Towards the Hybrid Approach for Predicting Stroke Risk: A Feature Augmented Model

Machine Learning-Based Stroke Prediction

Ting Tin Tin[1]*, Wong Jia Qian[2], Ali Aitizaz[3]*,
Ayodeji Olalekan Salau[4a, b], Omolayo M. Ikumapayi[5], Sunday A. Afolalu[6]
Faculty of Data Science and Information Technology, INTI International University, Nilai, Malaysia[1,4,5,6]
Faculty of Computing and Information Technology, Tunku Abdul Rahman University of Management and Technology, Kuala Lumpur, Malaysia[2]
School of Information Technology, UNITAR International University, Petaling Jaya, Malaysia[1]
School of Technology, Asia Pacific University, Malaysia[3]
Department of Electrical/Electronics and Computer Engineering, Afe Babalola University, Ado, Ekiti, Nigeria[4a]
Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India[4b]
Department of Mechanical and Industrial Engineering, University of Johannesburg, Johannesburg, South Africa[5]
Department of Mechanical and Mechatronics Engineering, Afe Babalola University, Ado Ekiti, Nigeria[5,6]
Department of Mechanical Engineering Science, University of Johannesburg, South Africa[6]

***Abstract*—This project addresses the critical challenge of stroke prediction by developing a hybrid model that integrates the strengths of the Random Forest (RF) and Support Vector Machine (SVM) algorithms. Stroke risk is highly influenced by lifestyle-related factors such as smoking, hypertension, heart disease, and elevated body mass index (BMI). Although existing models, such as standalone Random Forest classifiers, offer moderate predictive performance, achieving an accuracy of approximately 74.53%, they often fall short in clinical reliability. The proposed hybrid model improves prediction accuracy by leveraging Random Forest to capture complex, nonlinear relationships and determine feature importance, while SVM enhances performance in high-dimensional spaces by establishing precise decision boundaries. This study also includes a comprehensive literature review that evaluates existing algorithms, their implementation in current systems, and cross-domain insights, ultimately forming the development of a novel conceptual framework. The anticipated outcome is a robust, data-driven predictive tool that enhances clinical decision-making and supports early intervention strategies. By combining complementary machine learning techniques, this hybrid approach aims to set a new benchmark in stroke risk assessment and contribute meaningfully to patient care in modern healthcare environments towards sustainable public health.**

***Keywords*—Public health; Random Forest; Support Vector Machine; hybrid model; stroke prediction**

I. Introduction

Stroke remains a major global health issue, consistently ranking among the leading causes of long-term mortality and disability. In Malaysia, it is the third leading cause of death [1]. Despite advances in medical science and healthcare infrastructure, the abrupt and often debilitating nature of stroke continues to pose serious challenges. Many patients receive little to no warning before a stroke occurs, which underscores the urgent need for early identification of risks. Detecting high-risk individuals is critical to reducing both the incidence and severity of strokes, as it allows the implementation of preventive strategies and targeted interventions. However, conventional risk assessment methods often fall short, as they typically rely on a narrow set of clinical indicators and do not fully utilise the wealth of data now available through modern healthcare systems and technologies.

Recent advances in artificial intelligence (AI) and machine learning (ML) have transformed the landscape of medical diagnostics and predictive analytics, offering powerful tools to address complex health challenges such as stroke prediction. A stroke predictive system driven by AI and ML can process and analyse comprehensive patient datasets, including demographics, medical history, and lifestyle factors, to generate highly accurate risk assessments. These systems allow healthcare providers to identify high-risk individuals earlier, customise preventive strategies, and improve predictive accuracy over time as more data becomes available. Such technologies have the potential to significantly reduce stroke-related emergencies, reduce long-term healthcare costs, and improve the quality of life of vulnerable populations.

Nutrition, although often underestimated in clinical evaluations, plays a critical role in stroke risk. According to Spence (2019), adopting a healthy lifestyle, particularly proper nutrition, can reduce stroke risk by up to 80%, with poor diet habits being the most influential risk factor. To support this, a study conducted in a dental school in Pakistan found that 44.4% of men and 60% of women were overweight or obese. The main contributors to this nutritional imbalance included skipping breakfast, frequent consumption of high-calorie snacks, extended screen time, and sedentary behaviour [2]. Although nutrition is an important risk factor, this study focuses primarily on demographic, medical, and lifestyle factors as the core components for the prediction of stroke.

Single-algorithm models such as Support Vector Machines (SVM), Random Forest (RF), and Logistic Regression (LR)

*Corresponding authors.

have been widely used in stroke prediction due to their algorithmic simplicity and satisfactory performance in controlled datasets. Despite their utility, these models exhibit notable limitations when applied to heterogeneous and high-dimensional clinical data. SVMs, for instance, are effective in handling high-dimensional feature spaces; however, they may underperform when modelling complex nonlinear interactions or when exposed to data distributions that are not present during training. Similarly, while Random Forests are robust to noise and overfitting, their performance can degrade with imbalanced datasets or subtle inter-variable dependencies. Logistic regression, although interpretable and widely adopted, is inherently linear and cannot model complex interactions between multiple predictors [3].

These limitations are particularly problematic in the context of stroke prediction, where multifactorial influences, such as smoking status, hypertension, heart disease, and an elevated body mass index (BMI), play critical roles. For example, models based solely on Random Forest have demonstrated only moderate predictive performance, with accuracy around 74.53% [3]. As healthcare data becomes increasingly complex, the inadequacy of single-algorithm models to generalise across diverse populations and capture intricate feature relationships underscores the need for more sophisticated predictive methodologies.

To address these challenges, this study proposes a hybrid approach that integrates the strengths of the SVM and Random Forest algorithms. The SVM component is utilised for its superior performance in high-dimensional spaces and its ability to establish well-defined decision boundaries, while the Random Forest component contributes its ensemble-based learning capability to model both linear and nonlinear relationships effectively. This dual architecture model aims to leverage the complementary strengths of both methods to improve predictive accuracy and robustness.

Hybrid models of this nature offer several key advantages: they can capture nuanced patterns in data that may be missed by individual algorithms, reduce the risk of overfitting through ensemble learning, and improve the model's ability to generalise across varying patient demographics and clinical conditions. Additionally, by incorporating Random Forest feature importance metrics and geometric precision of SVM, the hybrid model enables more informed clinical interpretations and targeted intervention strategies.

This study contributes to the evolving field of AI-driven healthcare by demonstrating that hybrid machine learning models can significantly improve the accuracy and reliability of stroke risk prediction. These advances have the potential to support clinicians in early identification of high-risk individuals, allowing timely and personalised preventive measures. Ultimately, the implementation of such predictive systems can reduce stroke incidence, improve patient outcomes, and alleviate the general burden on global healthcare systems.

This paper is constructed with Section II reviewing the current studies of machine learning based stroke prediction. This is followed by Section III, which illustrates the methods and tools used in this study. The result of the study is presented in Section IV. Section V discusses the results of this study and compares them with previous studies. Finally, Section VI concludes the research findings, limitations, and future work.

II. Literature Review

A. Historical Development of Random Forest and Support Vector Machine

A Random Forest or random decision tree is a cooperative group of decision trees cooperating to produce a single output. Fig. 1 presents the historical development of Random Forest. The Support Vector Machine (SVM) is a supervised machine learning algorithm that includes regression and classification. Fig. 2 summarises the historical development of SVM.

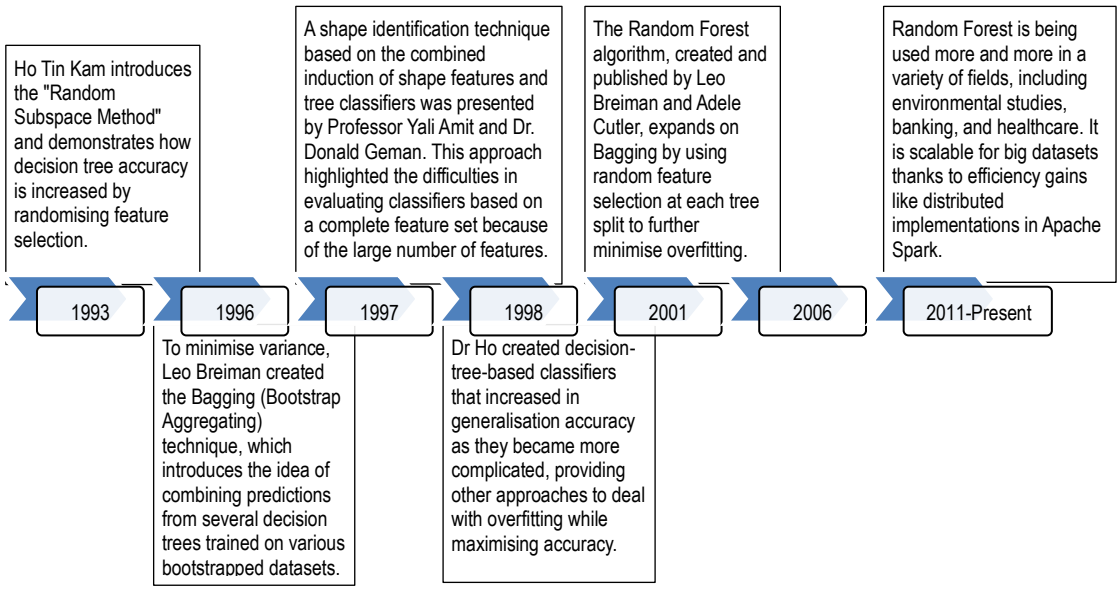


Fig. 1. Historical development of Random Forest [4], [5].

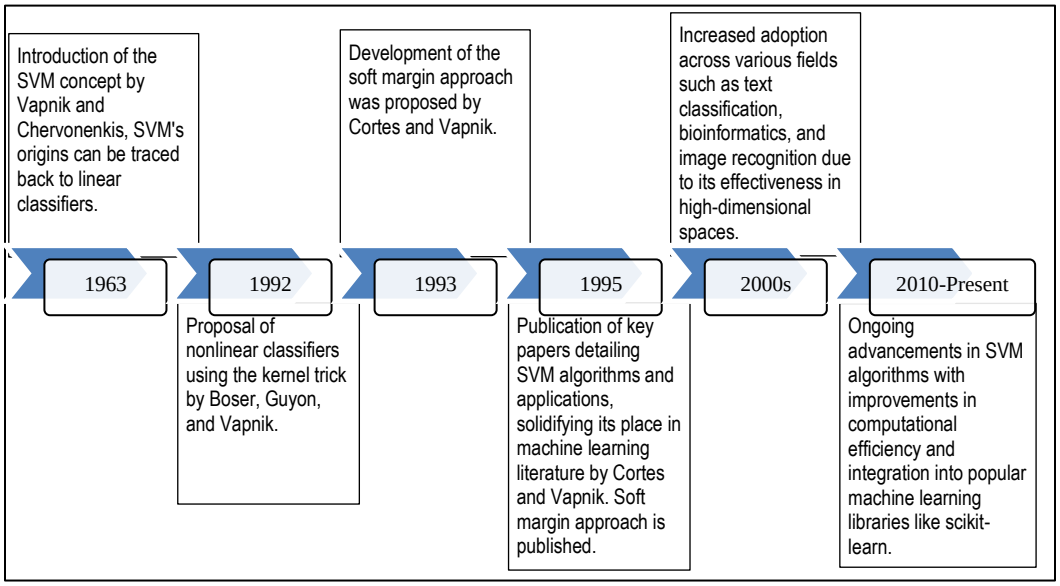


Fig. 2. Historical development of SVM [6], [7].

B. Stroke Prediction Software/Product

1) Random Forest: This section discusses four existing products and their applications in healthcare and machine learning.

a) AutoAI: AutoAI extends AutoML by automating the entire AI lifecycle, including model building. Select and optimise predictive machine learning models for tasks such as predicting cardiovascular disease (CVD) using the Random Forest classifier algorithm. AutoAI processes data sets with features such as age and medical history, improving prediction accuracy through automated hyperparameter tuning and feature classification. The Random Forest model achieved high accuracy in CVD prediction and is suggested for stroke prediction, as it manages high-dimensional data effectively.

b) Magnetic Resonance Imaging (MRI): MRI utilises radio waves and strong magnetic fields to create detailed body images. The Random Forest algorithm improves MRI applications, particularly in tissue segmentation for radiation oncology. It enables a better identification of stroke indicators by segmenting brain tissue and distinguishing between healthy and affected areas. MRI combined with Random Forest improves diagnosis, reduces the need for CT scans, and aids in early stroke detection.

c) Amazon SageMaker: As an ML solution, SageMaker simplifies the development, training, and deployment of machine learning models. Integrates with AWS services for data management and model deployment. The Random Forest algorithm, when used with SageMaker, helps predict stroke and improves cloud-based processes by handling large datasets and optimising model performance for risk assessments.

d) Computed Tomography (CT): CT scans provide detailed cross-sectional images using X-ray imaging. Random Forest improves the analysis of these images for oral cancer diagnosis, noting excellent sensitivity in processing image data. For stroke prediction, CT scans, assisted by Random Forest, improve the detection of critical indicators and risk factors, supporting a timely and accurate diagnosis.

In general, the integration of machine learning algorithms such as Random Forest with imaging technologies and platforms such as AutoAI and SageMaker offers significant improvements in medical diagnosis, treatment planning, and early prediction of conditions such as stroke and cardiovascular diseases.

2) Support Vector Machine: This section discusses various applications of Support Vector Machine (SVM) and related technologies in different fields, with a focus on their potential for stroke prediction.

a) Face++ Platform: Face++ is a commercial facial recognition platform that offers services such as facial detection and demographic analysis. It is capable of monitoring facial asymmetry, aiding in stroke detection by identifying facial drooping, a sign of stroke. In addition, demographic analysis can help assess risk factors such as age.

b) Machine learning techniques: Techniques such as SVM, Genetic Optimisation, and Particle Swarm Optimisation are used in medical diagnostics with high accuracy. These techniques show potential for stroke prediction through brain imaging and physiological data analysis, which could achieve high prediction accuracy.

c) Renewable energy forecasting: Methods combining K-Nearest-Neighbour (KNN) and SVM improve solar power forecasts and could improve stroke prediction by analysing physiological data to identify stroke risks.

d) Sentiment analysis: SVM classifiers are used to analyse sentiments on social networks. A similar approach could classify medical data into stroke risk categories, improving prediction accuracy and supporting early interventions.

e) Rice plant disease detection: SVM and its variant LS-SVM are used to detect plant diseases with high accuracy. LS-SVM, with its efficiency in handling non-linear data, could

significantly boost stroke detection accuracy by analysing complex physiological data, helping to timely medical intervention.

3) Comparing the performance of different machine learning algorithms: Bentley et al. (2014) compared traditional tools such as SEDAN and Hemorrhage After Thrombolysis (HAT) scores with an automated Support Vector Machine (SVM) model. The study found that using the National Institutes of Health Stroke Scale (NIHSS) and imaging data, the SVM achieved a higher area under the curve (AUC) of 0.744, surpassing traditional methods, which had AUCs between 0.626 and 0.720 [8], [9].

Similarly, ML uses Random Forests and synthetic minority oversampling outperformed logistic regression in predicting long-term stroke mortality. The ML model achieved an AUC of 0.928, significantly better than the logistic regression of 0.745, highlighting the superior predictive capabilities of ML in clinical applications.

Furthermore, deep neural networks are more effective than the ASTRAL score in predicting functional outcomes for stroke patients. A study reported an AUC of 0.888 for the neural network model, compared to 0.839 for the ASTRAL score ($P < 0.001$), further highlighting the growing role of ML in the prognosis of stroke.

Research by Wang et al. (2019) demonstrated the precision of Random Forest algorithms in predicting functional outcomes after intracerebral haemorrhage, with AUCs of 0.899 at one month and 0.917 at six months [10]. Lastly, linear SVM regression has been used to predict rehabilitation outcomes in stroke patients, showing promising results for motor and cognitive functions, although it remains challenging to predict the Barthel index. These studies collectively highlight the transformative potential of machine learning to improve stroke outcome predictions and improve clinical decision making.

Abujaber et al. (2023) predict a 90-day prognosis for patients with stroke: a machine learning approach [11]. Stroke remains an important global health issue, ranking as the second leading cause of death worldwide. This study aimed to create and evaluate a machine learning tool to predict the 90-day prognosis of stroke patients after discharge, using the modified Rankin score as the outcome measure. The research analysed data from a national multiethnic stroke registry, which included 15,859 patients with ischemic or hemorrhagic stroke, of whom 7,452 met the study criteria. Feature selection was carried out using correlation and permutation importance methods, and six classifiers were applied, including Random Forest (RF), Classification and Regression Tree, Linear Discriminant Analysis, Support Vector Machine, and K-Nearest Neighbours. The RF model achieved the best performance, with an accuracy of 0.823 and an AUC of 0.893, demonstrating excellent discrimination power. The most influential predictors were stroke type, hospital-acquired infections, admission location, and length of stay. Although the RF model shows potential to tailor patient care and improve stroke prevention, further prospective validation is needed to verify its effectiveness in real-world clinical settings [11].

Research by Rahman et al. (2023) predicts brain stroke using machine learning algorithms and deep neural network techniques. The study aimed to predict the likelihood of an early-stage stroke using both machine learning and deep learning techniques, using a reliable data set for stroke prediction. Various machine learning models, including XGBoost, AdaBoost, LightGBM, Random Forest, Decision Tree, Logistic Regression, K-Nearest Neighbours (KNN), SVM (Linear Kernel), and Naive Bayes, were applied alongside deep learning models, specifically a three-layer and a four-layer artificial neural network (ANN). The Random Forest classifier achieved the highest classification accuracy at 99%, outperforming all other machine learning models. Among deep learning models, the 4-layer ANN showed the best performance with an accuracy of 92.39%, surpassing the 3-layer ANN. In general, the findings revealed that machine learning techniques, particularly Random Forest, outperformed deep neural network models in predicting stroke occurrence [12].

Zhang (2023) research is from radical stroke prediction based on SVM which demonstrates the best overall performance with an accuracy of 0.792, a precision of 0.712, and a high recall of 0.912, resulting in an F1 score of 0.8. This indicates that SVM is particularly effective in identifying true stroke cases (high recall) while maintaining balanced precision and F1 score. Naïve Bayes also performs well, with an accuracy of 0.768, precision of 0.733, and recall of 0.772, and it shows a slightly higher F1 score of 0.7521 compared to other models. Random Forest achieves an accuracy of 0.76, and although its recall (0.842) is relatively high, its precision is lower (0.696), leading to an F1 score of 0.762. KNN matches Random Forest in accuracy (0.76) and shows balanced precision (0.737) and recall (0.736), with an F1 score of 0.737. Logistic regression performs moderately, with an accuracy of 0.752 and an F1 score of 0.739. Lastly, the Decision Tree has the lowest performance, with an accuracy of 0.728 and an F1 score of 0.721. In general, SVM stands out as the most effective model for stroke prediction, especially in recall, which is crucial to accurately identifying stroke cases [13].

SVM stands out as the best performing model for stroke prediction, with a high accuracy of 0.792 and an F1 score of 0.8. Its recall of 0.912 makes it particularly effective in identifying actual stroke cases, which is crucial for minimising missed diagnoses. SVM also balances well with a precision of 0.712, outperforming other models and making it the most reliable choice for real-life stroke prediction applications.

Wu & Fang (2020) research is stroke prediction with Machine Learning Methods among Older Chinese. In this study, the significant class imbalance between stroke and non-stroke cases (approximately 1:19) required the use of data balancing techniques. Regularised Logistic Regression (RLR), Support Vector Machine (SVM), and Random Forest (RF) models were applied to both the original imbalanced data and three balanced datasets generated using Random Over-Sampling (ROS), Random Under-Sampling (RUS), and the Synthetic Minority Over-Sampling Technique (SMOTE). The imbalanced data set yielded high accuracy but extremely low sensitivity (close to 0.00) and an AUC of around 0.50. However, once data balancing techniques were applied, the model performance improved significantly. Although precision and specificity experienced a

slight decrease, sensitivity increased to 0.78 and AUC rose to 0.72, leading to a more balanced and effective model for stroke prediction. For the unbalanced data set, RF had a higher AUC (0.52, 95% CI 0.51–0.53) compared to RLR ($p < 0.01$), though RF's overall performance remained relatively low. No significant differences were observed between SVM and RLR ($p > 0.05$) [14].

In the ROS-balanced dataset, SVM outperformed RLR significantly, achieving an AUC of 0.71 (95% CI 0.68–0.74), while the difference between RF and RLR was not statistically significant ($p > 0.05$). In both the RUS-balanced and SMOTE-balanced datasets, SVM and RF performed similarly to RLR, with no significant differences noted ($p > 0.05$). Overall, data balancing improved the performance of SVM and RF, with SVM showing the greatest AUC improvement in the ROS-balanced dataset [14].

Bandi et al. (2020) research predicts the severity of brain stroke using machine learning. The Random Forest algorithm outperformed other models with an accuracy of 94.23%, a sensitivity of 92.16%, and a specificity of 95.07%, along with a low error rate of 0.04%. Due to its superior performance, it was chosen for further development in the prediction algorithm to improve accuracy. Other models, such as Decision Tree and AdaBoost, also performed well, but Random Forest proved to be the most reliable for this task [15].

The improvised Random Forest model shows significant improvements over the basic version, as indicated in the table. It achieves a high accuracy of 96.97%, precision of 94.56%, sensitivity of 94.9%, specificity of 97.81%, and an F1 score of 94.73%, with a lower error rate of 0.03%. This enhanced model has been integrated into the SPN algorithm to classify stroke risk levels into three categories: low, moderate, and high. Additionally, the SPN algorithm utilises the proposed model to identify key features within the data set that are essential for detecting various types of strokes, including ischemic, intracerebral, and subarachnoid hemorrhagic strokes. These crucial attributes contribute to a better prediction of stroke risk and help to more effectively manage patients [15].

Azam et al. (2020) study focuses on evaluating the efficacy of high-performance machine learning algorithms for the prediction of stroke risk. Three techniques are investigated: Random Forest (RF), Decision Tree (DT) and Logistic Regression (LR). The research also examines important risk variables that lead to stroke, comparing models with and without the smoking status variable. Data preprocessing techniques are used, particularly to balance the data set and improve model performance. With a focus on supporting efforts to prevent strokes, the objective is to compare how well different algorithms predict stroke risk and pinpoint important characteristics that affect the occurrence of strokes. The data set used for the investigation contains 62001 rows and a total of 12 columns. The first eleven columns contain the features that they will use subsequently to forecast the final column, "target(stroke)," which will indicate whether or not the patient will experience a stroke. The 62001 rows are the patient data that we were able to locate in the data set. The results indicate that the RF classifier achieved the highest accuracy at 99.98%, closely followed by the DT classifier with smoking status at 98.78%, while the DT classifier without smoking status recorded an even higher accuracy at 99.46%. In contrast, both LR classifiers performed significantly lower, with accuracies of 71.21% and 81.34%, respectively. These findings highlight the superior predictive capacity of DT and RF models, particularly when incorporating smoking status as a feature, underscoring the importance of feature selection in improving stroke risk prediction [16].

Islam et al. (2021) suggested employing a machine learning algorithm to analyse stroke risk using a healthcare dataset that included a variety of risk factors. There are 5110 observations with 12 attributes in the data set used. Characteristics include age, gender, heart disease, hypertension, type of work, type of residence, average blood sugar level, BMI, smoking status, and stroke. Other factors are independent and stroke is a dependent variable. The algorithms used are Logistic Regression, Decision Tree Classification, K-Nearest Neighbours and Random Forest. They have employed exploratory data analysis (EDA), which helps find patterns, spot anomalies, and create hypotheses, by analysing data sets and summarizing their essential features using data visualisation tools. Feature engineering is used after EDA to convert unprocessed data into useful features that improve model accuracy, especially when working with unbalanced datasets. Synthetic Minority Oversampling Technique (SMOTE) was specifically used in this work to resolve the imbalance in the target variable, which comprises 4908 patients who did not experience a stroke and 201 patients who experienced one stroke. The Random Forest model achieved the highest accuracy on all metrics, recording a precision, recall, and F1 score of 96%. Following in performance, the Decision Tree Classifier (DTC) secured 93% for each metric, while the K-Nearest Neighbours (K-NN) model achieved 90% for both Precision and Recall, with an F1-Score of 90%. Logistic regression performed the least, with a consistent accuracy of 87% across all three metrics. These results indicate the superior efficacy of the Random Forest model in predicting stroke risk compared to other algorithms tested [17].

Alruily et al. (2023) forecast cerebral stroke illnesses; they presented a tuning ensemble RXLM made up of XGBoost, LightGBM, and RF. They used an open-access stroke dataset to predict cerebral stroke risk using Random Forest (RF), Extreme Gradient Boosting (XGBoost) and LightGBM. The dataset was pre-processed using the KNN imputer to handle missing data, feature normalisation, one-hot encoding, and outlier removal. Synthetic Minority Oversampling (SMO) was used to balance the stroke and non-stroke samples after data splitting. Hyperparameter tuning was performed using a random search technique, and the optimal parameters were then integrated into an ensemble model known as RXLM. When this adjusted set of classes was compared with conventional classifiers, all algorithms performed admirably. The data set comprises 249 stroke patients and 4861 normal patients, together with 2994 females, 2115 males, and one other. A training set of 4088 rows (3901, no-stroke and 187 strokes) (80%) and a test set of 1022 rows (20%) were created from the data set [18].

In the first experiment, the performance of four models—Random Forest (RF), XGBoost, LightGBM, and the ensemble RXLM—was evaluated before hyperparameter optimisation.

RXLM achieved the highest scores for precision (96.08%), precision (96.65%), F1 score (96.06%), Kappa (92.16%), and MCC (92.2%), while XGBoost had the highest AUC (99.13%). RF had the highest recall (96.56%), but the lowest scores for precision (94.49%), AUC (98.84%), precision (92.73%), F1 score (94.6%), Kappa (88.98%) and MCC (89.06%). LightGBM performed similarly to RF and XGBoost, with an accuracy of 94.82%. Overall, RXLM outperformed the other models in most metrics except recall, where it had the lowest value at 95.5%.

In the second experiment, the performance of Random Forest (RF), XGBoost, LightGBM, and the ensemble RXLM was evaluated after hyperparameter tuning using the random search technique. The ensemble RXLM outperformed the others, achieving accuracy (96.34%), AUC (99.38%), precision (96.55%), F1-score (96.33%), Kappa (92.68%), and MCC (92.69%). XGBoost had the highest recall at 98.65%, while its other metrics included accuracy (92.42%), AUC (99.12%), precision (87.74%), F1 score (92.87%), Kappa (84.84%), and MCC (85.52%). LightGBM also performed well, with an accuracy of 95.18% and other metrics showing solid results (AUC: 98.86%, recall: 94.91%, precision: 95.45%, F1-score: 95.17%, Kappa: 90.37%, MCC: 90.39%). On the contrary, RF had the lowest performance across all metrics, with accuracy at 84.73%, AUC at 92.75%, and other scores significantly lower than those of the ensemble and the boosted models.

C. Summary of Reviews

This section summarises existing reviews on stroke predictions by categorising them into different domains, which are healthcare, dental, cloud-based software engineering, renewable energy systems, face recognition, sentiment analysis, and plantation, as shown in Table I.

TABLE I. CATEGORISATION OF PREVIOUS STUDIES ACCORDING TO FOUR DIFFERENT DOMAINS

Criteria	Domain	Reference
Random Forest (RF)	A, B, E	[19], [20], [21], [22]
Support Vector Machine (SVM)	A, F, G, H, I	[19], [23], [24], [25]

Note: A-healthcare technology; B-dental; C-cloud-based software engineering; D-renewable energy systems; E-face recognition; F-sentiment analysis; G-plantation.

III. MATERIALS AND METHODS

Fig. 3 shows the research methodology used in this study. The following sub-sections will explain each step in the research.

A. Data Collection

For the stroke prediction project, we collect data from Kaggle, which contains over 5,000 records, each with 12 essential features for analysis. These features include a unique patient ID, gender, age, and binary indicators for hypertension and heart disease. Additionally, it encompasses marital status, work type, residence type, average glucose level, and body mass index (BMI).

B. Data Pre-Processing

Data preprocessing is essential to prepare the data set for stroke prediction and involves several key steps. First, missing values are identified and imputed using appropriate methods, such as mean or mode, or records may be removed if necessary. Techniques for noise reduction are used to improve data quality, including removing duplicates, cleaning data, and applying data transformation, to enhance data quality. Outlier detection is performed using visualisations and box plot methods, followed by capping or removing extreme values to prevent distortion in analysis. Categorical variables are converted into numerical formats through label encoding or one-hot encoding; for example, the gender variable, which includes "female" and "male," is transformed into 0 for female and 1 for male. Feature scaling is carried out using normalisation or standardization to ensure that all features contribute equally to the model. Finally, strategies to address class imbalance, such as oversampling, under sampling, or adjusting class weights, are implemented to improve model performance and ensure that all classes are adequately represented.

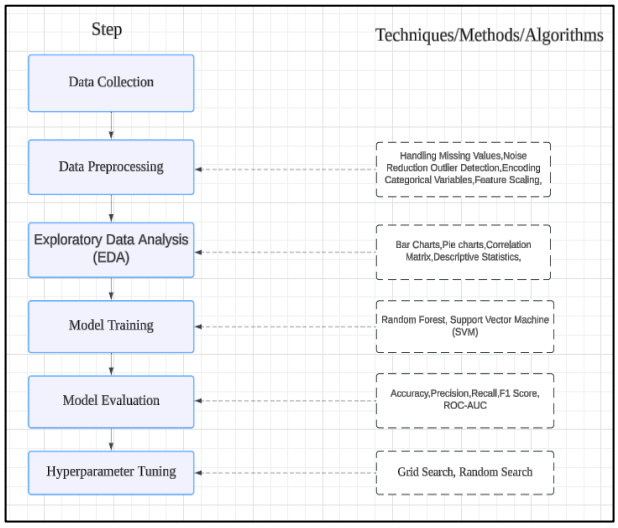


Fig. 3. Research methodology flow diagram.

C. Exploratory Data Analysis (EDA)

Bar charts display the frequency of categorical variables, helping to assess the demographic composition of the data. Pie charts can illustrate the proportions of critical health indicators, such as hypertension and heart disease status. A correlation matrix assesses relationships between numerical features, revealing potential multicollinearity, while descriptive statistics summarise key characteristics, such as mean and standard deviation, aiding in understanding the data's distribution.

D. Model Training

In the model training phase for the stroke prediction project, I will implement and evaluate multiple algorithms, including the Support Vector Machine (SVM) and Random Forest. Specifically, I will explore hybrid approaches that combine SVM with Random Forests to leverage their complementary strengths. The data set will be divided into 80% for training the models and 20% for testing their performance.

E. Model Evaluation

The models in this research are evaluated using Table II equations and evaluation metrics.

TABLE II. MODEL EVALUATION EQUATIONS

Evaluation metrics	Equation/Description
Accuracy	$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Population} = \frac{TP + TN}{TP + TN + FP + FN}$ Note: TP-True Positive; TN-True Negative; TP-Total Population; FP-False Positive; FN-False Negative
Precision	$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} = \frac{TP}{TP + FP}$ Note: TP-True Positive; TN-True Negative; TP-Total Population; FP-False Positive; FN-False Negative
Recall (Sensitivity)	$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} = \frac{TP}{TP + FN}$ Note: TP-True Positive; TN-True Negative; TP-Total Population; FP-False Positive; FN-False Negative
F1 Score	$F1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} = 2\ x\ \frac{TP}{2TP + FP + FN}$ Note: TP-True Positive; TN-True Negative; TP-Total Population; FP-False Positive; FN-False Negative
ROC-AUC	Receiver Operating Characteristics - Area under Curve The ROC curve illustrates the trade-off between sensitivity (true positive rate) and specificity (true negative rate) at various threshold settings.
PR-AUC	Precision-Recall Area Under the Curve The purpose of PR-AUC is to evaluate how well a binary classification model identifies the positive class, especially in cases where the data set is imbalanced.

F. Hyperparameter Tuning

Hyperparameter tuning will be performed to enhance model performance, focussing on key hyperparameters, such as learning rate, batch size, number of epochs, and dropout rate. By systematically adjusting these parameters, we aim to optimise the models for stroke prediction and improve their accuracy and generalisation.

IV. RESULTS

A. Dataset

The stroke prediction data set comprises 5,110 observations, each with 12 attributes. Among these, 10 attributes are considered relevant for prediction. These attributes include essential patient details, such as identification number, age, gender, hypertension, marital status, occupation, type of residence, heart disease status, average glucose level, BMI, smoking habits, and stroke status. The data['stroke'].value_counts() output shows the distribution of the target variable stroke, where 4745 customers (92.9%) did not experience a stroke (0), while 365 customers (7.1%) had a stroke (1). The code applies the ggplot style and creates a bar chart to visualise the distribution of stroke cases. It uses Seaborn's countplot with a yellow-green colour palette to show how many customers had a stroke (1) versus those who did not (0). The bar labels display exact counts, making the imbalance in the data set clear: most of the customers did not experience a stroke.

B. Data Pre-Processing

1) Missing values: The data.isnull().sum() output shows the number of missing values in each column (Fig. 4). All columns have complete data except for BMI, which has 184 missing values. This means that of 5110 entries, only 4926 rows have a valid BMI value. The code calculates the percentage of missing values for each column and visualises them using a point plot. First, it creates a DataFrame (missing) with missing value percentages and then plots these values using Seaborn's pointplot. The x-axis represents column names (rotated for readability), and the y-axis shows the percentage of missing data. The BMI column is expected to show around 3.6% missing values (184 out of 5110). This helps identify missing data patterns, guiding decisions on imputation or removal strategies. Only the BMI column has missing values, accounting for 3.6% of the data set. Next, the fillna() code is used to fill in missing values. The inplace=True parameter updates the DataFrame directly. After replacement, it checks for any remaining missing values using isnull().sum(), confirming that all missing BMI values are now replaced (0 missing values remain). This technique helps maintain data consistency without losing records.

2) Noise reduction

a) Duplicates: The data.duplicated() code is used to check for duplicate rows in the dataset. It then filters the data set to display any duplicate rows. The output shows an empty DataFrame, which means that no duplicate rows were found.

b) Distribution and outliers: The boxplot is used to visualise the distribution and outliers for the numerical columns age, BMI, and avg_glucose_level, as shown in Fig. 5. The age column appears to have a well-distributed range from 0 to 80 years, with a median around 45 years and only a few mild outliers. The BMI column shows several outliers above 50, indicating that some individuals have significantly higher BMI values, although most values fall between 10 and 40. The avg_glucose_level column has a large number of extreme outliers, especially above 150 mg/dL, suggesting that some individuals have unusually high glucose levels. This analysis helps identify potential anomalies, which may need to be addressed through techniques such as capping, transformation, or removal to improve data quality for model training.

```
missing = pd.DataFrame((data.isnull().sum()) * 100 / len(data)).reset_index()  # Calculate percentage of missing values

plt.figure(figsize=(16, 5))  # Corrected syntax for figsize
ax = sns.pointplot(x='index', y=0, data=missing)  # Corrected syntax for specifying x and y parameters
plt.xticks(rotation=90, fontsize=7)
plt.title("Percentage of Missing Value")
plt.ylabel("PERCENTAGE")
plt.show()
```

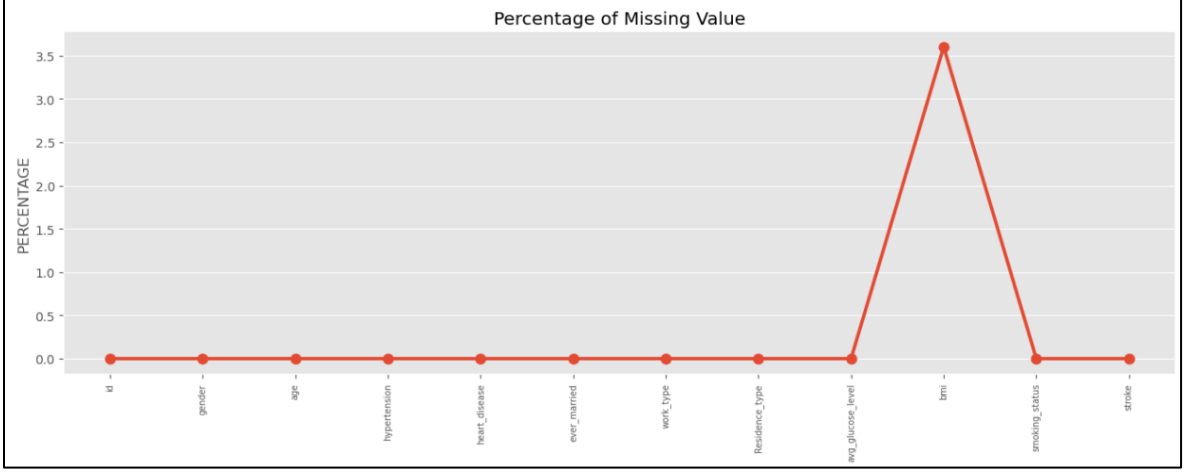


Fig. 4. Data missing analysis and preprocessing.

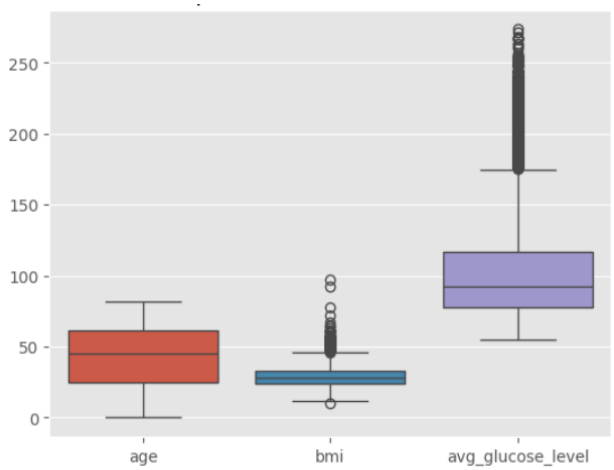


Fig. 5. Box plot for numerical columns.

Data.drop (columns = ['id']) code is used to remove the id column from the data set using. The id column is usually a unique identifier for each row and does not contribute to predictive modelling. The removal of it helps reduce the dimensionality and prevents the model from considering it as a meaningful feature, ensuring a cleaner data set for analysis and machine learning.

c) Out-of-range values: The code in Fig. 6 checks for unrealistic values in the columns of age, BMI, and avg_glucose_level by filtering for values outside reasonable ranges (age: 0-82, BMI: 10-55, avg_glucose_level: 0-400). The results show no invalid values for age and glucose level, but some records have BMI values greater than 55, with a maximum of 97.6. These high BMI values may be outliers or data entry errors that require further investigation or treatment (e.g. capping or removal). Identifying and handling such anomalies is crucial to improving model performance and avoiding biased predictions.

Next, the codes in Fig. 7 are used to filter out unrealistic values in the age, BMI, and avg_glucose_level columns by keeping only records within reasonable ranges (age: 0-82, BMI: 10-55, avg_glucose_level: 0-400). It then ensures that any remaining invalid values (although none should exist after filtering) are replaced with the median value of the respective column. This approach removes extreme outliers while maintaining a consistent and realistic, which helps improve model accuracy and reliability.

```
# Check for age outside a reasonable range (0-82)
print(data[(data['age'] < 0) | (data['age'] > 82.0)])

# Check for BMI outside a reasonable range (10-50)
print(data[(data['bmi'] < 10) | (data['bmi'] > 55)])

# Check for avg_glucose_level outside a reasonable range (0-400)
print(data[(data['avg_glucose_level'] < 0) | (data['avg_glucose_level'] > 400)])
```

```
Empty DataFrame
Columns: [gender, age, hypertension, heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status, stroke]
Index: []
      gender   age  hypertension  heart_disease ever_married      work_type  \
1116    Male  63.0             0              0          Yes       Govt_job
1178    Male  62.0             0              0          Yes       Govt_job
1236  Female  61.0             1              0          Yes        Private
1535  Female  57.0             1              0          Yes        Private
1658  Female  56.0             0              0          Yes       Govt_job
1887  Female  53.0             1              0          Yes        Private
```

Fig. 6. Snippet of code used to check for out-of-range values.

```
[38]: data = data[(data['age'] >= 0) & (data['age'] <= 82.0)]
      data = data[(data['bmi'] >= 10) & (data['bmi'] <= 55)]
      data = data[(data['avg_glucose_level'] >= 0) & (data['avg_glucose_level'] <= 400)]

[41]: data.loc[(data['age'] < 0) | (data['age'] > 82.0), 'age'] = data['age'].median()
      data.loc[(data['bmi'] < 10) | (data['bmi'] > 55), 'bmi'] = data['bmi'].median()
      data.loc[(data['avg_glucose_level'] < 0) | (data['avg_glucose_level'] > 400), 'avg_glucose_level'] = data['avg_glucose_level'].median()
```

Fig. 7. Snippet of code used to filter unrealistic values.

```
ranges = {
    'age': (0, 82),
    'bmi': (10, 55),
    'avg_glucose_level': (0, 400)
}

for column, (min_val, max_val) in ranges.items():
    out_of_range = data[(data[column] < min_val) | (data[column] > max_val)]
    print(f"{column}: {out_of_range.shape[0]} out-of-range values")
```

```
age: 0 out-of-range values
bmi: 0 out-of-range values
avg_glucose_level: 0 out-of-range values
```

Fig. 8. Snippet of codes used to define acceptable value ranges.

Fig. 8 codes are used to define acceptable value ranges for the columns age (0-82), BMI (10-55), and avg_glucose_level (0-400) using a dictionary. It then iterates over these columns, filtering the data set to count the number of values that fall outside the specified ranges. The output confirms that no out-of-range values remain in the dataset, which means that all extreme or unrealistic values have been successfully removed or corrected. This ensures a clean data set for analysis and modelling, reducing the risk of biases caused by erroneous data points.

d) Inconsistent categorical values: The code in Fig. 9(a) is used to check for unique values in the categorical column smoking_status to identify inconsistencies. It then standardises values by converting them to lowercase, ensuring uniformity (e.g. changing Unknown to “unknown”). This transformation prevents issues such as case-sensitive mismatches during analysis or modelling. After standardisation, the data set contains four consistent categories: “unknown”, “never smoked”, “formerly smoked”, and “smokes”, improving the data quality and consistency for machine learning or statistical analysis. Fig. 9(b) checks for unique values in the gender column and detects three categories: “Male”, “Female” and “Other”. To ensure consistency, convert all values to lowercase, standardising them as “male”, “female”, and “other”. This step helps avoid case-sensitive mismatches, ensuring that the data remains clean and uniform for analysis and modelling. However, the presence of “other” could require further review, depending on the context and application. Fig. 9(c) checks the unique values in the work_type column and finds five categories: “Private”, “Self-employed”, “Govt_job”, “Never_worked” and “children”. To ensure consistency, it converts all values to lowercase, standardising them as “private”, “self-employed”, “govt_job”, “never_worked” and “children”. This step prevents case-sensitive discrepancies, ensuring that categorical values remain uniform for analysis and modelling. However, further review may be needed to confirm whether “children” and “never_worked” require special handling.

Fig. 9(d) code checks for unique values in the Residence_type column and finds two categories: “Rural” and “Urban”. To ensure consistency, it converts all values to lowercase, standardising them as “rural” and “urban”. This prevents case-sensitive discrepancies that could cause problems in analysis or machine learning models. The final result confirms that all values are now uniform, improving quality and reliability. Lastly, Fig. 9(e) code checks for inconsistencies in the ever_married column by identifying cases where a person is marked as “Yes” (married) but is younger than 18 years old. If such inconsistencies exist, the code corrects them by changing “ever_married” to “No”. However, since no such cases were found in the dataset, no corrections were needed, and the message "No inconsistencies found" was displayed. This ensures that the data remain logical and accurate for analysis.

a) Encoding: The mapping in Fig. 10 represents how categorical variables have been encoded into numerical values for machine learning. The gender is labelled as 0 for the female, 1 for the the male, and 2 for other. Ever_married is binary, with 0 for "No" and 1 for "Yes". Work_type is categorised into five groups: 0 for children, 1 for government jobs, 2 for never worked, 3 for private sector jobs, and 4 for self-employed individuals. Residence_type is divided into 0 for rural and 1 for urban. Smoking_status is assigned values based on smoking history: 0 for formerly smoked, 1 for never smoked, 2 for currently smoke and 3 for unknown status. Lastly, stroke is classified as 0 for no stroke and 1 for stroke occurrence. This encoding standardises the dataset, making it compatible with machine learning models while preserving essential categorical distinctions.

```
# Check for unique values in categorical columns
print(data['smoking_status'].unique())

# Standardize categorical values if needed (e.g., all lowercase)
data['smoking_status'] = data['smoking_status'].str.lower()

# Check again after standardization
print(data['smoking_status'].unique())
```

```
['Unknown' 'never smoked' 'formerly smoked' 'smokes']
['unknown' 'never smoked' 'formerly smoked' 'smokes']
```

(a) Smoking_status

```
# Check for unique values in categorical columns
print(data['gender'].unique())

# Ensure consistency in gender (e.g., make all values lowercase)
data['gender'] = data['gender'].str.lower()

# Check if all values are now consistented
print(data['gender'].unique())
```

```
['Male' 'Female' 'Other']
['male' 'female' 'other']
```

(b) Gender

```
# Check for unique values in categorical columns
print(data['work_type'].unique())

# Ensure consistency in work_type (e.g., make all values lowercase)
data['work_type'] = data['work_type'].str.lower()

# Check if all values are now consistented
print(data['work_type'].unique())
```

```
['Private' 'Self-employed' 'Govt_job' 'Never_worked' 'children']
['private' 'self-employed' 'govt_job' 'never_worked' 'children']
```

(c) Work_type

```
# Check for unique values in categorical columns
print(data['Residence_type'].unique())

# Ensure consistency in Residence_type (e.g., make all values lowercase)
data['Residence_type'] = data['Residence_type'].str.lower()

# Check if all values are now consistented
print(data['Residence_type'].unique())
```

```
['Rural' 'Urban']
['rural' 'urban']
```

(d) Residence_type

```
# Check inconsistencies
inconsistent_data = data[(data['ever_married'] == 'Yes') & (data['age'] < 18)]

if not inconsistent_data.empty:
    # Correct 'ever_married' to 'No' for inconsistent rows
    data.loc[(data['ever_married'] == 'Yes') & (data['age'] < 18), 'ever_married'] = 'No'
    print("Inconsistencies corrected by updating 'ever_married' to 'No'.")
else:
    print("No inconsistencies found.")
```

```
No inconsistencies found.
```

(e) Ever_married

Fig. 9. Snippet of code used to check for unique values.

```
from sklearn.preprocessing import LabelEncoder, OrdinalEncoder

# Columns for Ordinal and Label Encoding
ordinal_encode_columns = ['ever_married', 'smoking_status']  # Nominal variables
label_encode_columns = ['Residence_type', 'work_type', 'gender']  # Suitable for Label Encoding

# Verify columns before applying Ordinal Encoding
ordinal_encode_columns = [col for col in ordinal_encode_columns if col in data.columns]
if ordinal_encode_columns:
    ordinal_encoder = OrdinalEncoder()
    data[ordinal_encode_columns] = ordinal_encoder.fit_transform(data[ordinal_encode_columns]).astype(int)
else:
    print(f"No valid columns found for Ordinal Encoding: {ordinal_encode_columns}")

# Verify columns before applying Label Encoding
label_encoder = LabelEncoder()
for column in label_encode_columns:
    if column in data.columns:
        data[column] = label_encoder.fit_transform(data[column])
    else:
        print(f"Column not found for Label Encoding: {column}")

# Display the resulting dataset
print("Data after applying Ordinal and Label Encoding techniques:")
data.head(5)
```

Data after applying Ordinal and Label Encoding techniques:

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	1	82.0	1	1	1	3	0	208.30	32.5	3	1
1	0	82.0	1	1	1	4	1	196.92	22.2	1	1

Fig. 10. Categorical variables encoded as numerical variables.

C. Exploratory Data Analysis (EDA)

To understand the class distribution of the target variable stroke after data preprocessing, we analyzed both the count and proportion of each class (0 = no stroke, 1 = stroke) in the dataset. After data preprocessing, the count for “No Stroke” is 4713 instances and “Stroke” is 364 instances: with a proportion of 92.83% “No Stroke” and 7.17% “Stroke”. The imbalance is a common issue in medical datasets and will be addressed during model development by using class-weighted algorithms to mitigate this issue and improve the model's ability to detect stroke cases accurately.

1) Univariate Analysis (Distribution of Individual Variables): The bar graph in Fig. 11 shows the frequency distribution of various categorical characteristics in the data set. This graph helps us to understand the composition and balance of each feature. By this bar chart, we can notice that most of the

gender in the dataset is female ‘0’ and the least is the other ‘2’. Most patients in our data set do not have hypertension ‘0’ and heart disease ‘0’ and most are married ‘1’. Most of the patients are under the private company ‘2’, and the least of them are under the ‘2’ category and have never worked. The resident of Urban ‘1’ is slightly higher than Rural ‘0’. Lastly, most patients never smoked ‘1’; there is almost the same proportion of patients who previously smoked ‘0’ and smoke ‘2’.

Meanwhile, the histogram in Fig. 12 displays the frequency distribution of various numerical features in the data set. This graph helps us to understand the composition and balance of each feature. By using the histogram, we can notice that most patients in the data set are between 50 and 60 years old with an appropriate frequency of 260. For the BMI variable, most of the data is around 28 to 30 with appropriate 630 frequencies. Lastly, most average glucose levels are around 80 to 90 with appropriately 660 frequencies.

2) Bivariate Analysis (Relationship between Variables and Stroke): The bar charts in Fig. 13 display the frequency distribution of various categorical features in the dataset. This visualisation helps us to understand the composition and balance of each feature and its potential relationship with stroke occurrence. From the chart, we observe the following:

- Gender: The number of female patients who experienced a stroke is slightly higher than that of male patients.
- Hypertension or heart disease: Patients with hypertension or heart disease appear to be more likely to have had a stroke compared to those without these conditions.
- Marital Status: The number of stroke cases is higher among married patients than among single patients.
- Work Type: Patients employed in the private sector make up the largest proportion of both stroke and non-stroke cases.
- Residence Type: The deaths and non-stroke cases in urban areas are slightly higher than in rural areas.
- Smoking Status: Among patients who never smoked, most did not have a stroke. In contrast, people who currently smoke have the highest number of stroke cases.

Based on these observations, we can make a preliminary assumption that individuals who are married females, with hypertension and heart disease, work in the private sector, and have a habit of smoking, may be at higher risk of stroke.

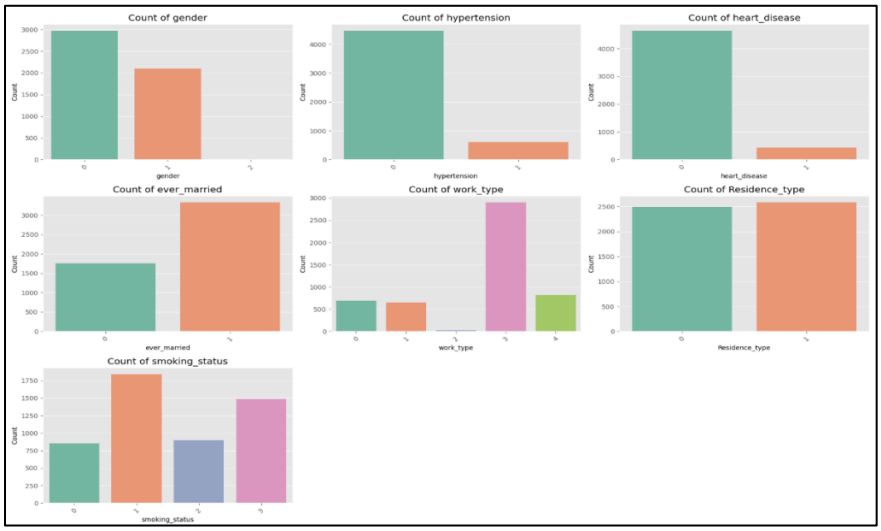


Fig. 11. Frequency distribution of various categorical features.

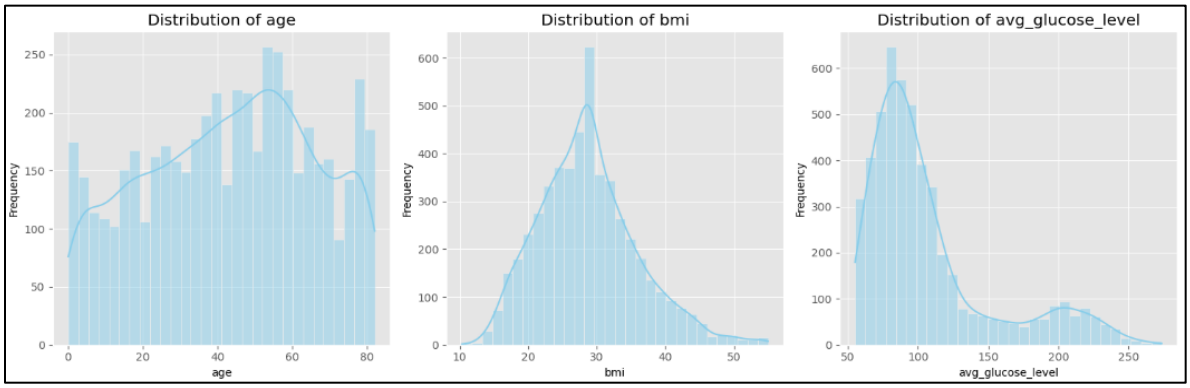


Fig. 12. Frequency distribution of various numerical features.

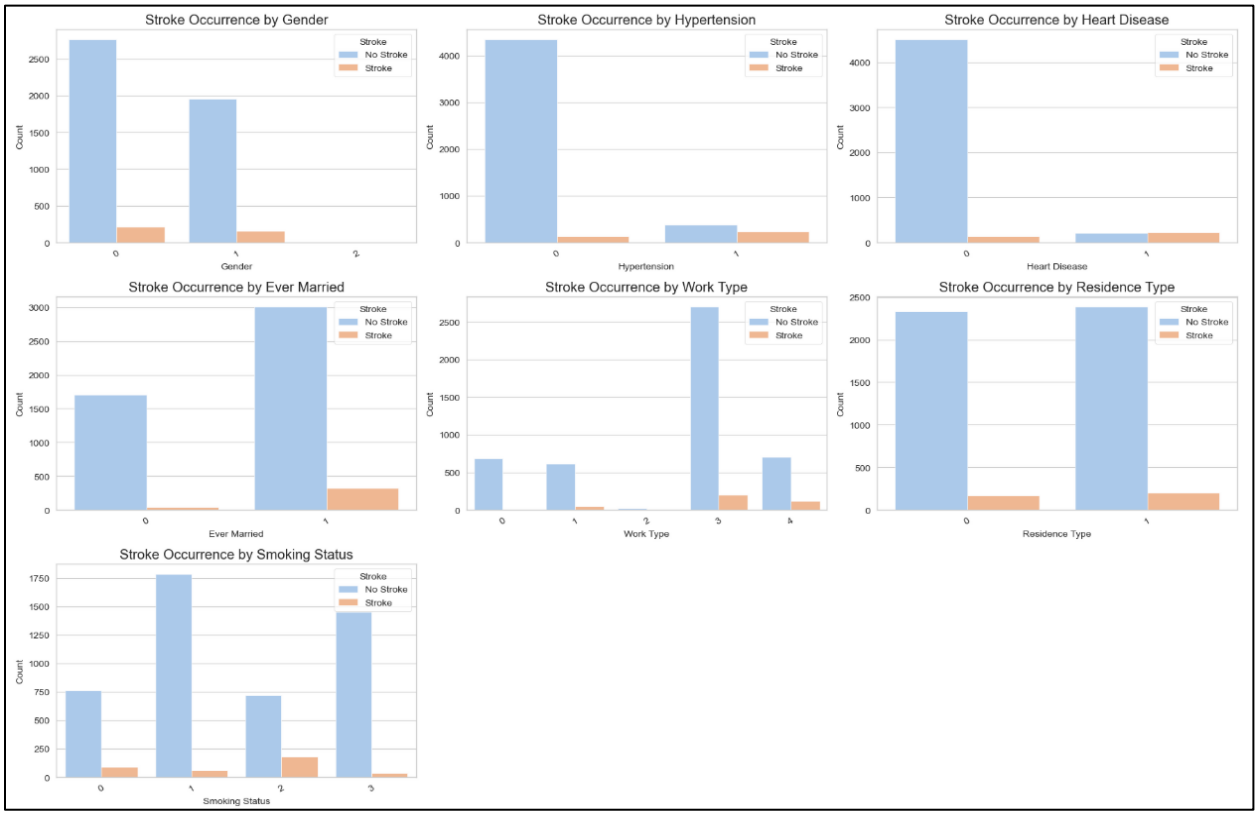


Fig. 13. Frequency distribution of various categorical features in the dataset.

3) Association Between Categorical Variables (Chi-Square Test): To examine whether there is a statistically significant association between categorical variables and stroke occurrence, a Chi-Square Test of Independence was performed for each categorical variable in the data set. According to Table III, variables such as "hypertension", "heart_disease", "ever_married", "work_type", and "smoking_status" show a statistically significant relationship with stroke. This suggests that these features might be useful predictors of stroke risk in further modelling or analysis. The variables "gender" and "residence_type" do not show a statistically significant association with stroke. This implies that these variables may not play a substantial role in determining stroke risk in this data set.

TABLE III. CHI-SQUARE STATISTICS

Variables	Chi-Square Statistics	p-value
gender	0.1907	0.9090
hypertension	1031.71	$2.2995e^{-226}$
heart_disease	1364.37	$1.1635e^{-298}$
ever_married	86.0216	$1.7798e^{-20}$
work_type	113.968	$1.0362e^{-23}$
residence_type	1.5532	0.2127
smoking_status	321.95	$1.7643e^{-69}$

The box plots in Fig. 14 illustrate the distribution of three continuous variables "age", "average', "avg_glucose_evel" and "BMI" between stroke outcomes (0 = no stroke, 1 = stroke). These visualisations help us identify how these variables differ between stroke and non-stroke patients. Stroke patients tend to be significantly older than non-stroke patients. The median age of stroke patients is around 75, while for non-stroke patients it is closer to 40. There is a wider age range among non-stroke patients, whereas stroke cases are more concentrated among the elderly. This suggests that age is a strong risk factor for stroke. Patients who have had a stroke tend to have higher average glucose levels. The median glucose level for stroke patients is above 150, compared to below 100 for non-stroke patients. The presence of high outliers in both groups indicates variability, but stroke patients generally show higher glucose distributions. This supports the idea that high blood sugar or diabetes may be associated with an increased risk of stroke. The distribution of stroke and non-stroke patients shows some overlap, but stroke patients have a slightly higher median BMI. There are more extreme BMI values (outliers) in the non-stroke group. Although the difference is less pronounced compared to age and glucose level, it still suggests a potential moderate relationship between obesity and stroke.

4) Correlation matrix: The correlation matrix in Fig. 15 shows Pearson's correlation coefficients between the numerical and binary variables in the data set. These coefficients range from -1 to 1. We are especially interested in how each feature correlates with the target variable stroke. The features with a stronger association with stroke ($r \geq 0.3$) are the features may have a more significant influence on stroke occurrence. Heart disease ($r = 0.52$) is the strongest positive correlation with stroke. This indicates that people with heart disease are more likely to suffer a stroke. Hypertension ($r = 0.45$) was positively

correlated, which means that high blood pressure can increase the risk of stroke. Meanwhile, age (r = 0.35) shows that older individuals are more prone to stroke. For the average glucose level (r = 0.30), elevated glucose levels are associated with a higher risk of stroke.

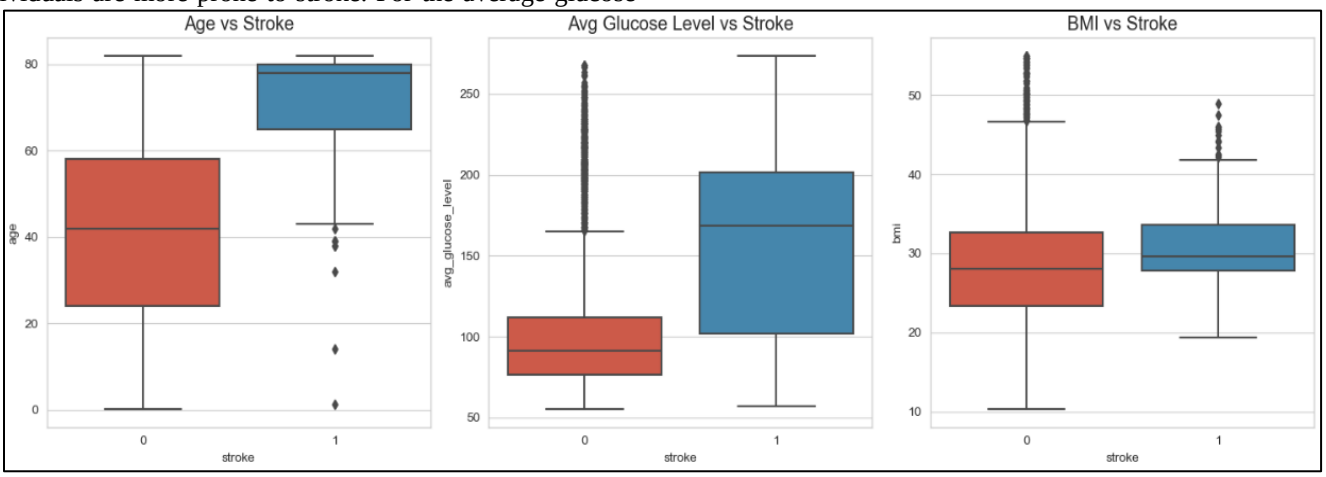


Fig. 14. Distribution of variables.

On the other hand, features with weak or No Significant Correlation to Stroke (r < 0.3) are variables do not show a strong linear relationship with stroke: Gender (r = 0.00), ever married (r = 0.13), type of work (r = 0.12), type of residence (r = 0.02), BMI (r = 0.08) and smoking status (r = -0.04). Although these features might still have some predictive power (especially in nonlinear models), they are not strongly linearly associated with stroke. Thus, through this heat map, we know that "heart_disease", "hypertension", "age", and "avg_glucose_level" are the most relevant features correlated with stroke. Features such as "gender", "ever_married", and "BMI" show little to no correlation, suggesting that they may play a less significant role in stroke prediction from a linear perspective.

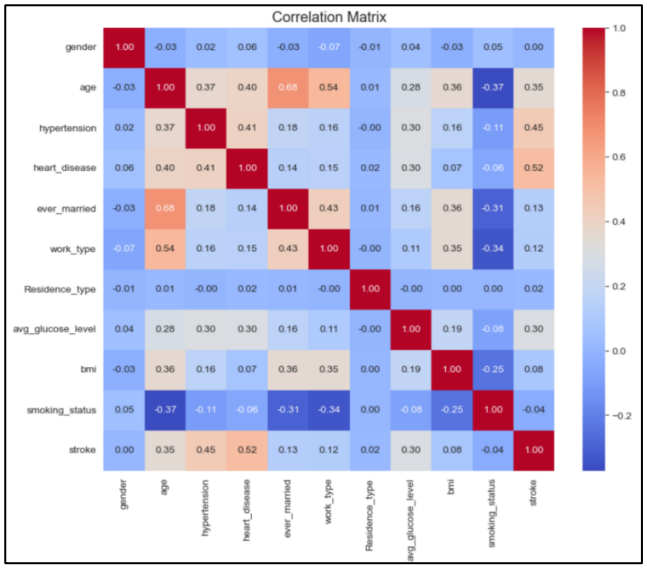


Fig. 15. Correlation matrix of variables.

5) Feature importance: To complement the correlation analysis, a RandomForestClassifier was trained to identify the most influential features in the prediction of stroke (Fig. 16). This model is effective because it captures not only linear relationships, but also non-linear patterns and interactions between features. According to the Random Forest model, the features with the highest scores (greater than 0.05) are age (0.29), avg_glucose_level (0.28), heart_disease (0.13), BMI (0.122), hypertension (0.12), and smoking_status (0.06). These variables contribute significantly to the predictions and are strongly associated with the probability of stroke.

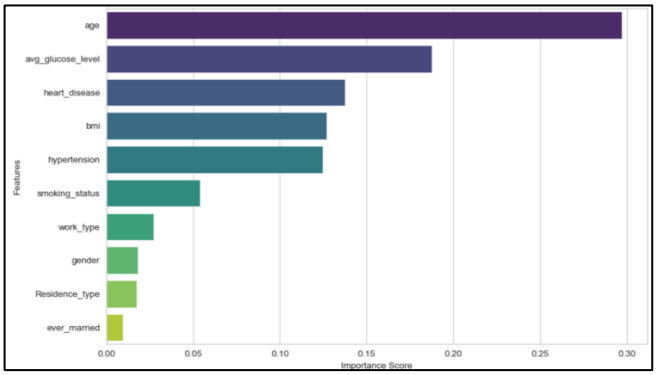


Fig. 16. Feature importance analysis (for EDA).

On the other hand, the features with lower importance scores (less than 0.05) are: work_type (0.025), gender (0.02), residence_type (0.019), ever_married (0.01). These variables may still play minor roles in prediction but are not considered strong indicators of stroke in this model compared to clinical factors such as age, "avg_glucose_level" or "heart_disease".

After reviewing the findings of the Chi-square test analysis, correlation matrix, and the result of the importance of features, we observed that both "gender" and "residence_type" consistently showed a low association with stroke. As a result, we decided to exclude these two variables from the final model to improve processing efficiency and eliminate weak predictors.

D. Split the Data for Training, Testing and Validation

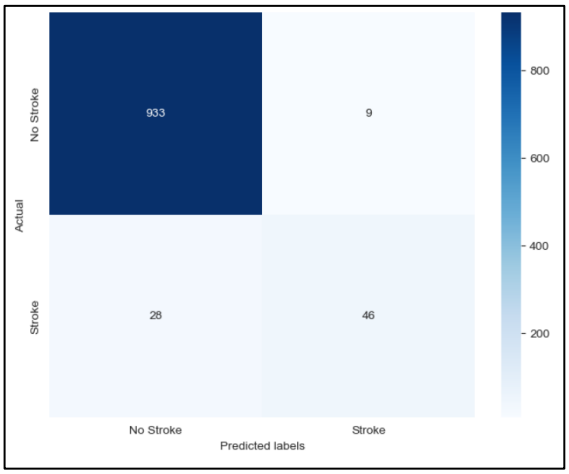
The data set has been pre-processed by separating the target variable (stroke) from the feature set and applying MinMax Scaling to normalise all feature values between 0 and 1. This ensures that all features contribute equally to the model. The scaled data set is then divided into training subsets (70%), validation (10%), and testing (20%) using the split train-test. First, 70% of the data are assigned to training, while the remaining 30% is temporarily stored. Then, this temporary set is further split into 10% for validation and 20% for testing. The final data set sizes are (3553, 10) for training, (508, 10) for validation, and (1016, 10) for testing, ensuring a well-balanced distribution for model training, tuning, and evaluation.

The SVM (Support Vector Machine) classifier is then evaluated on different training subset sizes to analyse its impact on classification performance. It first subsets the training data to specified sizes, applies MinMax Scaling, and then trains the SVM model on the subset. The model is tested in a separate test set and its accuracy, precision, recall, and F1 score are measured. The results show consistent performance across different subset sizes, with accuracy around 96.4%, indicating that the model generalises well with even smaller training data. The purpose of this experiment is to assess how the size of the dataset influences the performance of the model and determine whether a reduced dataset can achieve similar predictive accuracy, optimising computational efficiency.

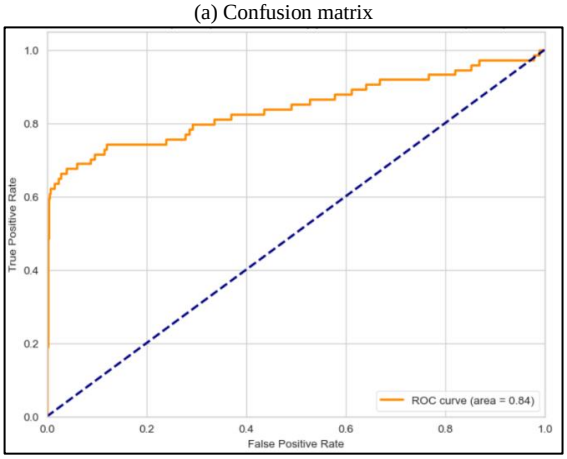
E. SVM

The Support Vector Machine (SVM) model achieved a high accuracy of 96.36%, indicating strong overall performance. The model effectively identified non-stroke cases, with a precision of 97% and a recall of 99%, meaning it rarely misclassified healthy individuals as stroke patients. However, its ability to detect actual stroke cases was moderate, with a precision of 83.64% and a recall of 62.16%, indicating that it correctly identified only 62.16% of actual stroke cases, while misclassifying some as non-stroke. The F1 score of 71.32% reflects a balance between precision and recall for stroke prediction.

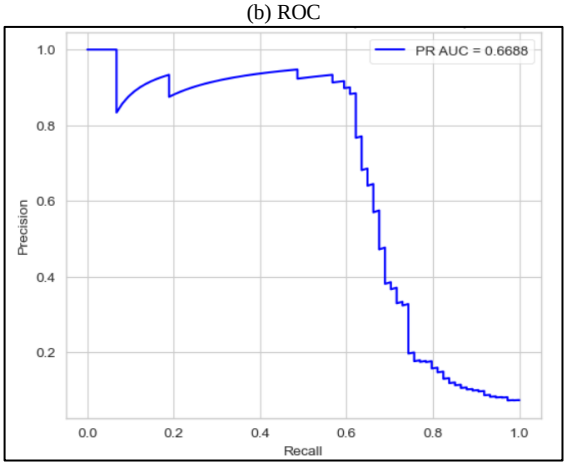
The confusion matrix Fig. 17(a) shows that the model correctly classifies 933 “No Stroke” cases and 46 “Stroke” cases, achieving high overall accuracy (96.36%). However, it misclassifies 28 actual stroke cases as “No Stroke”, leading to a low recall of 62.16%, meaning the model fails to detect nearly 38% of stroke cases, which is critical in medical applications. Although precision for stroke detection is high (83.64%), indicating that most predicted strokes are correct, the model's bias toward the majority class results in missed stroke cases. The Receiver Operating Characteristics (ROC) curve [Fig. 17(b)] for the Support Vector Machine (SVM) model demonstrates its ability to distinguish between stroke and nonstroke cases. The AUC score of 0.8418 indicates that the model has a good level of discrimination, which means that it correctly ranks stroke cases higher than non-stroke cases approximately 84.18% of the time. The Precision-Recall (PR) curve [Fig. 17(c)] evaluates the model performance in distinguishing stroke cases, especially when dealing with an imbalanced dataset. The PR AUC score of 0.6688 indicates that the Support Vector Machine (SVM) model achieves a moderate balance between precision (how many predicted stroke cases are actually strokes) and recall (how many actual stroke cases are correctly identified).



(a) Confusion matrix



(b) ROC



(c) Precision Recall

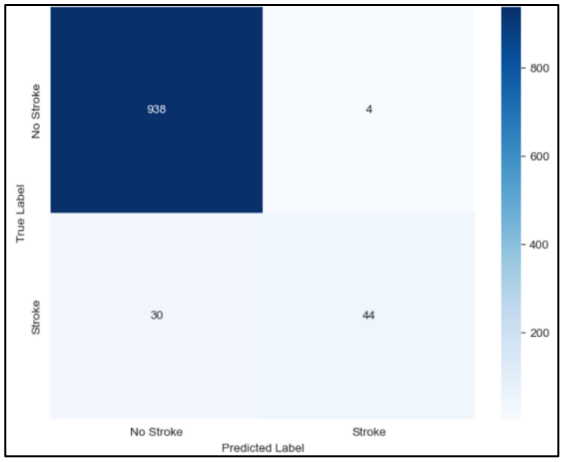
Fig. 17. SVM performance evaluation.

F. Random Forest

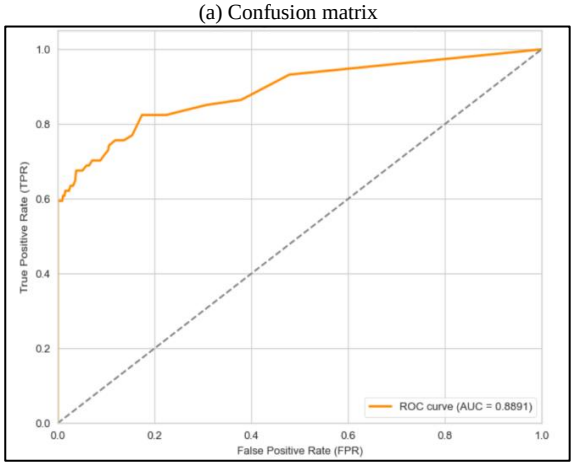
The Random Forest Classifier achieved a high accuracy of 96.75%, indicating a strong overall performance. The precision for stroke cases (1) is 93.62%, which means that most predicted stroke cases were correct. However, the recall is only 59.46%, suggesting that the model struggles to identify all actual stroke cases, leading to a significant number of false negatives. The F1

score of 72.73% reflects this trade-off between precision and recall.

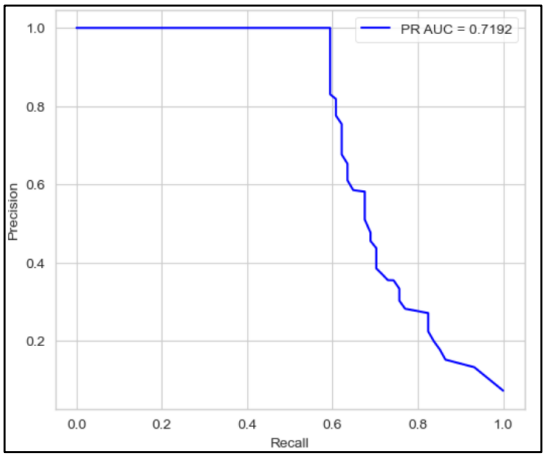
The confusion matrix in Fig. 18(a) shows that the model correctly classifies 938 "No Stroke" cases and 44 "Stroke" cases, achieving high overall accuracy (96.75%). It excels at correctly identifying nonstroke cases (0 class, with near-perfect recall) with only misclassified four non-stroke cases as stroke (False Positives). However, it misclassifies 30 actual stroke cases as "No Stroke", leading to a low recall of 59.46%, meaning that the model fails to detect nearly 41% of stroke cases, which is critical in medical applications. While precision for stroke detection is high (93.62%), indicating that most predicted strokes are correct, the model's bias toward the majority class results in missed stroke cases. The Receiver Operating Characteristics (ROC) curve [Fig 18(b)] for the Random Forest (RF) model demonstrates its ability to distinguish between stroke and nonstroke cases. The AUC score of 0.8891 indicates that the model has a good level of discrimination, meaning it correctly ranks stroke cases higher than non-stroke cases approximately 88.91% of the time. The Precision Recall (PR) curve [Fig. 18(c)] evaluates the performance in distinguishing stroke cases, especially when dealing with an unbalanced data set. The PR AUC score of 0.7192 indicates that the Random Forest (RF) model achieves a moderate balance between precision (how many predicted stroke cases are actually strokes) and recall (how many actual stroke cases are correctly identified).



(a) Confusion matrix



(b) ROC



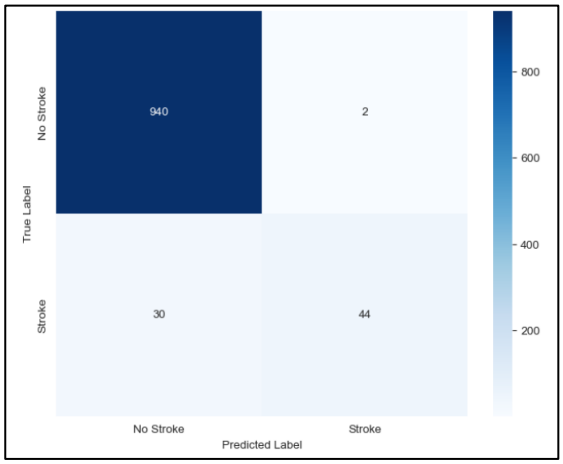
(c) Precision Recall

Fig. 18. Random Forest performance evaluation.

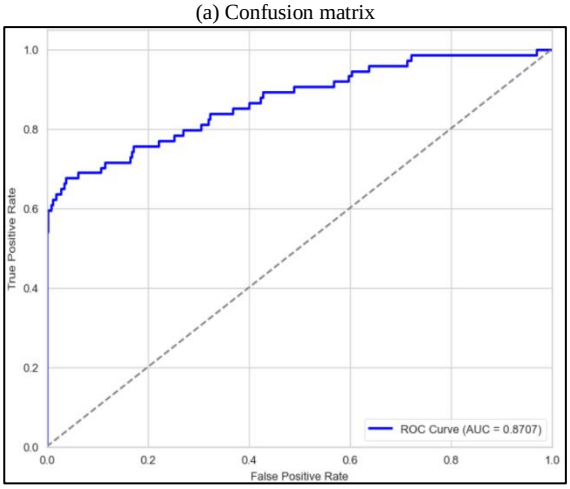
G. Hybrid Feature Augmentation Model (RF+SVM)

The feature-augmented SVM model, enhanced with Random Forest predictions, achieved a high test accuracy of 96.85%, with a precision of 95.65%, recall of 59.46%, and an F1-score of 0.7333. This indicates that the hybrid model is highly effective in correctly identifying stroke cases while minimising false positives. Although the recall is moderate, it reflects a significant improvement in capturing true stroke cases compared to standard models.

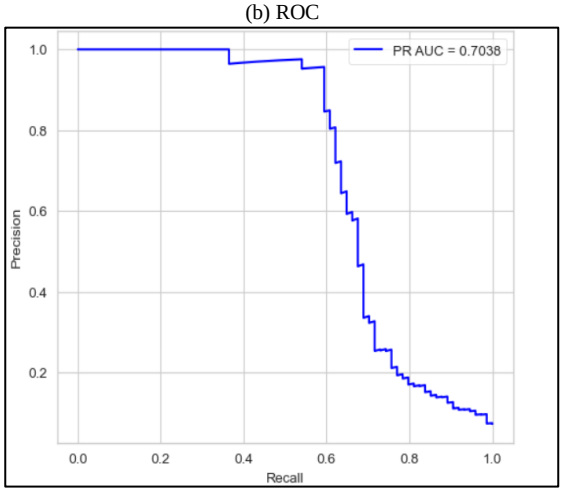
The confusion matrix in Fig. 19(a) shows that the model correctly classifies 940 cases of “No Stroke” and 44 cases of “Stroke”, achieving high overall accuracy (96.85%). It excels at correctly identifying non-stroke cases (0 class, with near-perfect recall), with only misclassifying two non-stroke cases as stroke (False Positives). However, it misclassifies 30 actual stroke cases as "No Stroke", leading to a low recall of 59.46%, meaning that the model fails to detect nearly 41% of stroke cases, which is critical in medical applications. While the precision for stroke detection is high (95.65%), indicating that most predicted strokes are correct, the model's bias toward the majority class results in missed stroke cases. The augmentation model (SVM+RF) in Fig. 19(b) achieves an AUC score of 0.8707, indicating strong classification performance. An AUC close to 1.0 suggests that the model is effective in distinguishing between stroke and non-stroke cases across various threshold levels. This score reflects a good balance between sensitivity (true positive rate) and specificity (1 false positive rate), supporting the reliability in medical decision-making where accurate classification is crucial. It means that it correctly ranks stroke cases higher than non-stroke cases approximately 87.07% of the time. The Precision-Recall (PR) Curve analysis [Fig. 19(c)] for the feature-augmented SVM model yields a PR AUC score of 0.7038, indicating a moderately strong ability to identify stroke cases, especially given the class imbalance. This score reflects how well the model maintains high precision and recall when predicting the minority class (stroke). A PR AUC closer to 1.0 suggests better performance in minimising false positives while capturing most true stroke cases. Thus, the value of 0.7038 demonstrates that the model is reasonably effective in prioritising stroke detection with fewer false alarms.



(a) Confusion matrix



(b) ROC



(c) Precision Recall

Fig. 19. Performance evaluation of the hybrid SVM + RF model.

H. Model Evaluation (RF, SVM, Feature Augmentation)

The bar chart in Fig. 20 compares the Support Vector Machine (SVM), the Random Forest (RF) and a hybrid Augmented SVM model on four performance metrics: accuracy, precision, recall, and F1 score. Although SVM shows the highest recall (0.622), it falls behind in precision and F1 score, indicating that it captures more positives, but also introduces more false alarms. Random Forest performs more consistently, especially in precision (0.917), but still lags slightly behind in recall and overall balance. The feature-augmented SVM, which combines the strengths of both SVM and RF, achieves the best overall performance with the highest accuracy (0.969), precision (0.957), and F1 score (0.733), while maintaining recall comparable to RF. This improvement demonstrates that integrating the high recall ability of SVM with the precision strength of Random Forest leads to a more robust and well-rounded hybrid model.

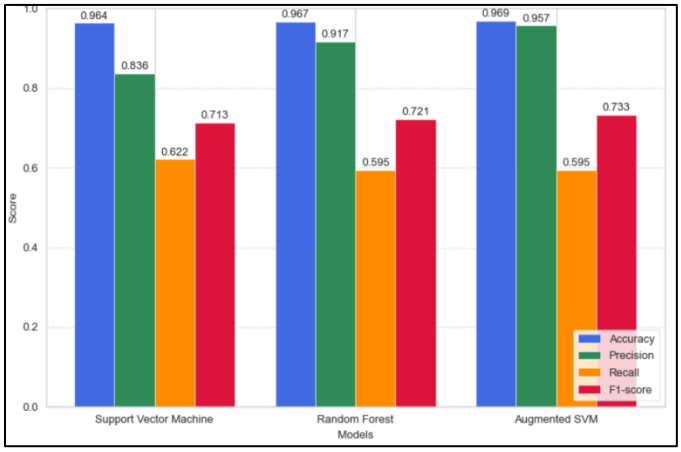


Fig. 20. Performance comparison between Random Forest, Support Vector Machine, and hybrid model (SVM+RF).

The confusion matrices for the SVM, Random Forest (RF), and Feature-Augmented Model (SVM+RF) in Fig. 21 illustrate how each model performs in classifying stroke and non-stroke cases. The SVM model correctly predicts 933 nonstroke cases and 46 stroke cases, while misclassifying 9 non-stroke cases as stroke and 28 stroke cases as non-stroke. Both the RF and Feature-Augmented models correctly predict 44 stroke cases and misclassify 30 stroke cases as non-stroke. However, RF correctly classifies 938 non-stroke cases and misclassifies 4 as stroke, while the Feature-Augmented model improves further by correctly predicting 940 non-stroke cases and misclassifying only 2. This demonstrates that combining the strengths of SVM and RF reduces the number of false positives (non-stroke cases wrongly classified as stroke), enhancing the model's precision by reducing unnecessary stroke predictions, leading to a more reliable and balanced classification system.

The comparison of the Precision Recall (PR) curve in Fig. 19 illustrates the trade-off between precision and recall for the SVM, Random Forest, and Feature-augmented model. Among the three, Random Forest achieves the highest PR AUC score (0.7192), indicating a more stable performance across all thresholds. Although the hybrid model shows superior point metrics (higher precision and F1 score at the default threshold), its slightly lower PR AUC (0.7038) suggests less consistent performance across varying thresholds. Similarly, the SVM has a PR AUC of 0.7101, reflecting moderate performance. In general, the hybrid model demonstrates strong effectiveness at the default classification threshold, while the Random Forest offers more reliable confidence across a range of threshold values.

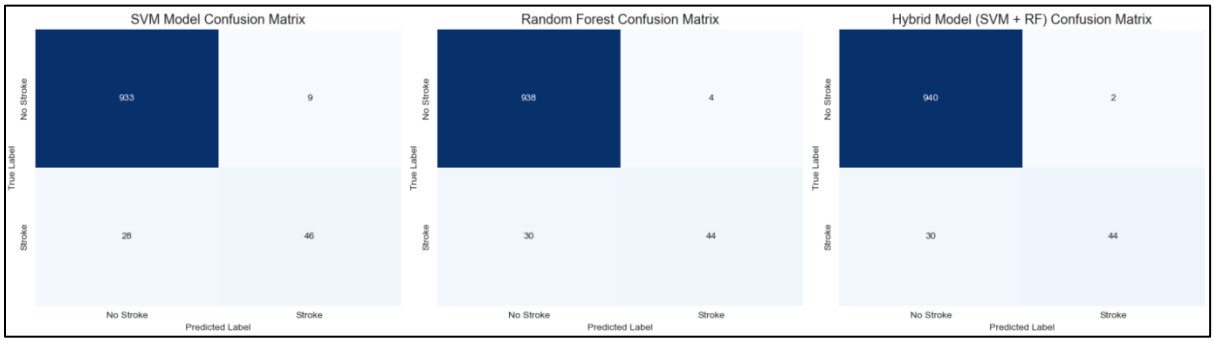
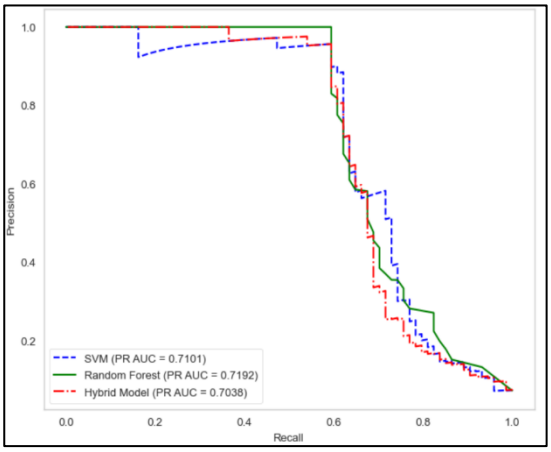
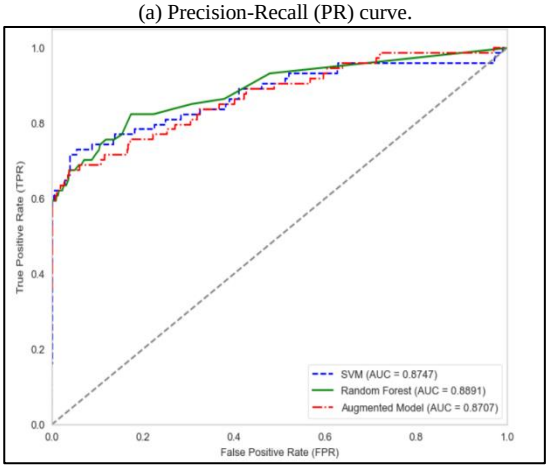


Fig. 21. Confusion matrices for the SVM, Random Forest (RF), and Feature-Augmented Model (SVM+RF).



(a) Precision-Recall (PR) curve.



(b) ROC AUC scores.

Fig. 22. Performance comparison for the SVM, Random Forest (RF) and hybrid model (SVM+RF).

The ROC AUC scores for the models are as follows: the Augmented (Hybrid) model achieved 0.8707, the Random Forest model achieved 0.8891, and the SVM model achieved 0.8747 (Fig. 22). These scores reflect the ability of each model to distinguish between stroke and non-stroke cases, with the Random Forest model showing the highest performance in this regard. The Augmented (Hybrid) model, although slightly lower in the ROC AUC compared to Random Forest, demonstrated improvements in other key evaluation metrics, such as precision and the F1 score, compared to the individual models. This suggests that the hybrid model effectively balances performance across different metrics, making it a viable alternative in stroke classification tasks.

I. Hyperparameter-Tuning Feature Augmented Model (SVM+RF)

1) Grid search: The feature augmented model was developed to improve model performance in predicting stroke cases using a VotingClassifier that combines Support Vector Machine (SVM) and Random Forest (RF) classifiers. Both models were optimised using GridSearchCV with 5-fold cross-validation and class weight adjustments to address class imbalance. The best SVM model used a linear kernel, C = 10, and class_weight = "balanced", while the best RF model used 200 estimators, max_depth = 10, min_samples_split = 5, and class_weight = "balanced_subsample". The final ensemble applied soft voting based on predicted probabilities. An optimal decision threshold (0.5805) was selected using the precision-recall curve to maximise the F1-score. The tuned model achieved an accuracy of 96.85%, a precision of 95.65%, a recall of 59.46%, and an F1 score of 73.33%. Furthermore, the model achieved a PR AUC of 0.7262 and a ROC AUC of 0.9031, indicating strong overall performance and improved ability to identify stroke cases while effectively managing the class imbalance.

2) Random search: Another way to perform hyperparameter tuning is to optimise Random Forest using RandomizedSearchCV with 5-fold cross-validation, and its predicted probabilities and class labels were appended as new features to the original dataset. These augmented data were used to train an SVM, also optimised via RandomizedSearchCV. Evaluated on a test set with an optimised for recall, the hybrid model achieved an accuracy of 97.05%, perfect precision (1.0000), a recall of 59.46%, and an F1 score of 0.7458. Despite slightly lower ROC AUC scores (0.8421) and PR AUC (0.6886) compared to the standalone Random Forest, the tuned hybrid approach significantly improved precision while maintaining reasonable recall, demonstrating its efficacy in balancing high precision with improved detection of stroke cases.

After comparing the results of the two hyperparameter tuning methods, we decided to proceed with the Random Search outcome, as it demonstrated superior classification performance compared to Grid Search. This decision aligns with the objective

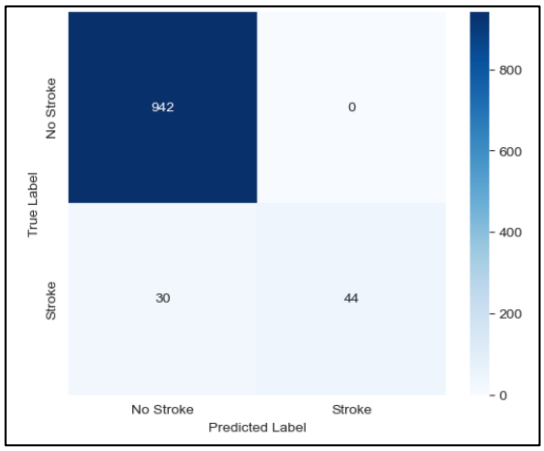
of this investigation, which is to improve the overall performance of the model. The tuned feature augmented model performance is saved for later development usage.

3) Performance evaluation: The confusion matrix in Fig. 23(a) shows that the model correctly classifies 942 "No Stroke" cases and 44 "Stroke" cases, achieving high overall accuracy (97.05%). It excels at correctly identifying non-stroke cases (0 class, with near-perfect recall) without misclassifying any non-stroke cases as stroke (False Positives). However, it misclassifies 30 actual stroke cases as "No Stroke", leading to a low recall of 59.46%, meaning the model fails to detect nearly 41% of stroke cases. Although precision for stroke detection is high (100%), indicating that most predicted strokes are correct, the model's bias toward the majority class results in missed stroke cases. The ROC AUC score for the hybrid model [Fig. 23(b)] after applying random search for hyperparameter tuning is 0.8421, indicating the strong ability to distinguish between stroke and non-stroke cases. The ROC curve rises steeply towards the top left corner, reflecting a high true positive rate (TPR) and a low false positive rate (FPR), suggesting effective classification performance. The Precision-Recall (PR) Curve analysis [Fig. 23(c)] for the feature-augmented SVM model resulted in a PR AUC score of 0.7038, indicating a moderately strong ability to identify stroke cases, particularly in the context of class imbalance. After tuning through Random Search, the model's PR AUC score decreased slightly decreased to 0.6886. This change reflects the model's performance adjustment after the tuning process, aiming for an improved balance between precision and recall.

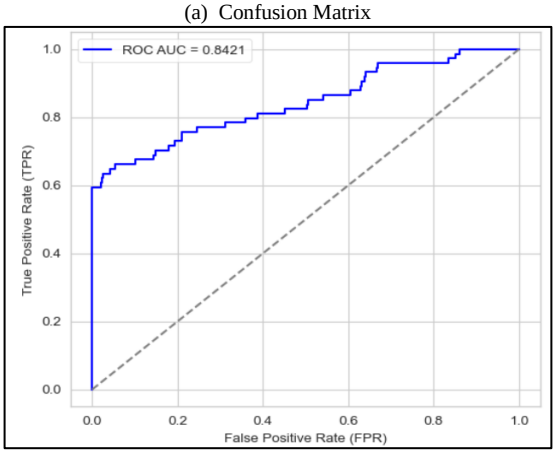
J. *Model Evaluation After Hyperparameter Tuning (SVM, RF, SVM+RF, Tuned SVM+RF)*

The bar chart in Fig. 24 compares the performance of four models: Support Vector Machine (SVM), Random Forest, Feature-Enhanced SVM and Tuned Feature Augmented SVM using accuracy, precision, recall, and F1 score. Among them, the tuned Augmented SVM achieved the highest accuracy (0.970), high precision (1.000), and F1 score (0.746), indicating a strong ability to correctly identify positive cases while balancing precision and recall. Although its recall (0.595) matches that of Random Forest and Feature Augmented SVM, its exceptional precision significantly increases its overall effectiveness. Random Forest showed high accuracy (0.967) and strong precision (0.917), but a lower recall (0.595) reduced its F1 score to 0.721. The standard SVM performed well with high accuracy (0.964) and better recall (0.622), resulting in an F1 score of 0.713. The enhanced SVM improved over the standard SVM with higher precision (0.957) and an F1 score (0.733), although its recall remained unchanged. Overall, the Tuned Augmented SVM demonstrated the best balance between metrics, highlighting the impact of augmentation and hyperparameter tuning.

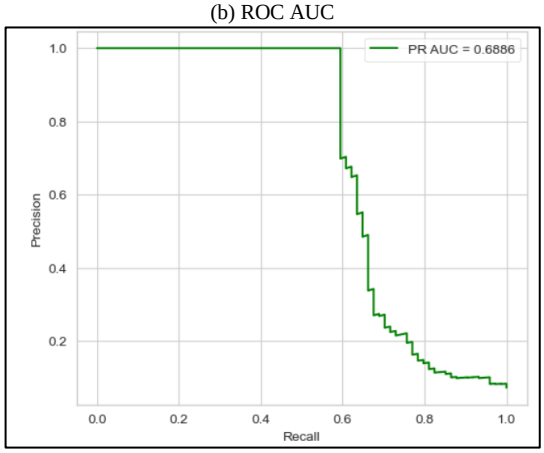
The confusion matrices of the four models reveal a consistent reduction in the number of false positives (cases where "No Stroke" was incorrectly classified as "Stroke") (Fig. 25). The SVM model had 9 false positives, whereas the Random Forest model reduced this to 4. The hybrid model (SVM + RF) further reduced the number to 2. The tuned hybrid model achieved zero false positives, correctly classifying all 942 "No Stroke" cases. Importantly, the number of correctly classified "Stroke" cases remained consistent at 44 across all models, and the number of False Negatives (Stroke cases misclassified as "No Stroke") remained at 30.



(a) Confusion Matrix



(b) ROC AUC



(c) PR AUC

Fig. 23. Performance evaluation after hyperparameter tuning.

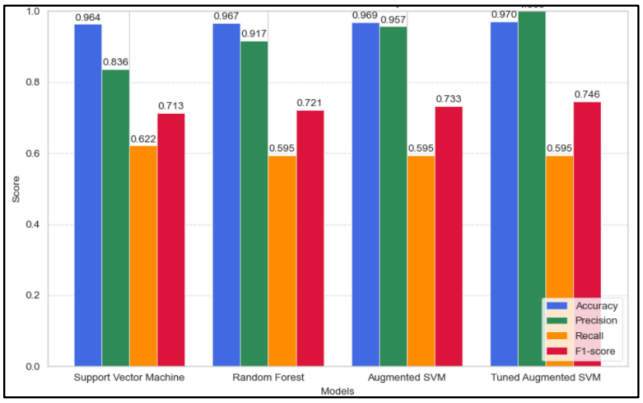


Fig. 24. Performance evaluation among four models.

According to Fig. 26(a), the tuned hybrid model achieved an AUC score of 0.842, slightly lower than the Random Forest (0.889) and SVM (0.875) models. However, it demonstrated a significant advantage by eliminating all false positives, as indicated in the confusion matrix, correctly classifying all 942 "No Stroke" cases. Despite a minor decrease in AUC, the tuned hybrid model maintained consistent performance in identifying true stroke cases, achieving a balanced outcome between sensitivity and specificity. In the performance comparison of Fig. 26(b), the Random Forest model achieved the highest PR AUC of 0.724, followed by the SVM with a PR AUC of 0.709, and the hybrid model with a PR AUC of 0.703. The tuned hybrid model achieved a PR AUC of 0.688, showing a slight drop compared to the other models.

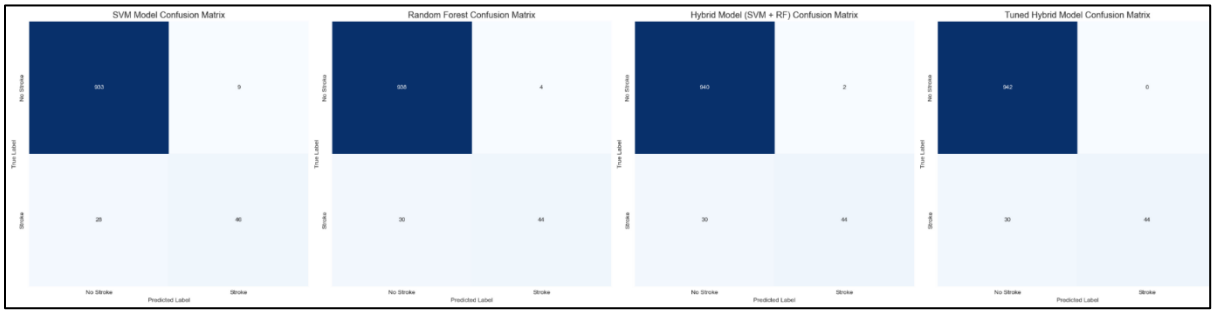
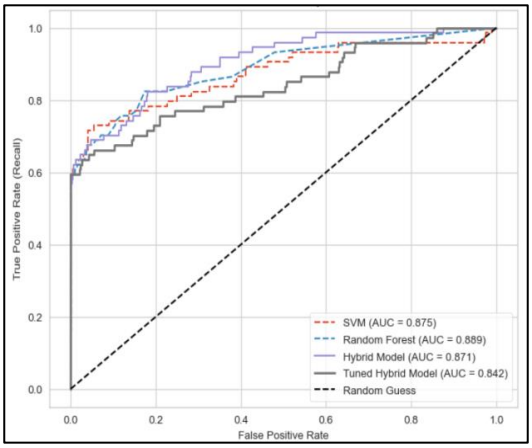
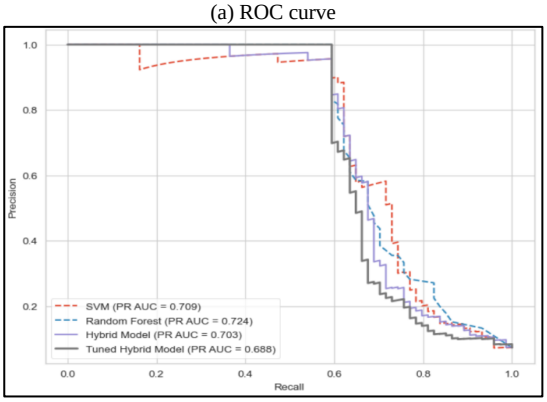


Fig. 25. Confusion matrices of four models.



(a) ROC curve



(b) Precision-Recall curve

Fig. 26. Performance evaluation (ROV curve and Precision Recall curve) for four models.

V. Discussion

In this project, the metrics used to evaluate the model performance included accuracy (overall correctness), precision (the proportion of predicted positives that are actually correct), recall (the proportion of actual positives correctly identified), F1-score (the harmonic mean of precision and recall), confusion matrix (detailed breakdown of predictions), ROC AUC (model's ability to distinguish between classes), and PR AUC (trade-off between precision and recall, particularly valuable for imbalanced datasets).

The confusion matrix highlights four key components: True Positive (TP), where the model correctly predicts a positive case; False Negative (FN), where the model correctly predicts a negative case for a positive instance; True Negative (TN), where the model correctly predicts a negative case; and False Positive (FP), where the model incorrectly predicts a positive case for a negative instance.

The standalone SVM achieved an accuracy of 96.36%, a precision of 83.64%, a recall of 62.16%, an F1 score of 71.32%, an ROC AUC of 0.8418, and a PR AUC of 0.6688, with a confusion matrix of 933 TN, 9 FP, 28 FN, and 46 TP. For Random Forest (RF), the performance improved with an accuracy of 96.75%, a precision of 93.62%, a recall of 59.46%, an F1 score of 72.73%, an ROC AUC of 0.8891, and a PR AUC of 0.7192, with a confusion matrix of 938 TN, 4 FP, 30 FN, and 44 TP. The feature-augmented model achieved 96.85% accuracy, 95.65% precision, 59.46% recall, 73.33% F1 score, 0.8707 ROC AUC, and 0.7038 PR AUC, with 940 TN, 2 FP, 30 FN and 44 TP.

The feature-augmented model demonstrated an improvement in accuracy, precision and F1-score, particularly in identifying stroke cases. Despite a slight decrease in the ROC AUC and PR AUC scores, the reduction in false positives (from 9 and 4 to 2) suggests that the model is more reliable, with fewer false alarms. This makes the model more robust for real-world deployment, especially where false positives could lead to unnecessary tests, anxiety, and increased healthcare costs.

To further enhance model performance, a Random Search hyperparameter tuning was applied. After tuning, the model achieved 97.05% accuracy, 100% precision, 59.46% recall, 74.58% F1-score, 0.8421 ROC AUC, and 0.6886 PR AUC. The confusion matrix showed 942 true negatives, 0 false positives, 30 false negatives, and 44 true positives.

The tuned feature-augmented model showed improvements in accuracy, precision and F1 score, although recall remained the same. The reduction of false positives to 0 and the slight increase in true negatives (from 940 to 942) indicate that the tuning process further optimised the model. This highlights the importance of minimising false alarms, particularly in medical applications, as false positives can result in unnecessary tests, increased anxiety, and additional healthcare costs.

This final tuned model provides a reliable and precise tool for stroke prediction, making it highly applicable in clinical environments where timely and accurate decision-making is essential. By reducing false positives and improving prediction reliability, the system is well suited for deployment in real-world applications, ensuring better health outcomes.

VI. Conclusions

In this project, a stroke prediction system is developed and evaluated using machine learning techniques. The hybrid model, combining Support Vector Machines (SVM) and Random Forest (RF), demonstrated a significant improvement over standalone models, achieving better accuracy, precision, and F1 score. The feature-augmented model, which incorporated additional features, further enhanced predictive performance, showing improved classification metrics, including a reduction in false positives and an increase in true negatives.

Through the process of hyperparameter tuning using Random Search, the model's performance was further optimised. The final tuned model achieved a high level of precision (100%), improved accuracy (97.05%), and maintained the same recall (59.46%) compared to the original feature-augmented model. Although there was a slight reduction in the ROC AUC and PR AUC, the ability to minimise false positives and improve true negatives makes it more reliable and suitable for real-world deployment in medical applications.

The practical implications of this project are significant, as it shows the potential of machine learning models to improve stroke prediction and aid healthcare providers in identifying high-risk patients. By reducing false alarms and increasing prediction accuracy, the model can help streamline healthcare processes, reduce unnecessary tests, and ultimately lower healthcare costs. This system has the potential to revolutionise stroke prediction by enabling early detection and intervention, thus significantly reducing the burden of stroke-related healthcare challenges globally.

Despite the strong performance, there are a few limitations. A major challenge is class imbalance, which is common in medical data sets, where positive stroke cases are significantly fewer than negative cases. This imbalance can affect recall, as seen with the relatively lower recall rate of 59.46%, suggesting that some true stroke cases are still being missed. There is also room for improvement in recall, which would ensure fewer missed stroke cases. Although high precision is critical, improving recall is equally important to ensure that at-risk patients are not overlooked.

To enhance the performance and utility of the stroke prediction system, several directions for future development can be considered.

- Incorporating additional features: Including more comprehensive patient data, such as laboratory test results, imaging data, or genetic markers, could improve the model's ability to detect subtle risk factors.
- Handling imbalanced datasets: Given the natural imbalance in stroke datasets (fewer stroke cases than non-stroke cases), applying techniques such as SMOTE (Synthetic Minority Oversampling Technique), ADASYN, Random Oversampling, or cost-sensitive learning can help improve recall without sacrificing precision.
- Exploring advanced ensemble techniques: Approaches like Gradient Boosting Machines (e.g. XGBoost, LightGBM) or stacking multiple models could potentially yield better performance by capturing complex patterns more effectively.

Funding

The APC of this paper publication is sponsored by INTI International University and Asia Pacific University.

Conflict of Interest

The authors declare that there is no conflict of interest.

References

[1] K. S. Tan and N. Venketasubramanian, "Stroke Burden in Malaysia," Cerebrovasc Dis Extra, vol. 12, no. 2, pp. 58–62, Mar. 2022, doi: 10.1159/000524271.

[2] S. Afzal et al., "Evaluating eating patterns and health status of undergraduate students majoring in human nutrition, Lahore, Pakistan: a cross-sectional study," Int J Adolesc Youth, vol. 30, no. 1, Dec. 2025, doi: 10.1080/02673843.2024.2448287.

[3] C. S. Nwosu, S. Dev, P. Bhardwaj, B. Veeravalli, and D. John, "Predicting Stroke from Electronic Health Records," in 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, Jul. 2019, pp. 5704–5707. doi: 10.1109/EMBC.2019.8857234.

[4] T. T. Tin, E. H. C. Sheng, L. S. Xian, L. P. Yee, and Y. S. Kit, "Machine learning classification of rainfall forecasts using Austin weather data," International Journal of Innovative Research and Scientific Studies, vol. 7, no. 2, pp. 727–741, Mar. 2024, doi: 10.53894/ijirss.v7i2.2881.

[5] J. El, "Historical Developments of Random Forest - Jari El - medium. Medium."

[6] T. T. Tin, L. S. Hock, and O. M. Ikumapayi, "Educational Big Data Mining: Comparison of Multiple Machine Learning Algorithms in Predictive Modelling of Student Academic Performance," International Journal of Advanced Computer Science and Applications, vol. 15, no. 6, 2024, doi: 10.14569/IJACSA.2024.0150664.

[7] W. Jeff, "Support Vector Machines." Accessed: Sep. 29, 2025. [Online]. Available: https://www2.isye.gatech.edu/~jeffwu/isye8813/SVM.pptx

[8] M. Shabil et al., "Association between hydrocarbon exposure and risk of stroke: a systematic literature review," BMC Neurol, vol. 25, no. 1, p. 71, Feb. 2025, doi: 10.1186/s12883-025-04083-x.

[9] P. Bentley et al., "Prediction of stroke thrombolysis outcome using CT brain machine learning," Neuroimage Clin, vol. 4, pp. 635–640, 2014, doi: 10.1016/j.nicl.2014.02.003.

[10] H.-L. Wang et al., "Automatic Machine-Learning-Based Outcome Prediction in Patients With Primary Intracerebral Hemorrhage," Front Neurol, vol. 10, Aug. 2019, doi: 10.3389/fneur.2019.00910.

[11] A. A. Abujaber et al., "Predicting 90-day prognosis for patients with stroke: a machine learning approach," Front Neurol, vol. 14, Dec. 2023, doi: 10.3389/fneur.2023.1270767.

[12] S. Rahman, M. Hasan, and A. K. Sarkar, "Prediction of Brain Stroke using Machine Learning Algorithms and Deep Neural Network Techniques," European Journal of Electrical Engineering and Computer Science, vol. 7, no. 1, pp. 23–30, Jan. 2023, doi: 10.24018/ejece.2023.7.1.483.

[13] H. Zhang, "Stroke Prediction Based on Support Vector Machine," Highlights in Science, Engineering and Technology, vol. 31, pp. 53–59, Feb. 2023, doi: 10.54097/hset.v31i.4812.

[14] Y. Wu and Y. Fang, "Stroke Prediction with Machine Learning Methods among Older Chinese," Int J Environ Res Public Health, vol. 17, no. 6, p. 1828, Mar. 2020, doi: 10.3390/ijerph17061828.

[15] V. Bandi, D. Bhattacharyya, and D. Midhunchakkravarthy, "Prediction of Brain Stroke Severity Using Machine Learning," Revue d'Intelligence Artificielle, vol. 34, no. 6, pp. 753–761, Dec. 2020, doi: 10.18280/ria.340609.

[16] M. S. Azam, M. Habibullah, and H. K. Rana, "Performance analysis of various machine learning approaches in stroke prediction," Int J Comput Appl, vol. 175, no. 21, pp. 11–15, 2020.

[17] Md. M. Islam, S. Akter, Md. Rokunojjaman, J. H. Rony, A. Amin, and S. Kar, "Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique," International Journal of Electronics and Communications Systems, vol. 1, no. 2, pp. 57–62, Dec. 2021, doi: 10.24042/ijecs.v1i2.10393.

[18] M. Alruily, S. A. El-Ghany, A. M. Mostafa, M. Ezz, and A. A. A. El-Aziz, "A-Tuning Ensemble Machine Learning Technique for Cerebral Stroke Prediction," Applied Sciences, vol. 13, no. 8, p. 5047, Apr. 2023, doi: 10.3390/app13085047.

[19] R. Ren, H. Luo, C. Su, Y. Yao, and W. Liao, "Machine learning in dental, oral and craniofacial imaging: a review of recent progress," PeerJ, vol. 9, p. e11451, May 2021, doi: 10.7717/peerj.11451.

[20] B. Khadka, "Enhancing cloud based software engineering with machine learning," Master of Engineering, Information Technology, Centria University of Applied Science, 2024.

[21] L. Yang et al., "MRI classification using semantic random forest with the auto-context model." Accessed: Sep. 29, 2025. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8611460/pdf/qims-11-12-4753.pdf

[22] M. Nirmala and V. Saravanan, "Clinical Implication of Machine Learning Based Cardiovascular Disease Prediction Using IBM Auto AI Service," Int J Res Appl Sci Eng Technol, vol. 10, no. 8, pp. 124–144, Aug. 2022, doi: 10.22214/ijraset.2022.46087.

[23] L. K. Ramasamy, S. Kadry, Y. Nam, and M. N. Meqdad, "Performance analysis of sentiments in Twitter dataset using SVM models," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 3, p. 2275, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2275-2284.

[24] N. Saxena et al., "Hybrid KNN-SVM machine learning approach for solar power forecasting," Environmental Challenges, vol. 14, p. 100838, Jan. 2024, doi: 10.1016/j.envc.2024.100838.

[25] D. Mustafa Abdullah and A. Mohsin Abdulazeez, "Machine Learning Applications based on SVM Classification A Review," Qubahan Academic Journal, vol. 1, no. 2, pp. 81–90, Apr. 2021, doi: 10.48161/qaj.v1n2a50.