JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 37 (2025) 476–483
Received: 1 July 2025
Accepted: 3 November 2025

Investigating Machine Learning Algorithms for Stroke Occurrence Prediction

Kazeem B. Adedeji[a,*], Titilayo A. Ogunjobi[a], Thabane H. Shabangu[b], Joshua A. Omowaye[a]

[a] *Department of Electrical and Electronics Engineering, Federal University of Technology, Akure, Ondo State, Nigeria*
[b] *Department of Electrical Engineering, Tshwane University of Technology, Pretoria, South Africa*

Abstract
Stroke is the leading cause of death and the principal cause of long term disability. Accurate prediction of stroke is highly valuable for early intervention of treatment. In this study, six (6) machine learning (ML) algorithms namely: Random Forest (RF) classifier, Decision Tree (DT) classifier, K-Nearest Neighbour (KNN) classifier, Support Vector Classifier (SVC), Logistic Regression (LR) and Stacking Classifier (SC) were trained on 10 stroke risk factors to determine the most precise model for predicting the risk of stroke occurrence. The primary contribution of this work is the development of a stacking method that achieves high performance, as measured by various metrics such as Area under Curve (AUC), precision, recall, F1-score, and accuracy. The experimental results indicate that the stacking classification outperforms other methods, with an AUC of 98.80%, F1-score of 95.18%, precision of 95.08%, recall of 95.41%, and accuracy of 95.25%. The results revealed that the stacking classifier achieves a high performance and outperforms the other methods. With the rapid evolution of machine learning, the clinical professionals, and decision-makers can use the established models to assess the corresponding risk likelihood.

Keywords: Accuracy; data processing; machine learning; stroke prediction

*Corresponding author
Email address: **kbadedeji@futa.edu.ng** (K. B. Adedeji)

1. Introduction

Stroke occurs due to the interruption of the flow of blood to a part of the brain as a result of blood clot. Globally stroke is one of the most severe diseases and it is directly responsible for a considerable number of death. According to the World Stroke Organization, 15 million people suffer a stroke each year out of these approximately 5 million people die as a result and another 5 million are left permanently disabled [1-5]. It is therefore considered as the leading causes of death and disability worldwide. It not only affects patients but also impacts their social environment, family, and workplace. Contrary to popular belief, stroke can happen to anyone, at any age, regardless of gender or physical condition. Each year, millions of stroke survivors have to adapt to a life with restrictions in daily activities. Many face problems such as memory, concentration, attention issues, speech difficulties, emotional problems, loss of balance, and difficulty swallowing [6]. Depending on the cause of stroke, stroke can be categorized into three; ischemic stroke, hemorrhagic stroke, and transient ischemic attack (TIA) as shown in Figure 1. In ischemic stroke, the arteries supplying blood to the brain completely become blocked. The hemorrhagic stroke occurs when an artery in the brain breaks leaks blood. As a result, the blood from that artery creates excess pressure in the skull and swells the brain, damaging brain cells and tissues. The TIA on the other hand is sometimes referred to as a mini stroke. It occurs when blood flow to the brain is blocked temporarily. While its symptoms are similar to those of hemorrhagic stroke, they typically disappear after a few minutes or hours when the blockage moves and blood flow is restored.

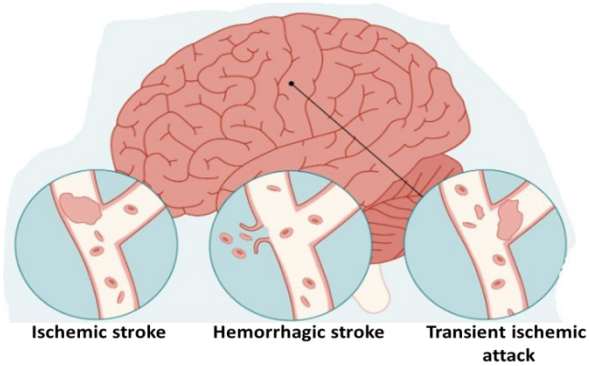

Figure 1: Classification of stroke [7].

The rising cost of hospitalization for stroke patients necessitates the development of advanced technologies to aid in clinical diagnosis, treatment, and prediction of clinical events. Early detection is critical for effective stroke treatment, making machine learning a vital tool. ML algorithms can learn complex patterns by integrating numerous variables from high-dimensional data. This capability enables health professionals to make informed clinical decisions and deliver accurate, rapid predictions [8-11]. To advance research in this area, this study employs six machine learning algorithms to predict the likelihood of stroke occurrence in individuals based on certain parameters such as age, gender, body mass index (BMI), smoking status, heart disease, marital status, hypertension, and average glucose level. The machine learning models are trained from the data generated from

these parameters. After training the models, their performance was evaluated based on the prediction accuracy, recall, F1-score, precision.

2. Review of Related Studies

Several studies have explored the use of machine learning algorithms for stroke prediction. Elias *et al*. [12] trained eight machine learning algorithms namely; Naive Bayes (NB), RF, LR, Stochastic Gradient Descent (SGB), Multilayer Perceptron (MLP), Majority Voting (MV) and Stacking Algorithm (SA) on the dataset from Kaggle to classify whether an individual will have a stroke or not. Out of all the algorithm used, Stacking classifier was the most efficient with a precision score of 97%, recall score of 97.8%, F1-score of 97.4%, accuracy score of 98%, and AUC score 98.9% respectively followed by Random forest classifier with a precision score of 95.5%, recall score of 97.6%, F1-score of 96.5%, accuracy score of 97% and AUC score of 98.6% then Majority voting classifier with a precision score of 92.3%, recall score of 93.8%, F1-score of 93.1%, accuracy score of 93% and AUC score of 93%. Gangavarapu *et al*. [13] used LR, DT, RF, KNN, SVC and NB to predict the occurrence of stroke. They collected the dataset from Kaggle which contains records of 5110 patient out of which 249 patient had stroke and 4861 does not. Due to the high level of imbalance, an under sampling technique was used to handle the imbalance. Among all the algorithms evaluated, NB gave the best performance with an accuracy of 82%, precision score of 79.2%, recall score of 85.7% and F1-score of 82.3%. This is followed by support vector classifier with an accuracy score of 80%, precision score of 78.6%, recall score of 83.8% and F1-score of 81.1%, RF performed the least with accuracy score of 73%, precision score of 72%, recall score of 73.5% and F1-score of 72.7%.

Tazin *et al*. [14] used a DT, RF, LR, voting classifier for stroke prediction. The dataset used contains 5110 records and 249 records has stroke while 4861 record does not. Due to the imbalance in the dataset, synthetic minority oversampling technique was used to balance the dataset. Among all the algorithms used for the prediction, RF performed best with an accuracy score of 96%, precision score of 95%, recall score of 97% and F1-score of 96% respectively. The efficiency of deep learning and machine learning models for predicting stroke attacks was examined in [15]. This study employed a number of classification models for classification tasks, including Ada Boost, Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine, RF, DT, LR, KN, SVM-Linear Kernel, Naive Bayes, and deep neural networks (3-layer and 4-layer ANN). According to the results, the RF classifier had the highest classification accuracy (among the machine learning classifiers) at 99%. In comparison to the three-layer ANN approach using the chosen features as input, the four-layer ANN, a three-layer deep neural network, also obtained an accuracy of 92.39%.

In Islam *et al*. [16], DT, RF, KNN, LR were trained for stroke prediction. The dataset used by them contains 5110 records of stroke patient. The result presented show that RF performed best with precision score, recall score, F1-score of 96%, 96% and 96% respectively**.**

3. Research Method

3.1. Data source and dataset description

Figure 2 shows the graphical representation of the steps employed in this study for predicting stroke occurrence. The dataset used in this study was obtained from Kaggle (an open source data repository), and it contains the records of 5110 patients, their age is between 25 years to 82 years and majority of the patient are females with a count of 2994 as compared to the male with a count of 2116. All the other attributes (10 of which served as input to the machine learning model) is described in Figure 3. Most of the features are categorical except for age, average glucose level and BMI which are numerical.

3.2. Data processing

This is a data mining technique which is used to clean, prepare the raw data order to make it more suitable for machine learning analysis. Since the dataset used in this study contains some categorical variables, it is then necessary to clean it in order for the models to produce a more accurate output. The data preprocessing steps include:

1) Data cleaning: Data cleaning was not necessary for the dataset set used in this study as there were no missing values, no null values and no Nan values as seen in Figure 4.
2) Data transformation: Categorical data were encoded using label encoding. Features were scaled using standardization.
3) Data balancing: SMOTE was used to address class imbalance, ensuring an equal distribution of stroke and no-stroke instances. Figure 5 and Figure 6 show the effect before and after applying SMOTE.

By learning to prioritise the majority class, the classifier may produce biased predictions if it is trained on an unbalanced dataset. This could lead to overfitting and decreased recall, which are frequent outcomes of imbalanced data categorisation tasks. We use SMOTE to enhance the number of minority class samples in order to allay this worry. This ensures an equal distribution of stroke and no-stroke instances with 50% stroke and 50% non-stroke observations, as shown in Figure 6. The SMOTE was applied after splitting the data into training and testing sets to prevent data leakage. We shuffled and shook the dataset once it had been balanced. We transformed our category values into a new categorical column and assigned a binary value of "1" or "0" in order to provide the data to our classification model. In this instance, the labels "1" and "0" stand for stroke class and non-stroke class, respectively.

3.3. Exploratory data analysis

Exploratory Data Analysis (EDA) was employed to examine the datasets to summarize their key characteristics

using statistical graphics and other visualization techniques. It also enhances understanding of the data and detects patterns that might not be immediately apparent from merely examination. EDA included visualizing the distribution of various attributes and their relationship with stroke. Figure 7 and Figure 8 show the distribution of the attributes in the dataset and correlation among the features.

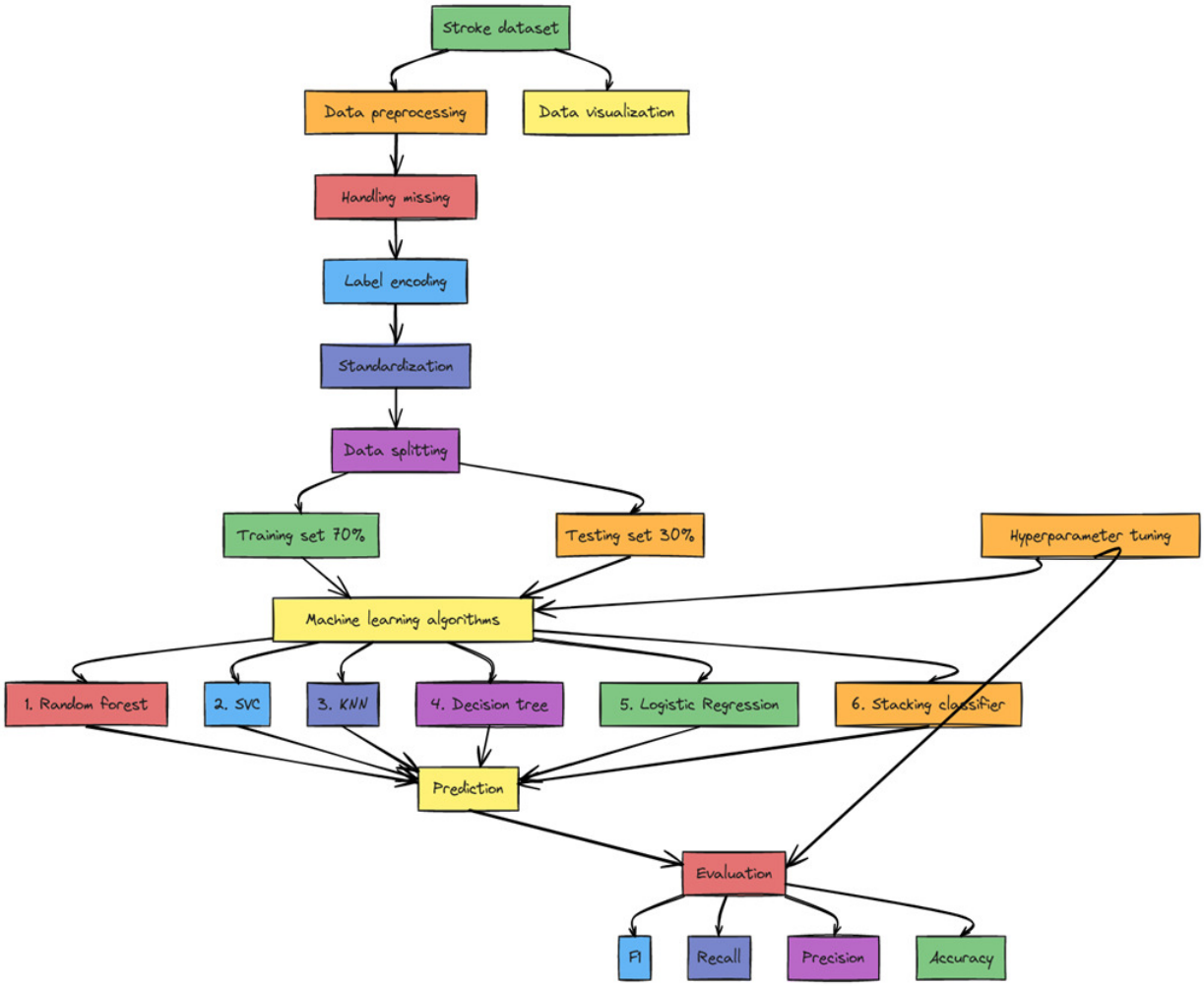

Figure 2: Activity diagram of the research method.

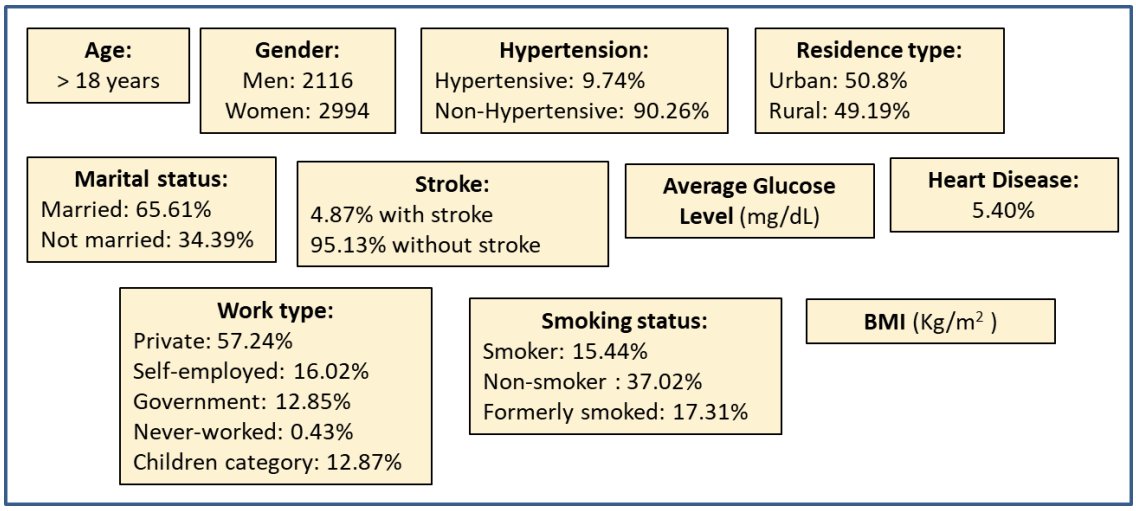

Figure 3: Feature attributes from the dataset.

3.4. The machine learning models

1) Logistic Regression: This predicts the probability of the target variable belonging to a certain class based on the values of the input feature (such as age, BMI, avg_glucose_level) according to (1).

$$\log_b\left(\frac{p}{1-p}\right) = \beta_0 + \beta_0 X_1 + ... + \beta_n X_n \quad (1)$$

where $p=P(Y=1)$ is the probability of an instance belonging to the stroke class, X_i are the input features, and β_i are the coefficients.

```
In [7]: stroke.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   work_type          5110 non-null   int32
 1   smoking_status     5110 non-null   int32
 2   gender             5110 non-null   int64
 3   ever_married       5110 non-null   int64
 4   Residence_type     5110 non-null   int64
 5   age                5110 non-null   float64
 6   hypertension       5110 non-null   int64
 7   heart_disease      5110 non-null   int64
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                5110 non-null   float64
 10  stroke             5110 non-null   int64
dtypes: float64(3), int32(2), int64(6)
memory usage: 399.3 KB
```

Figure 4: No missing or null value present in the dataset.

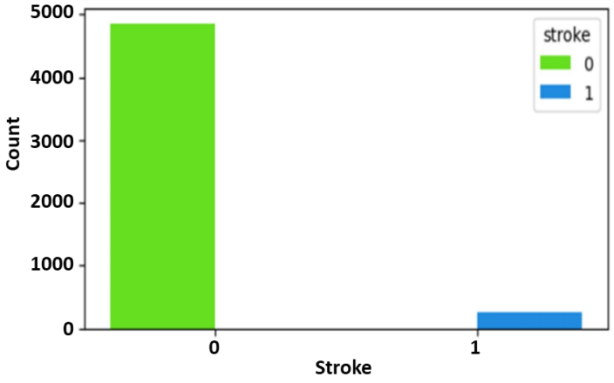

Figure 5: The count of stroke occurrence in the dataset before applying SMOTE.

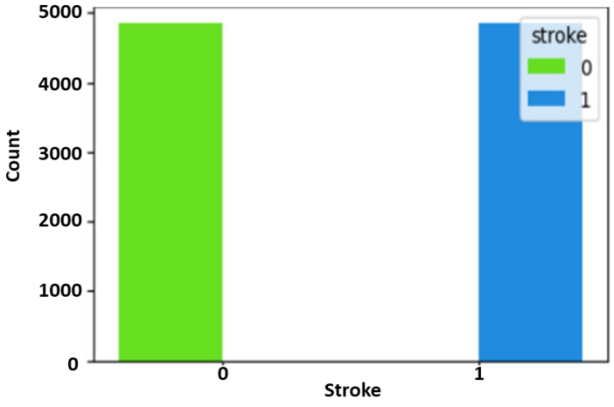

Figure 6: The count of stroke occurrence in the dataset after it was balanced with SMOTE.

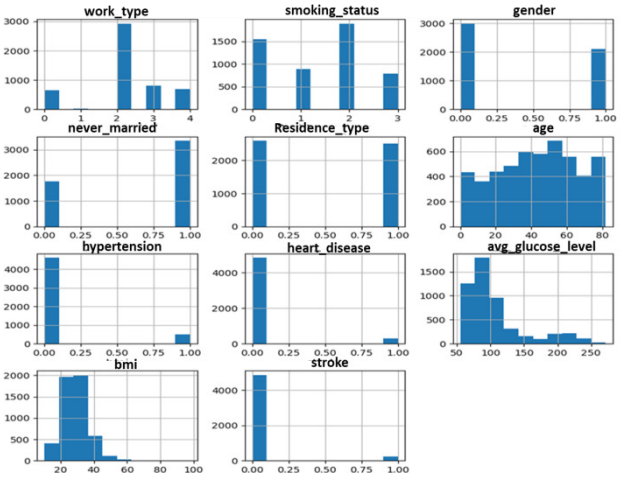

Figure 7: The distribution of all variables in the dataset.

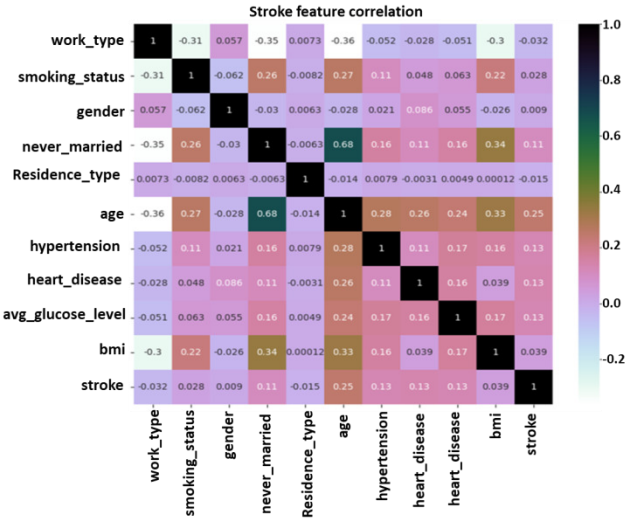

Figure 8: A heatmap showing the correlation among the stroke attributes.

2) Support Vector Classifier: SVCs work by finding a N-dimensional hyper plane which can distinguish N-dimensional data points. The hyper plane is the line that best separates the data points with the largest margin which is the distance between the hyper plane and the closest data points (support vectors). When the data is not linearly separable, a kernel function was used. The kernel function transforms the data into a higher-dimensional space, where the data may be linearly separable.
3) Decision Tree: It works by recursively splitting the training dataset into smaller subsets based on the most significant feature that provides the most information gain or decrease in impurity.
4) Random Forest: Random forests are an ensemble of decision trees. Each tree is trained on a random subset of the data, and the final prediction is made by averaging the predictions of all trees.
5) KNN: The idea behind KNN is to identify the K nearest data points (i.e., neighbors) to the data point to be classified, and use the class labels of these neighbors to predict the class label of the data point.

6) Stacking Classifier: This is an ensemble learning technique that involves combining multiple classification models via a meta-classifier. Thereafter the output of several individual classifiers are combined and then fed into a meta-classifier to make the final prediction.

3.5. Performance Evaluation

The performance of all the classifier was accessed based on the following metrics;

1) Accuracy (A): This is the ratio of correctly predicted instances (both positive and negative) to the total instances. This was estimated using (2).

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

where TP is the true positive, that its, the number of positive instances correctly identified by the model, FP is the false positive (the number of negative instances incorrectly identified as positive by the model), TN is the True Negative (the number of negative instances correctly identified by the model), and FN is the false negative (the number of positive instances incorrectly identified as negative by the model).

2) Precision (P): This is the ratio of correctly predicted positive instances to the total predicted positive instances. This was computed using (3).

$$P = \frac{TP}{TP + FP} \quad (3)$$

3) Recall (R): This is the ratio of correctly predicted positive instances to the total actual positive instances. This was computed using (4)

$$R = \frac{TP}{TP + FN} \quad (4)$$

4) F-measure (F1-Score): This is the harmonic mean of P and R, providing a balance between the two. This was estimated using (5).

$$F - measure = \frac{2 \times P \times R}{P + R} \quad (5)$$

5) ROC-AUC Curve: The Receiver Operating Characteristic (ROC) is a graphical representation of a classifier's performance across all classification thresholds. The Area Under the Curve (AUC) represents the degree or measure of separability between classes.
The ROC curve plots True Positive Rate (Recall) against False Positive Rate (FPR). The false positive rate is calculated using (6).

$$FPR = \frac{FP}{TN + FP} \quad (6)$$

3.6. Experimental Setup

The simulations were carried out in Python 3.0. The data processing and evaluation were implemented by extension packages including NumPy, Pandas, and Scikit-learn. In this study, 10-cross validation was applied to assess the models' efficiency in the balanced dataset. For the DT, the max-depth of 16 was used. The minimum number of instances per leaf node was set to the default value and the minimum sample split set to 2 using the Gini criterion. For the RF classifier, the max depth of 70 was used; min sample split set to default and min sample leaf set to 4. For the k-NN classifier, we set $k = 10$. Also, the Euclidean distance is a widely used distance metric and was adopted in this study. The GridSearchCV in the Scikit-learn was used to optimize the LR and SVC performance. For the implementation of the stacking model, four base classifiers were combined. Each of the classifiers was trained and tested to predict stroke considering both binary classifications. All experiments were carried out on a HP ProBook running Windows 10, a 64-bit operating system. The processor was an Intel Core i7 3.60 GHz CPU equipped with 8 GB of RAM.

4. Results and Discussion

Figure 9 shows the training score recorded by the six algorithms. This bar chart compares the training scores of various machine learning models applied to the stroke dataset. The Stacking classifier achieved the highest training score of 99.86%, followed closely by RF with 99.07%. LR had the lowest score of 81.07%, indicating its lower performance in capturing patterns during training compared to other models.

Figure 10 illustrates the accuracy scores of the six classifiers when predicting stroke occurrences. Stacking had the highest accuracy at 95.23%, indicating robust performance. SVC and Random Forest also performed well, with scores of 94.03% and 93.42%, respectively. Logistic Regression had the lowest accuracy at 79.97%, showing that it may not be the best choice for this dataset compared to the other classifiers.

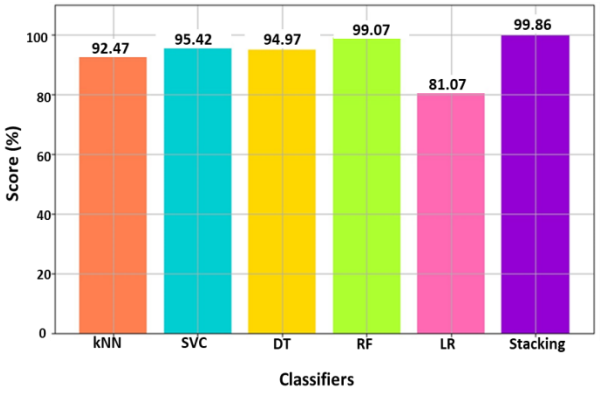

Figure 9: Comparison of the training score for the trained classifiers.

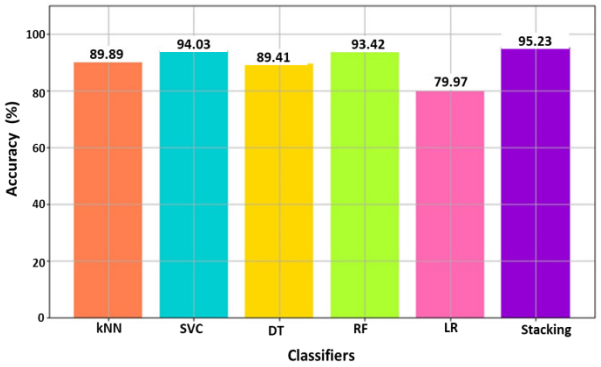

Figure 10: Comparison of the accuracy score for the trained classifiers.

In Figure 11, the precision score shows how well each classifier identified positive cases of stroke without including too many false positives. SVC and Stacking performed best, with precision scores of 95.64% and 95.08%, respectively, indicating strong precision in their predictions. Random Forest also had a high precision of 92.1%. Logistic Regression had the lowest precision score of 79.06%, suggesting that it may misclassify more non-stroke cases as strokes compared to other models.

Figure 12 shows the recall score recorded by the six algorithms. The KNN model achieved the highest recall score of 99.19%, followed by Random Forest and Stacking with scores of 95.31% and 95.28%, respectively. LR had the lowest recall score of 81.72%, suggesting it misses a higher proportion of true stroke cases compared to other models. These results highlight the superior recall performance of KNN, RF, and Stacking classifiers in detecting strokes. Figure 13 shows a comparison of the F1-scores for six different machine learning models. As shown in Figure 13, the Stacking classifier achieved the highest F1-score of 95.18%, indicating it had the best balance between precision and recall for this stroke dataset. Random Forest also performed well with an F1-score of 93.68%, followed by SVC at 94.07%. LR had the lowest score at 80.37%, showing relatively poorer performance compared to the others.

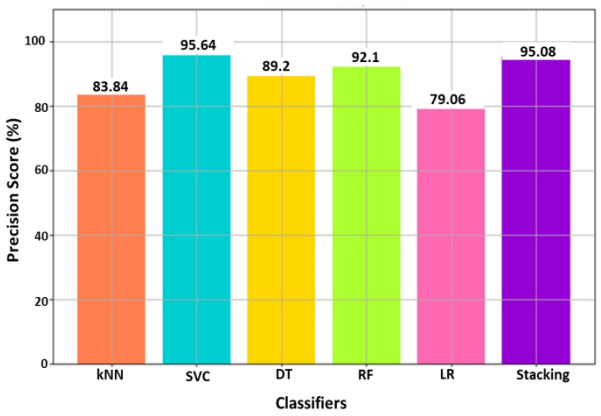

Figure 11: Comparison of the precision score for the trained classifiers.

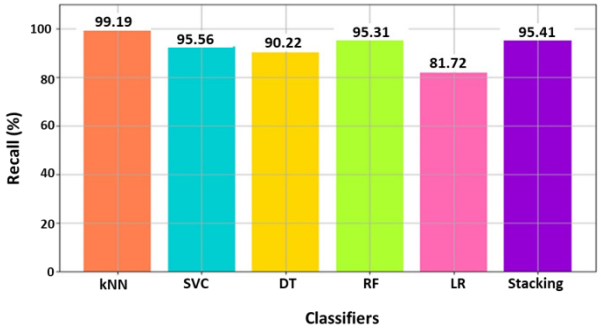

Figure 12: Comparison of the recall score for the trained classifiers.

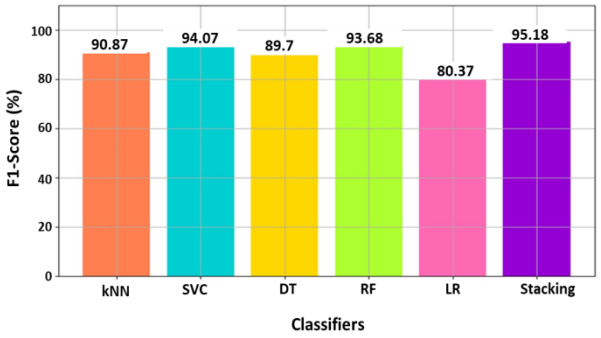

Figure 13. Comparison of the F1-score for the trained classifiers.

Figure 14 shows the AUC score recorded by the six algorithms. Stacking classifier performs the best with an AUC score of 98.8%, followed closely by SVC at 98.58%. KNN, Random Forest, and Decision Tree also performed well, with scores in the 93%-96% range. Logistic Regression has the lowest performance, with an AUC score of 88.35%.

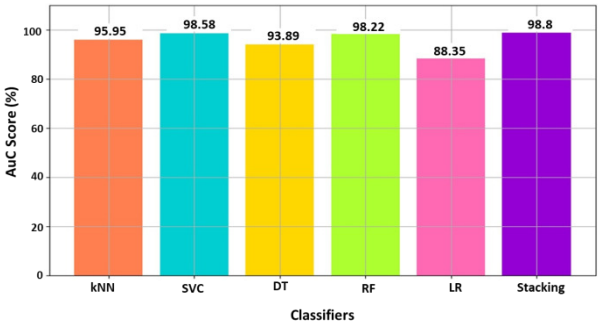

Figure 14: Comparison of the AUC score for the trained classifiers.

Table 1 shows the results of the precision, recall and F1-score for each class sample. As can be observed in Table 1, the precision of the model is superb for both the stroke and Non-stroke class sample. One notable observation is that the model was able to handle the minority class (class 0) since the class imbalance have been addressed.

Table 1: Precision, recall and F1-score per class sample.

Method	Precision (%)		Recall (%)		F1-score (%)	
	class 0	class 1	class 0	class 1	class 0	class 1
RF	91.82	92.1	95.21	95.31	93.52	93.68
DT	89.1	89.2	90.11	90.22	89.61	89.7
KNN	83.51	83.84	99.02	99.19	90.72	90.87
SVC	95.13	95.64	95.48	95.56	94.0	94.07
LR	79.00	79.06	81.69	81.72	80.21	80.37
SC	95.01	95.08	95.38	95.41	95.11	95.18

Class 0: (Non-stroke class); class 1: (Stroke class)

Table 2 presents the summary of the performance of the six classifiers. Based on the results from the Table 2, it is evident that the stacking classifier outperforms other models in predicting stroke occurrence, achieving the highest testing accuracy score of 95.2348%. The support vector machine also performed well with a high accuracy score of 94.0349%. Notably, the KNN classifier had the lowest false negative rate, indicating its effectiveness in correctly identifying individuals at high risk of stroke. The random forest classifier also demonstrated strong performance with an accuracy score of 93.4178% and a high AUC score of 98.2212. Conversely, the logistic regression classifier had the lowest accuracy score of 79.967%, showing it was less effective compared to the other models. In summary, the stacking classifier stands out as the best algorithm for predicting stroke.

Table 2: Summary of the overall performance of the classifiers.

Classifiers	Accuracy	AUC	TP	FP	TN	FN
KNN	89.88	95.95	1468	283	1154	12
SVM	94.03	98.59	1381	63	1362	111
DT	89.4	93.89	1346	163	1262	146
RF	93.41	98.22	1422	122	1308	70
LR	79.96	88.35	997	264	947	223
SC	95.2	98.8	1373	71	1405	66

5. Conclusions and Future Study

Stroke is a leading cause of death and the principal cause of long term disability. The negative effect of stroke has led to further research efforts in the use of artificial intelligence for stroke occurrence prediction. Early recognition of vital signs is valuable for stroke prediction which will promote a healthy life. This study has demonstrated the potential of machine learning algorithms, particularly the stacking classifier, in predicting stroke risk with remarkable accuracy. Unlike other studies, in this study, 10 stroke risk factors were used to train several machine learning classifiers for predicting the risk of stroke occurrence. The experimental results indicate that the stacking classifier outperforms other methods, with an AUC of 98.80%, F1-score of 95.18%, precision of 95.08%, recall of 95.41%, and accuracy of 95.25%. The result revealed that the stacking classifier achieves a high performance and outperforms the other methods. With the rapid evolution of machine learning, the clinical professionals, and decision-makers can use the established models to assess the corresponding risk likelihood. It is important to note that this research primarily relies on text-based data, which constrains the model's ability to fully capture the multifaceted nature of stroke risk factors. Further study will be to transcend these limitations by integrating brain imaging data, such as MRI or CT scans to enhance the model's predictive power and enable the identification of specific stroke types.

References

[1] About stroke by cdc. https://www.cdc.gov/stroke/index.html. [24.10.2024].

[2] E. S. Donkor, Stroke in the 21st century: a snapshot of the burden, epidemiology, and quality of life, Stroke Research and Treatment 2018 (2018) 1-10, https://doi.org/10.1155/2018/3238165

[3] R. O. Akinyemi, B. Ovbiagele, O. A. Adeniji, F. S. Sarfo, F. Abd-Allah, T. Adoukonou, O. S. Ogah, Naidoo, A. Damasceno, R. W. Walker, A. Ogunniyi, Stroke in Africa: profile, progress, prospects and priorities, Nature Reviews Neurology 17 (2021) 634-656, https://doi.org/10.1038/s41582-021-00542-4

[4] J .N. Fernandes, V. E. Cardoso, A. Comesaña-Campos, A. Pinheira, Comprehensive review: Machine and deep learning in brain stroke diagnosis, Sensors 24 (2024) 1-27, https://doi.org/10.3390/s24134355

[5] S. Gupta, S. Raheja, Stroke prediction using machine learning methods, Proceedings of the 12th IEEE International Conference on Cloud Computing, Data Science and Engineering (2022) 553-558, https://doi.org/10.1109/Confluence52989.2022

[6] B. Delpont, C. Blanc, G. Osseby, M. Hervieu-Bègue, M. Giroud, Y. Béjot, Pain after stroke: A review, Revue Neurologique 174 (2018) 671–674, https://doi.org/10.1016/j.neurol.2017.11.011.

[7] Healthline. Everything you need to know about stroke. https://www.healthline.com/health/stroke [06.05.2025].

[8] M. Javaid, A. Haleem, R. P. Singh, R. Suman, S. Rab, Significance of machine learning in healthcare: Features, pillars and applications, International Journal of Intelligent Networks 3 (2022) 58-73, https://doi.org/10.1016/j.ijin.2022.05.002.

[9] R. R. Kothinti, Deep learning in healthcare: Transforming disease diagnosis, personalized treatment, and clinical decision-making through AI-driven innovations, World Journal of Advanced Research and Reviews 24(2024) 2841-2856, https://doi.org/10.30574/wjarr.2024.24.2.3435

[10] S. Rani, R. Kumar, B. S. Panda, R. Kumar, N. F. Muften, M. A. Abass, J. Lozanović, Machine Learning-Powered Smart Healthcare Systems in the Era of Big Data: Applications, Diagnostic Insights, Challenges, and Ethical Implications, Diagnostics 15 (2025) 1-40, https://doi.org/10.3390/diagnostics15151914

[11] M. Nasiruddin, S. Dutta, R. Sikder, R. Islam, A. A. Mukaddim, M. A. Hider, Predicting heart failure survival with machine learning: Assessing my risk, Journal of Computer Science and Technology Studies 6 (2024) 42-55, https://doi.org/10.32996/jcsts.2024.6.3.5

[12] D. Elias, M. Trigka, Stroke risk prediction with machine learning techniques, Sensors 22 (2022) 1-13, https://doi.org/10.3390/s22134670

[13] S. Gangavarapu, G. L. A. Kumari, Analyzing the performance of stroke prediction using ML classification algorithms, International Journal of Advanced Computer Science and Applications 12 (2021) 145-149, https://doi.org/10.14569/IJACSA.2021.0120662

[14] T. Tazin, M. Alam, N. N. Dola, M. S. Bari, S. Bourouis, N. M. Khan, Stroke disease detection and prediction using robust learning approaches, Journal of Healthcare Engineering 2021 (2021) 1-12, https://doi.org/10.1155/2021/7633381

[15] S. Rahman, M. Hasan, A. K. Sarkar, Prediction of brain stroke using machine learning algorithms and deep neural network techniques, European Journal of Electrical Engineering and Computer Science 7 (2023) 23-30, http://dx.doi.org/10.24018/ejece.2023.7.1.483

[16] M. Islam, S. Akter, M. Rokunojjaman, J.H. Rony, Stroke prediction analysis using machine learning classifiers and feature technique, International Journal of Electronics and Communication System 1 (2021) 17-22, https://doi.org/10.24042/ijecs.v1i2.10393