 95% confidence intervals are calculated for each feature and the target variable. Table 12 displays these values for all features and the target variable.

For every feature and the target variable, the true population means fall within the lower and upper limits of their 95% confidence intervals. This indicates a 95% chance that the confidence interval includes the true population mean for each feature and the target variable. The mean values for the features sex, hypertension, heart_disease, ever_married, residence_type, smoking_status and stroke are between 0 and 1, as these features and the target variable are binary. These features, along with the target variable (except for work_type), have lower standard deviations, all less than 1, indicating small data dispersion. In contrast, the mean values for age, work_type, avg_glucose_level and BMI are greater than 1 because these features are not binary. For example, the work_type feature has five distinct values [0–4], while age, avg_glucose_level and BMI have a range of different values. The standard deviations for age, avg_glucose_level and BMI are higher than those for the other features and the target variable, reflecting greater data dispersion for these features. Additionally, the lower and upper limits of the 95% confidence intervals for all features and the target variable are close to their respective means, indicating that the mean values are estimated accurately.

Table 12: Means, standard deviations and the 95% confidence intervals for all features and the target variable.

Feature/Target	**Mean**	**Standard Deviation**	**Lower Limit**	**Upper limit**
sex	0.56	0.50	0.55	0.56
age	51.33	21.62	51.12	51.54
hypertension	0.21	0.41	0.21	0.22
heart_disease	0.13	0.33	0.12	0.13
ever_married	0.82	0.38	0.82	0.83
work_type	3.46	0.78	3.45	3.47
Residence_type	0.51	0.50	0.51	0.52
avg_glucose_level	122.08	57.56	121.52	122.64
bmi	30.41	6.84	30.34	30.47
smoking_status	0.49	0.50	0.48	0.49
stroke	0.50	0.50	0.50	0.51

D. *Trade-offs between precision and recall results*

The trade-offs between precision and recall for different ML algorithms are discussed in this subsection. Figures 13-18 illustrate these trade-offs. Specifically, Figure 13 shows the precision versus recall results for KNN, while Figures 14 and 15 depict the precision versus recall results for DT and RF, respectively. Figures 16, 17 and 18 present these results for NB, MLP and SVM, respectively.

Recall focuses on minimising the number of false negatives (FN), while precision aims to reduce the number of false positives (FP). When recall is higher than precision, it indicates that the number of FN cases is less than the number of FP cases. Conversely, when precision is higher than recall, the number of FP cases is less than the number of FN cases. To improve recall, the number of FN cases needs to be reduced, whereas precision improves with a reduction in FP cases.

Figure 13 demonstrates the trade-off between precision and recall for KNN. As recall increases, precision decreases, and vice versa.

Figure 14 shows that precision and recall for DT are similar. In Figure 15, it is observed that precision slightly decreases as recall increases for RF. Figures 16, 17 and 18 reveal that for NB, MLP and SVM, when precision decreases, recall increases, and vice versa.

As noted earlier, precision aims to minimise FP cases. For instance, if a patient without a stroke is predicted to have one, the number of FP cases increases, decreasing precision, though this does not affect the patient. On the other hand, if a patient with a stroke is predicted not to have one, FN cases increase, reducing recall. This can endanger the patient's life by delaying necessary treatment. Therefore, increasing recall by reducing FN cases is more critical than improving precision by decreasing FP cases.

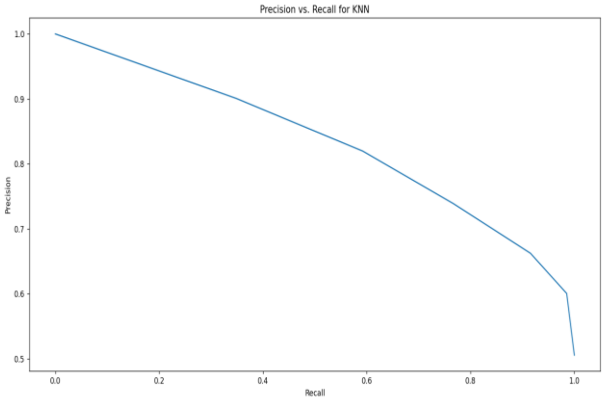

Figure 13: Precision vs. recall results for KNN.

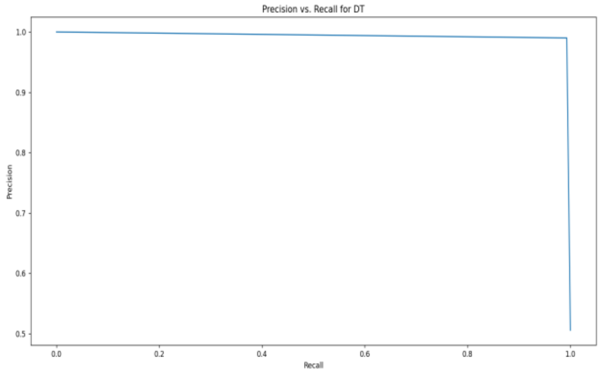

Figure 14: Precision vs. recall results for DT.

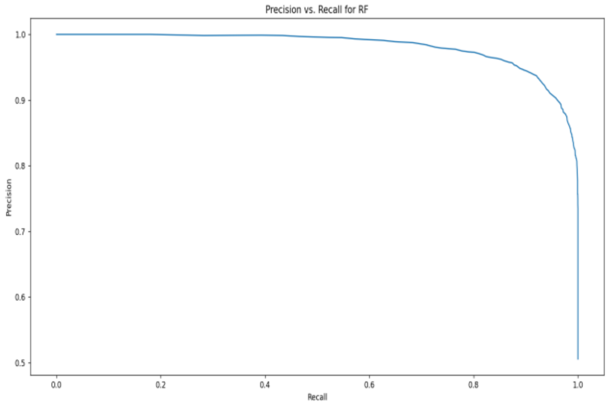

Figure 15: Precision vs. recall results for RF.

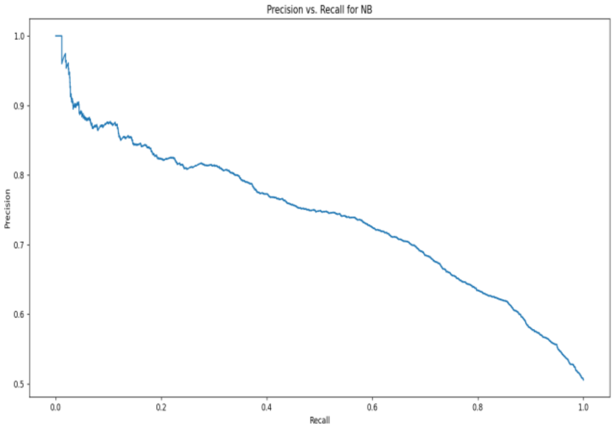

Figure 16: Precision vs. recall results for NB.

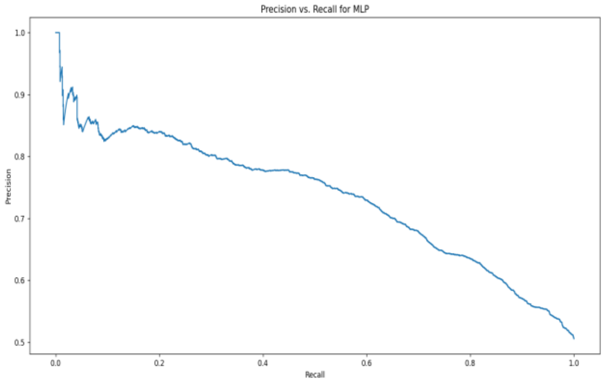

Figure 17: Precision vs. recall results for MLP

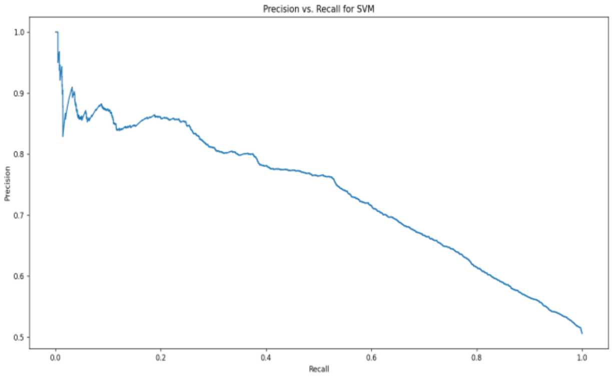

Figure 18: Precision vs. recall results for SVM.

5 Conclusion

Prompt identification of a stroke is crucial, as any delay in detection can endanger patients' lives. This paper proposes a model based on various machine learning classification algorithms to detect stroke or nonstroke at an early stage. The model uses KNN, DT, RF, NB, MLP and SVM. Different feature selection methods are applied, including all features, SelectKBest, SelectPercentile, SelectFdr, SelectFpr and SelectFwe. The algorithms are compared using classification measures to determine which offers the best results. The measures used are accuracy, precision, recall and F1-score. The results of the comparison are summarised as follows:

- DT outperforms all other algorithms in terms of accuracy, recall and F1-score, regardless of the feature selection method used. The only exception is when all features are selected, where DT and RF have the same recall result. RF consistently outperforms KNN, NB, MLP and SVM in accuracy, recall and F1-score across all feature selection methods. KNN provides the third-best results for accuracy, recall and F1-score with any feature selection method. MLP delivers the fourth-best results, while NB and SVM rank fifth and sixth, respectively.
- When using all features, DT achieves the highest precision. RF shows the second-best precision. KNN, SVM, NB and MLP follow in the third, fourth, fifth and sixth positions, respectively.
- With the SelectKBest feature selection method, the algorithms rank for precision as follows: DT, RF, SVM, KNN, NB and MLP.
- For the SelectPercentile, SelectFdr, SelectFpr and SelectFwe feature selection methods, the ranking of algorithms by precision is DT, RF, SVM, NB, KNN and MLP.
- The best results for accuracy, precision, recall and F1-score are achieved by DT, RF and KNN using all features.
- NB and SVM achieve the best results for accuracy, precision, recall and F1-Score with the SelectFpr feature selection method.
- The SelectFpr feature selection method also provides the best accuracy and precision results for MLP, while the best recall and F1-score for MLP are obtained using the SelectPercentile feature selection method.

  A comparison between the proposed model and relevant works in terms of performance metrics for different ML algorithms revealed that the DT-based model achieved better classification results overall. Additionally, the RF-based model outperformed the relevant works in terms of recall.

The results of the mean, standard deviation and 95% confidence interval for all features and the target variable were calculated. The standard deviations for sex, hypertension, heart_disease, ever_married, residence_type, smoking_status, work_type and stroke are small, indicating that the dispersion of these features and the target variable from their means is minimal. The lower and upper limits of the 95% confidence intervals for all features and the target variable are close to their means, suggesting accurate mean results.

Future work could involve applying additional ML algorithms with different feature selection methods to the stroke datasets to achieve improved classification results. Further suggestions include applying the developed model based on various ML algorithms to different datasets to determine which algorithm provides better classification results. Additionally, exploring more feature selection methods, such as Recursive Feature Elimination and Sequential Feature Selection, could help identify which methods offer the best classification performance.

It is suggested to collaborate with clinical institutions to apply the developed model for detecting whether a patient has had a stroke. Implementing the model on various stroke datasets in clinical settings can validate its performance in real-world environments.

Additionally, integrating the ML-based model with clinical decision support systems (CDSSs) could be beneficial. CDSSs can analyse stroke data, providing healthcare professionals with predictions and recommendations. Combining the ML model with CDSSs would assist healthcare providers in making informed decisions and improving patient treatment, helping to accurately classify whether a patient has had a stroke.

Acknowledgements

The authors extend their appreciation to the Arab Open University for funding this work through AOU research fund No. (AOURG-2023-012).

References

[1] J. Pan, G. Wu, J. Yu, D. Geng, J. Zhang, and Y. Wang, "Detecting the Early Infarct Core on Non-Contrast CT Images with a Deep Learning Residual Network," *Journal of Stroke and Cerebrovascular Diseases*, vol. 30, no. 6, p. 105752, Jun. 2021. http://dx.doi.org/10.1016/j.jstrokecerebrovasdis.2021.105752

[2] L. Cui, S. Han, S. Qi, Y. Duan, Y. Kang, and Y. Luo, "Deep symmetric three-dimensional convolutional neural networks for identifying acute ischemic stroke via diffusion-weighted images," *Journal of X-Ray Science and Technology*, vol. 29, no. 4, pp. 551–566, Jul.2021. http://dx.doi.org/10.3233/xst-210861

[3] M. Shao, Z. Zhou, G. Bin, Y. Bai, and S. Wu, "A Wearable Electrocardiogram Telemonitoring System for Atrial Fibrillation Detection," *Sensors*, vol. 20, no. 3, p. 606, Jan. 2020. http://dx.doi.org/10.3390/s20030606

[4] M. Kene, D. Ballard, D. Vinson, A. Rauchwerger, H. Iskin, and A. Kim, "Emergency Physician Attitudes, Preferences, and Risk Tolerance for Stroke as a Potential Cause of Dizziness Symptoms," *Western Journal of Emergency Medicine*, vol. 16, no. 5, pp.

768–776, Sep2015. http://dx.doi.org/10.5811/westjem.2015.7.26158

[5] Y. Miah, C. N. E. Prima, S. J. Seema, M. Mahmud, and M. Shamim Kaiser, “Performance Comparison of Machine Learning Techniques in Identifying Dementia from Open Access Clinical Datasets,” *Advances on Smart and Soft Computing*, pp. 79–89, Oct. 2020. http://dx.doi.org/10.1007/978-981-15-6048-4_8

[6] Hussain and S. J. Park, “Big-ECG: Cardiographic Predictive Cyber-Physical System for Stroke Management,” *IEEE Access*, vol.9,pp.123146–123164,2021. http://dx.doi.org/10.1109/access.2021.3109806

[7] M. S. Sirsat, E. Fermé, and J. Câmara, “Machine Learning for Brain Stroke: A Review,” *Journal of Stroke and Cerebrovascular Diseases*, vol. 29, no. 10, p. 105162, Oct. 2020. http://dx.doi.org/10.1016/j.jstrokecerebrovasdis.2020.105162

[8] T. I. Shoily, T. Islam, S. Jannat, S. A. Tanna, T. M. Alif, and R. R. Ema, “Detection of Stroke Disease using Machine Learning Algorithms,” *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2019. http://dx.doi.org/10.1109/icccnt45670.2019.8944689

[9] Y. Yang, J. Zheng, Z. Du, Y. Li, and Y. Cai, “Accurate Prediction of Stroke for Hypertensive Patients Based on Medical Big Data and Machine Learning Algorithms: Retrospective Study,” *JMIR Medical Informatics*, vol. 9, no. 11, p. e30277, Nov. 2021. http://dx.doi.org/10.2196/30277

[10] T. Tazin, M. N. Alam, N. N. Dola, M. S. Bari, S. Bourouis, and M. Monirujjaman Khan, “Stroke Disease Detection and Prediction Using Robust Learning Approaches,” *Journal of Healthcare Engineering*, vol. 2021, pp. 1–12, Nov. 2021. http://dx.doi.org/10.1155/2021/7633381

[11] B. Akter, A. Rajbongshi, S. Sazzad, R. Shakil, J. Biswas, and U. Sara, “A Machine Learning Approach to Detect the Brain Stroke Disease,” *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Jan. 2022. http://dx.doi.org/10.1109/icssit53264.2022.9716345

[12] E. M. Alanazi, A. Abdou, and J. Luo, “Predicting Risk of Stroke from Lab Tests Using Machine Learning Algorithms: Development and Evaluation of Prediction Models,” *JMIR Formative Research*, vol. 5, no. 12, p. e23440, Dec. 2021. http://dx.doi.org/10.2196/23440

[13] E. Dritsas and M. Trigka, “Stroke Risk Prediction with Machine Learning Techniques,” *Sensors*, vol. 22, no. 13, p. 4670, Jun. 2022. http://dx.doi.org/10.3390/s22134670

[14] T. Ahammad, “Risk factors identification for stroke prognosis using machine learning algorithms,” *Jordanian Journal of Computers and Information Technology*, no. 0, p. 1, 2022. http://dx.doi.org/10.5455/jjcit.71-1652725746

[15] S. Dev, H. Wang, C. S. Nwosu, N. Jain, B. Veeravalli, and D. John, “A predictive analytics approach for stroke prediction using machine learning and neural networks,” *Healthcare Analytics*, vol. 2, p.100032, Nov.2022. http://dx.doi.org/10.1016/j.health.2022.100032

[16] C. Sharma, S. Sharma, M. Kumar, and A. Sodhi, “Early Stroke Prediction Using Machine Learning,” *2022 International Conference on Decision Aid Sciences and Applications (DASA)*, Mar.2022. http://dx.doi.org/10.1109/dasa54658.2022.9765307

[17] O. Shobayo, O. Zachariah, M. O. Odusami, and B. Ogunleye, “Prediction of Stroke Disease with Demographic and Behavioural Data Using Random Forest Algorithm,” *Analytics*, vol. 2, no. 3, pp. 604–617, Aug.2023. http://dx.doi.org/10.3390/analytics2030034

[18] Md. Shafiul Azam, Md. Habibullah, and H. Kabir Rana, “Performance Analysis of Various Machine Learning Approaches in Stroke Prediction,” *International Journal of Computer Applications*, vol. 175, no. 21, pp. 11–15, Sep. 2020. http://dx.doi.org/10.5120/ijca2020920740

[19] M. U. Emon, M. S. Keya, T. I. Meghla, Md. M. Rahman, M. S. A. Mamun, and M. S. Kaiser, “Performance Analysis of Machine Learning Approaches in Stroke Prediction,” *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Nov.2020. http://dx.doi.org/10.1109/iceca49313.2020.9297525

[20] N. Biswas, K. M. M. Uddin, S. T. Rikta, and S. K. Dey, “A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach,” *Healthcare Analytics*, vol.2, p.100116, Nov.2022. http://dx.doi.org/10.1016/j.health.2022.100116

[21] Rishabh, “healthcare-dataset-stroke-data,” *Kaggle*, 18-Mar-2021. [Online]. Available: https://www.kaggle.com/code/rishabh057/healthcare-dataset-stroke-data.

[22] O. Kramer, “K-Nearest Neighbors,” *Dimensionality Reduction with Unsupervised Nearest Neighbors*, pp. 13–23, 2013. http://dx.doi.org/10.1007/978-3-642-38652-7_2

[23] B. Charbuty and A. Abdulazeez, “Classification Based on Decision Tree Algorithm for Machine Learning,” *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, Mar. 2021. http://dx.doi.org/10.38094/jastt20165

[24] B. Shaik and S. Srinivasan, "A Brief Survey on Random Forest Ensembles in Classification Model," International Conference on Innovative Computing and Communications, pp. 253–260, Nov. 2018. http://dx.doi.org/10.1007/978-981-13-2354-6_27

[25] Bhavani and B. Santhosh Kumar, "A Review of State Art of Text Classification Algorithms," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, Apr. 2021. http://dx.doi.org/10.1109/iccmc51019.2021.9418262

[26] J. Singh and R. Banerjee, "A Study on Single and Multi-layer Perceptron Neural Network," *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2019. http://dx.doi.org/10.1109/iccmc.2019.8819775

[27] S. Suthaharan, "Support Vector Machine," *Machine Learning Models and Algorithms for Big Data Classification*, pp. 207–235, 2016. http://dx.doi.org/10.1007/978-1-4899-7641-3_9

[28] Hamza Quddus, "How is the univariate feature selection used in machine learning?," Educative. [Online]. Available: https://www.educative.io/answers/how-is-the-univariate-feature-selection-used-in-machine-learning.

[29] C. Banerjee, "P value and Feature Selection", *Medium*, [Online]. Available: https://medium.com/@chandradip93/p-value-and-feature-selection-629bec71d828