A Machine Learning Ensemble Classifier for Prediction of Brain Strokes

Samaa A.Mostafa[1], Doaa S.Elzanfaly[2], Ahmed E.Yakoub[3]
Department of Information Systems-Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt[1, 3]
Department of Information Systems-Faculty of Informatics and Computer Science, The British University in Egypt[2]
Computers and Artificial Intelligence, Helwan University, Cairo, Egypt[2]

***Abstract*—Brain Strokes are considered one of the deadliest brain diseases due to their sudden occurrence, so predicting their occurrence and treating the factors may reduce their risk. This paper aimed to propose a brain stroke prediction model using machine learning classifiers and a stacking ensemble classifier. The smote technique was employed for data balancing, and the standardization technique was for data scaling. The classifiers' best parameters were chosen using the hyperparameter tuning technique. The proposed stacking prediction model was created by combining Random Forest (RF), K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Naive Bayes (NB) as base classifiers, and meta learner was chosen to be Random Forest. The performance of the proposed stacking model has been evaluated using Accuracy, Precision, Recall, and F1 score. In addition, the Matthews Correlation Coefficient (MCC) has been also used for more reliable evaluation when having an unbalanced dataset, which is the case in most medical datasets. The results demonstrate that the proposed stacking model outperforms the standalone classifiers by achieving an accuracy of 97% and an MCC value of 94%.**

***Keywords*—Stroke disease; prediction model; ensemble methods; stacking classifier**

I. Introduction

Stroke is considered one of the riskiest and deadliest diseases affecting humans, as it suddenly strikes the brain. This occurs when the blood flow to the brain is interrupted. Consequently, the brain's ability to receive oxygen and nutrients is compromised, which results in brain cell death within minutes [1]. It is the second leading cause of death globally after ischemic heart disease, as reported by the World Health Organization (WHO) [2]. Ischemic and hemorrhagic strokes are the two primary types. Ischemic stroke happens when a blockage decreases or interrupts blood flow to brain cells, killing the cells within minutes and leading to death. In contrast, Hemorrhagic stroke occurs when weak blood vessels are severely damaged as a result of hypertension, high cholesterol, and other risk factors [3]. Strokes are caused by a variety of risk factors, including medical factors such as high blood pressure, heart disease, diabetes, high cholesterol, and atrial fibrillation, as well as bad habit factors such as smoking, obesity, unhealthy foods, and lack of physical inactivity. The word "FAST" can be used to recognize the main stroke symptoms [4]. The abbreviated FAST stands for four words. F stands for facial laughter perception that they cannot smile or that their mouth or eyes have closed. A refers to the individual with a stroke who might not be capable of raising both arms and maintaining them up. S stands for speech that the person is unable to speak or hard to understand. T is the time at which the patient needs to visit the hospital right away.

The early prediction of brain stroke occurrence to deal with their risk factors is considered a lifesaving matter. Machine learning and AI techniques can be used to determine the likelihood of a stroke occurring in light of their significant advances in predicting different diseases. Different classification algorithms have been used for predicting strokes with reasonable results [5] [6] [7]. The ensemble method is widely used in medical applications due to its accuracy in predicting different diseases [8] [9] [10]. These methods integrate the prediction outcomes of various classification models to enhance the overall performance. They fall into three main categories: Bagging, Boosting, and Stacking. In bagging, several base classification models are sequentially trained, and then use the majority voting to integrate the prediction outcomes [11]. Whereas, in boosting, several base models are trained sequentially to correct the previous models' errors sequentially [12]. In stacking, the classification task is completed in two stages: the first involves training multiple base models on the entire dataset, and the second involves using a meta-learner classifier to train on the first layer's prediction results to provide the final prediction [13]. The base models in bagging and stacking must be homogenous, but in stacking it could be heterogeneous. Rather than relying solely on the output of a single model, these techniques guarantee the delivery of more accurate and trustworthy results from multiple models [14] [15]. Few studies have used ensemble methods for developing brain stroke prediction models, despite the value of using a stacking ensemble classifier to build predictive models with trustworthy outcomes in a variety of fields, including medical and natural phenomena [16] [17] [18].

The main contribution of this paper is to propose a stroke prediction model using a stacking classifier with multiple base model classifiers to enhance the prediction process. Level one classifiers in the stacking model are the Random Forest, K-Nearest Neighbors, Logistic Regression, Support Vector Machine, and Naive Bayes. In level two, the random forest classifier serves as a meta-learner, combining the prediction results of level one to provide the final prediction. To fulfill this aim, the following tasks have been completed:

1) As with most medical datasets, the used dataset is unbalanced. To balance the dataset, this study conducted the SMOTE technique [19].

2) The standard scalar technique has been used to put the data values on the same scale.

3) The model has been constructed using cross-validation with cv = 10.

4) Hyperparameter tuning was employed on the base classifiers to pick the best parameters for each classifier.

5) In addition to the classical evaluation metrics, the MCC (Matthews Correlation Coefficient) value [20] has been used to evaluate the proposed model as it is more realistic for unbalanced datasets.

This paper is structured as follows: Section II provides the literature reviews that have been done on using machine learning classifiers to generate a stroke prediction model. Section III describes the methodology and proposed model used to construct the prediction process. Section IV offers insights into the assessment of the study's findings. Section V displays the discussion of the study. Finally, Section VI reports the conclusion of the work and aspects of future works.

II. Literature Survey

A. Building Stroke Prediction Model by using Classification Algorithms

Most works in the area of brain stroke prediction, using machine learning techniques, are building their models based on standalone classifiers.

Singh and Choudhary [21] built a neural network model for stroke prediction. They used a dataset from the cardiovascular health study (CHS). They applied the Principal Component Analysis (PCA) to minimize the dimensionality of the features and then the decision tree algorithm to choose the most relevant features. The number of instances they have used to build the predictive model is small enough to ensure the accuracy of the results.

Nwosu et al. [22] presented a prediction model for brain strokes by using various machine-learning classifiers. The prediction model was built by using three classifiers Neural network (multi-layer perception), Decision tree, and Random Forest. They achieved 75% accuracy when using the neural network classifier. The main purpose of any medical prediction model is to increase the model's accuracy, but in this study, their results are not sufficient to be trustworthy.

Almadani and Alshammari [23] proposed a stroke prediction model using J48, Jrip, and neural networks (multilayer recognition). The model was built using datasets from the data management department of King Abdulaziz Medical City, Saudi Arabia. Comparing the accuracy of the algorithms, they found that the J84 algorithm achieved a higher accuracy prediction of about 95.25% using principal component analysis (PCA).

Jeena and Kumar [24] developed a stroke prediction model that predicts the probability of developing a stroke based on various risk factors. Model age, atrial fibrillation, gait symptoms, visual impairment, etc. Predictive models were created using support vector machine classification with various kernel functions such as linear, quadratic, RBF, and polynomial. The most accurate function was the linear kernel function, which achieved 91% accuracy. The main drawback here is that the database size is not large enough to make the prediction results more reliable and consists of 350 cases. However, this study did not consider the unbalancing in the stroke dataset, resulting in inaccurate results.

Mahesh and Srikanth [25] wanted to develop a stroke prediction model using decision trees, naive Bayes, and artificial neural network classification algorithms for machine learning. Their study highlights the impact of modifiable and non-modifiable risk factors for stroke. The data set has some risk factors, such as high blood pressure, smoking, and other factors. They utilize the AUC (area under the curve) and ROC (receiver operating characteristics) to measure the total performance of predictive models. The higher the AUC result, the better the prognosis. Their results show that the three algorithms provide acceptable accuracy in the prediction process. A web application was used as the user interface to provide stroke risk alerts. The AUC_ROC score alone cannot be considered a measure of a predictive model.

Sailasya and Kumari [26] trained their prediction model for stroke using Logistic regression, Decision tree, Random Forest, K-nearest neighbors, Support vector machine, and Nave Bayes, six machine learning classifiers. They used a dataset containing risk factors for strokes. They also developed an HTML page as a user interface to get the values of stroke parameters from the user and provide him with the prediction result. They evaluated the overall performance by using the F1 score, accuracy, precision, and recall. The outcome demonstrates that the Nave Bayes classifier has achieved the highest accuracy of 82% compared with the other used classifiers. The achieved accuracy is not accurate enough to predict such a critical medical condition.

Sudha et al. [27] used three machine learning algorithms: decision trees, naive Bayes, and neural networks to build a stroke prediction model. They used a series of data consisting of the patient's troops. The dimensional reduction process is done using a PCA. The decision tree algorithm achieved the highest accuracy of 94%.

Tazin et al. [28] implemented a stroke prediction model using the machine learning techniques of the decision tree, random forest, logistic regression, and voting classifier. The prediction model was built using a stroke dataset that included risk factors. They evaluated the classifiers by using the confusion matrix. The random forest classifier has the highest accuracy of 96% among all classifiers.

Cheon et al. [29] performed a study to decide the ability to predict patients with strokes and the ability of death. They constructed their prediction model using data from the Korean Centers for Disease Control and Prevention (KCDC). Deep neural networks were utilized in the model's construction. They reduced the dataset dimension by using PCA (principal component analysis). They evaluated their model by the confusion matrix. Their area under the curve (AUC) was at its highest, at about 83.5%.

1) Monteiro et al. [30] built a model to predict stroke functional diagnosis. A dataset consisting of 541 patients was used. Popular algorithms such as logistic regression, decision

tree, support vector machine, random forest, and XGBoost were used to construct the prediction model. The area under the curve (AUC), which is greater than 90%, was used to assess the final performance of the models.

Amini et al. [31] conducted decision tree and k-nearest neighbors' classifiers in their stroke prediction model. A stroke dataset with various risk factors was used in their study. The evaluation step reveals that the decision tree algorithm outperformed the KNN algorithm in terms of accuracy, achieving a score of 95.42%.

Ali et al. [32] extended their prediction model of strokes by using distributed machine learning algorithms with the aid of a popular platform in big data called Apache Spark. The prediction model was built using a Decision Tree, Support Vector Machine, Random Forest, and Logistic Regression classifiers. A healthcare stroke dataset was used in their work. They evaluated the model's performance using accuracy, precision, recall, and the f1-measure. Out of all classifiers, Random Forest has achieved the highest accuracy of 90%.

Islam et al. [33] executed a cloud-based mobile application that helps provide the user with a warning about the probability of having a stroke. Building a prediction model using classifiers from machine learning, such as Logistic Regression, Decision Tree, K-Nearest Neighbors, and Random Forest, is the fundamental principle on which the web application is based. A dataset consisting of stroke risk factors was used. They evaluated their classifiers by using accuracy, precision, and f1-score. The highest accuracy of 96% was attained by random forest across all performance metrics.

Akter et al. [34] proposed a stroke prediction model with acceptable accuracy. Popular machine learning algorithms like Random Forest, Support Vector Machine, and Decision Tree were developed using their model. The confusion matrix was used to evaluate their prediction model, and the results show that the Random Forest classifier had the highest accuracy (95.30%).

My main criticism of the above works is that the evaluation process is based on traditional measures that do not consider that the datasets are unbalanced by nature (like most medical datasets). Furthermore, few research works have been proposed in the area of stroke prediction using ensemble classifiers.

B. Stroke Prediction Model by using Ensemble Classifiers

Govindarajan et al. [35] used homogenous ensemble classifiers and conventional machine learning algorithms to create their prediction model, including artificial neural networks, Support Vector Machines, boosting, bagging, and random forests. Their work has been done on 507 stroke patients. They evaluated their work using accuracy, precision, recall, sensitivity, specificity, and standard deviation. The neural network classifier has achieved the highest accuracy of 95.3.

Rado et al. [36] built an ensemble model using the homogeneous ensemble method Random Forest (Bagging), Adaptive Boosting, and the heterogeneous ensemble method Stacking and compared their results. They evaluated the model's performance by using accuracy, Mean Squared Error (MSE), precision, and F-measure and compared it with standalone classifiers. Their results show that the ensemble classifiers have attained better accuracy than standalone classifiers. With an accuracy of 87.58%, the stacking classifier is the most accurate.

III. PROPOSED MODEL

The proposed ensemble model for predicting whether an individual will have a brain stroke or not is based on many risk factors such as age, gender, heart disease, marital condition, and other factors. The model uses multiple algorithms as shown in Fig. 1. The dataset called Stroke Prediction Dataset has been used [37]. The dataset was firstly loaded, and then data preprocessing techniques were applied such as Simple Imputer for handling null values, Label Encoder for converting categorical values into numerical values, standardization technique for making data on the same scale, and SMOTE technique for making data more effective to build the model. Following that, machine learning algorithms such as K nearest neighbors, Gaussian naive Bayes, Logistic Regression, Support Vector Machine, and Random Forest were used after tuning them by using the hyperparameter tuning concept to find the best hyperparameters for each algorithm. These algorithms are the base learners in level one for building the stacking model. The Random Forest was then employed as a meta-learner in level two of the stacking model, which generate the final prediction by using the predictions from the base learners in level one as input. Finally, the proposed prediction model was evaluated by measuring the accuracy, precision, recall, F1 score, and MCC (Matthews Correlation Coefficient) for realistic evaluation.

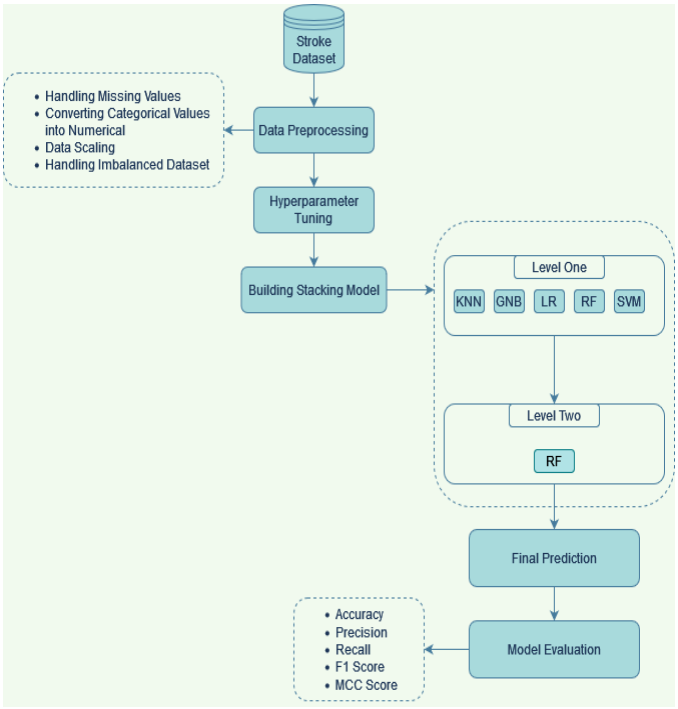

Fig. 1. Stroke prediction model.

A. Stroke Dataset

The prediction model was built by using a stroke prediction dataset from Kaggle that has been presented in Fig. 2. This dataset consists of 5110 rows and 12 columns. The features include ID, gender, age, hypertension, heart disease, ever married, work type, residence type, avg. glucose level, BMI, and smoking status as shown in Table I. The target column is stroke. The identifier column was deleted during the experiment because it does not give any information, it is just the number of the patient.

B. Data Preprocessing

1) Handling missing values: There is an important step before building the predictive model, data preprocessing should be done in which any noise removed, duplication, or incomplete information and handle any missing data. These issues may lead the model to produce incorrect results or affect the overall model quality. In this stroke dataset, there are no duplicated rows. But there are 201 missing values in the BMI column as shown in Fig. 3. Those missing values were filled by using the data column's mean.

2) Converting categorical values into numerical values: The next step was to convert categorical values into numerical ones. The dataset consists of five features with type strings; namely gender, ever married, work type, residence type, and smoking status. The label encoding technique has been employed to convert those features into numerical values.

3) Data scaling: After that, the Standardization technique was used to make the data values in the same range because the input data values fall in different scales. The Standard Scalar function was applied which makes the data values between zero and one, and it is also working with the standard deviation and the data point mean.

4) Handling imbalanced dataset: Class imbalance of datasets is a communal problem in machine learning. Imbalance data can affect the accuracy of machine learning models negatively. This problem occurs when the target class has observations not equal in distribution. That is, there is a high number of instances for a one-class label but an exceptionally small number of observations for the other class exists. In the dataset, the target class of stroke is imbalanced because class "0" which is the number of occurrences of patients who do not have a stroke exceeds class "1" which is the number of patients who have a stroke. As shown in Fig. 4, the total number of instances for classes "0" and "1" is 4861 (about 4.9%) and 249 (about 95.1%), respectively.

To manage this issue, Synthetic Minority Oversampling Technique (SMOTE) which is an oversampling technique is applied. Oversampling requires increasing the number of instances in the minority class by duplicating the records of the minority class to make the instances in the minority class equal to the instances in the minority class. SMOTE technique is a modified version of oversampling in which it is not just duplicating the records in the minority class because it will not add any latest information, it uses the concept of k nearest neighbor to randomly select the neighbor instances and create a synthetic instance. It easily works by selecting examples that are close to the feature space, drawing a line between the examples in the feature space, and drawing a new sample at a point along that line. After applying it, the dataset became balanced with the number of instances of 0 equals 1 in the target class as shown in Fig. 5.

TABLE I. DATASET ATTRIBUTES AND THEIR DESCRIPTION

Attribute Number	Attribute Name	Description
1	id	A unique identifier number for the patient
2	gender	Refers to the gender of the patient
3	age	The age of the patient
4	hypertension	refers to if the patient suffering from hypertension or not
5	heart_disease	refers to whether the patient is suffering from any heart disease or not
6	ever_married	refers to if the patient is married or not
7	work_type	refers to the different types of work
8	Residence_type	refers to the type of the patient's residence
9	avg_glucose_level	refers to the level of blood sugar
10	bmi	refers to the body mass index of the patient
11	smoking_status	refers to the patient's smoking status
12	stroke	refers to if the patient had a stroke or not

id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	1
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	24	never smoked	1
56669	Male	81	0	0	Yes	Private	Urban	186.21	29	formerly smoked	1
53882	Male	74	1	1	Yes	Private	Rural	70.09	27.4	never smoked	1
10434	Female	69	0	0	No	Private	Urban	94.39	22.8	never smoked	1
–	–	–	–	–	–	–	–	–	–	–	–
64908	Male	79	0	1	Yes	Private	Urban	57.08	22	formerly smoked	0
63884	Female	37	0	0	Yes	Private	Rural	162.96	39.4	never smoked	0
37893	Female	37	0	0	Yes	Private	Rural	73.5	26.1	formerly smoked	0
67855	Female	40	0	0	Yes	Private	Rural	95.04	42.4	never smoked	0
25774	Male	35	0	0	No	Private	Rural	85.37	33	never smoked	0
19584	Female	20	0	0	No	Private	Urban	84.62	19.7	smokes	0

Fig. 2. Stroke prediction dataset.

gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
bmi	201
smoking_status	0
stroke	0

Fig. 3. Total number of missing values in each column.

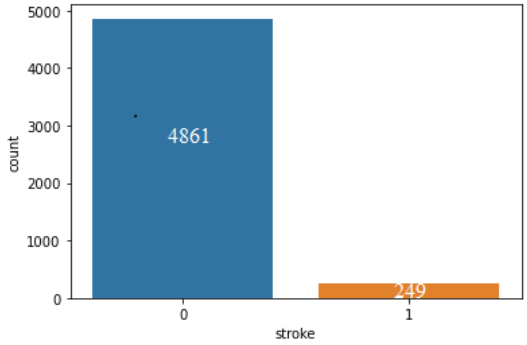

Fig. 4. Stroke proportion before SMOTE.

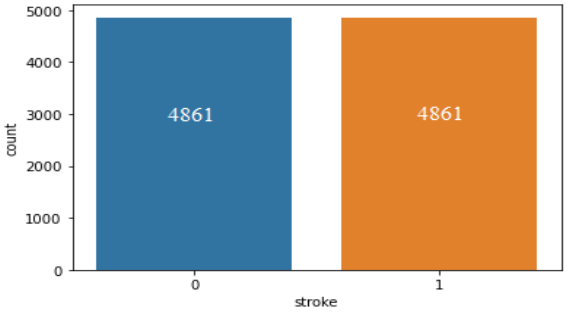

Fig. 5. Stroke proportion after SMOTE.

C. Hyperparameter Tuning

Hyperparameter tuning is the process of finding the optimal hyperparameters for the classifier. It tests several combinations of the parameter values and finds the optimal values that maximize the accuracy of the prediction model. In that work, the grid search technique was used to find the best parameters for using classifiers as presented in Table II.

D. Building Stacking Model

1) Base classifiers for the stacking model: After the step of data preprocessing and hyperparameter tuning, the stacking model was started to build for stroke prediction. The first step is to train multiple heterogeneous algorithms on the dataset, this step is called stacking level one. The second step is to build a meta-model that helps in combining the base learners' predictions with the final prediction. Five popular machine learning classification algorithms were trained on the dataset at level one, which was as follows:

- K-nearest neighbors
- Random Forest
- Gaussian Naïve Bayes
- Logistic Regression
- Support Vector Machine

a) K nearest neighbors: K-nearest neighbors are the most commonly used algorithm for both classification and regression problems in machine learning. It is also known as KNN or K-NN. The basic idea of KNN is to group data points falling in near to each other in the same class. It classifies the new data point based on a similarity measure. It uses Euclidean distance to determine the nearest neighbor class to a data point that needs to be classified.

b) Random Forest: Random Forest is the most commonly used algorithm for classification and regression problems in machine learning based on Bagging ensemble learning. In ensemble learning, it combines the results of the prediction of multiple base classifiers to build one robustness model with higher performance. A random forest consists of several decision trees that were trained individually on a random data sample and subset features. Decision trees produce several results, and these results are combined using the majority voting in the case of the classification problem and the mean average in the case of the regression problem to produce the final result. The greater number of trees in the random forest leads to the highest prediction performance. This algorithm prevents the issue of overfitting and enhances the model's accuracy.

c) Gaussian Naïve Bayes: This classification algorithm employs the Bayes theorem. It assumes that all the input features or attributes are independent of each other. Bayes' theorem finds the probability of an occurrence of an event given the probability of another event that has already occurred before.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

TABLE II. HYPERPARAMETER TUNING FOR BASE CLASSIFIERS

	Best Parameter values with grid search cv	Best score by selecting the best parameters
K-Nearest neighbors	neighbors=1 weights='uniform'	0.947
Random Forest	n_estimators= 2000 max_depth= 50 criterion=entropy bootstrap= False random_state=1	0.966
Gaussian Naïve Bayes	var_smoothing= 0.0533	0.776
Logistic Regression	C=0.00294 max_iter=100 penalty=L1 solver=saga	0.792
Support Vector Machine	C=1000 Kernel=rbf Degree=1	0.928

d) Logistic Regression: Logistic regression is an effective algorithm for binary classification problems. It uses some independent features to predict a categorical or discrete dependent variable, such as 0 or 1, male or female, yes or no, and so on. It uses the Sigmoid function that gives the probability values between 0 and 1 instead of giving the output values 0 and 1 by mapping the predicted values to probabilities.

e) Support Vector Machine: The main concept of SVM is to create the best fit line or decision boundary that can split the classes, so it can easily classify the new data point in the correct class in the future. The best fit line or decision boundary is called a hyperplane.

2) Stacking meta learner: Stacking is one of the most efficient machine-learning techniques. It is a widely used ensemble technique because it improves the model performance and solves complex problems. It is used to combine the predictions from multiple models by using a meta-model. In stacking, the dataset is divided into two sets, the first one is the training set and the second one is the test set. This training set is divided into a training set that is trained by heterogeneous base learners to create the first-level models and a validation set that is used by the models to make the predictions of level one, which are used as new features for the second-level meta-learner. This meta-learner is trained on this new training data, which consists of the first-level predictions, and uses the test set to make the final prediction. The major point is to construct a meta-model that is trained with the first-level outcomes. This step helps in providing an accurate final prediction. Fig. 6 shows the sequence of the stacking model. The base model classifiers were K-nearest neighbor, Random Forest, Gaussian Naive Bayes, Logistic Regression, and Support Vector Machine. In the second level, the Random Forest has been chosen to be the meta learner which has been trained on the outputs of the base learners, as it is the most suitable algorithm for providing an accurate result. It has the best accuracy of 96% compared to other stand-alone classifiers.

3) K-Fold Cross-Validation: While building the model, the k-fold cross-validation is conducted to divide the dataset into K collections with equal sizes, Where K represents an integer number. Those collections are called folds. Making iterations equal to the number of folds. At each iteration:

- Take k-1 folds for training and k-fold for validating.
- Changing the folds of training and validating.
- Calculate the accuracy of each iteration.
- Take the average of all accuracies.

It usually gives more accurate results for the model as it trains the model multiple times by changing the training and testing data slot at each iteration. Fig. 7 is an example of cross-validation with five folds.

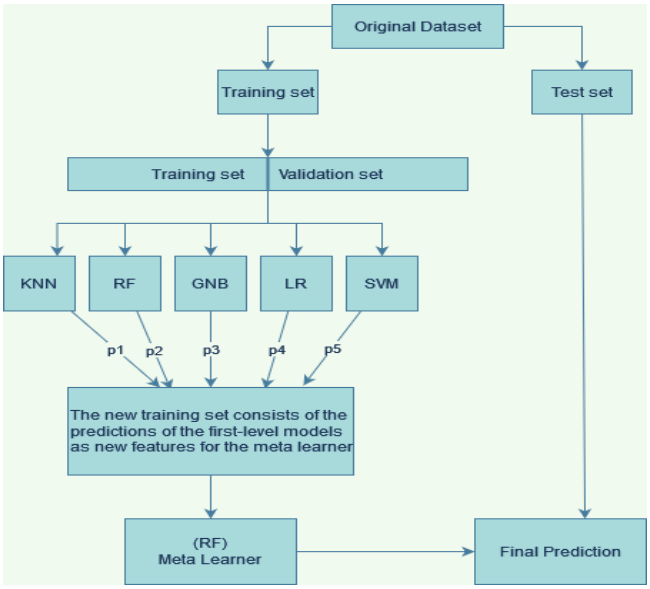

Fig. 6. Sequence of stacking model.

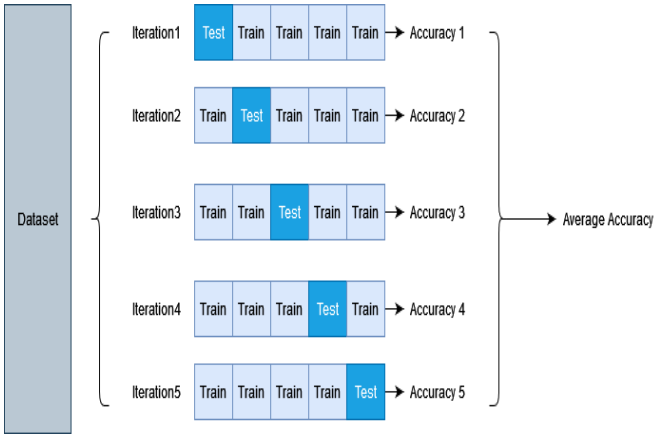

Fig. 7. Example of cross-validation with k=5.

E. Final Prediction

The Random Forest classifier has been used in the second level of the stacking model, which is a meta-learner that combines the results of the first level. This learner trained on those results and provided the final prediction result.

F. Model Evaluation

The prediction model was evaluated by using various evaluation metrics such as accuracy, precision, recall, f1-score, and MCC (Matthews Correlation Coefficient). This paper considered the MCC value for the classifiers because it is an effective measure for binary classification and unbalanced datasets, as in the used stroke dataset. It calculates the correlation between the actual and the predicted values. If that correlation value is higher, that means that the prediction is better. It considered all the confusion matrix values. When the value of MCC is close to one then it means that the model well predicted the actual and the predicted values.

Accuracy = TP/TP+TN+FB+FN

Precision = TP/TP+FP

Recall = TP/TP+FN

F1-score = 2(recall*precision)/(recall+precision)

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

True Positive (TP) Predicted Positive and they are Positive

True Negative (TN) Predicted Negative and they are Negative

False Positive (FP) Predicted Positive but they are Negative

False Negative (FN) Predicted Negative but they are Positive

Positive here means that patient has a stroke (1) and negative means that the patient does not have a stroke (0).

IV. Results

After building the prediction model, the classification algorithms were compared with the accuracy, precision, recall, f1 score, and MCC measures as shown in Table III and Fig. 8:

- All features were used.
- SMOTE was applied to balance the data.
- Standard scalar was applied to make the data values on the same scale.
- Cross-validation was used to build the prediction model with cv=10.
- Hyperparameter Tuning was applied to the base classifiers to choose the optimal parameters for each classifier as shown in Table II.
- Hyperparameter Tuning is the process of choosing the best parameters for the classifier to increase the classifier's performance.
- Built the stacking model with KNN, RF, NB, LR, and SVC in level one and RF as meta learner in level two.
- Model evaluation was done by using accuracy, precision, recall, f1 score, and MCC measures.

TABLE III. Comparison between the Algorithms with Hyperparameter Tuning

	Accuracy	Precision	Recall	F1 Score	MCC
KNN	0.95	0.91	0.99	0.95	0.89
Random Forest	0.96	0.99	0.97	0.96	0.92
Naïve Bayes	0.78	0.79	0.83	0.78	0.55
Logistic Regression	0.79	0.79	0.85	0.80	0.58
SVC	0.93	0.94	0.98	0.94	0.87
Stacking	0.97	0.99	0.97	0.97	0.94

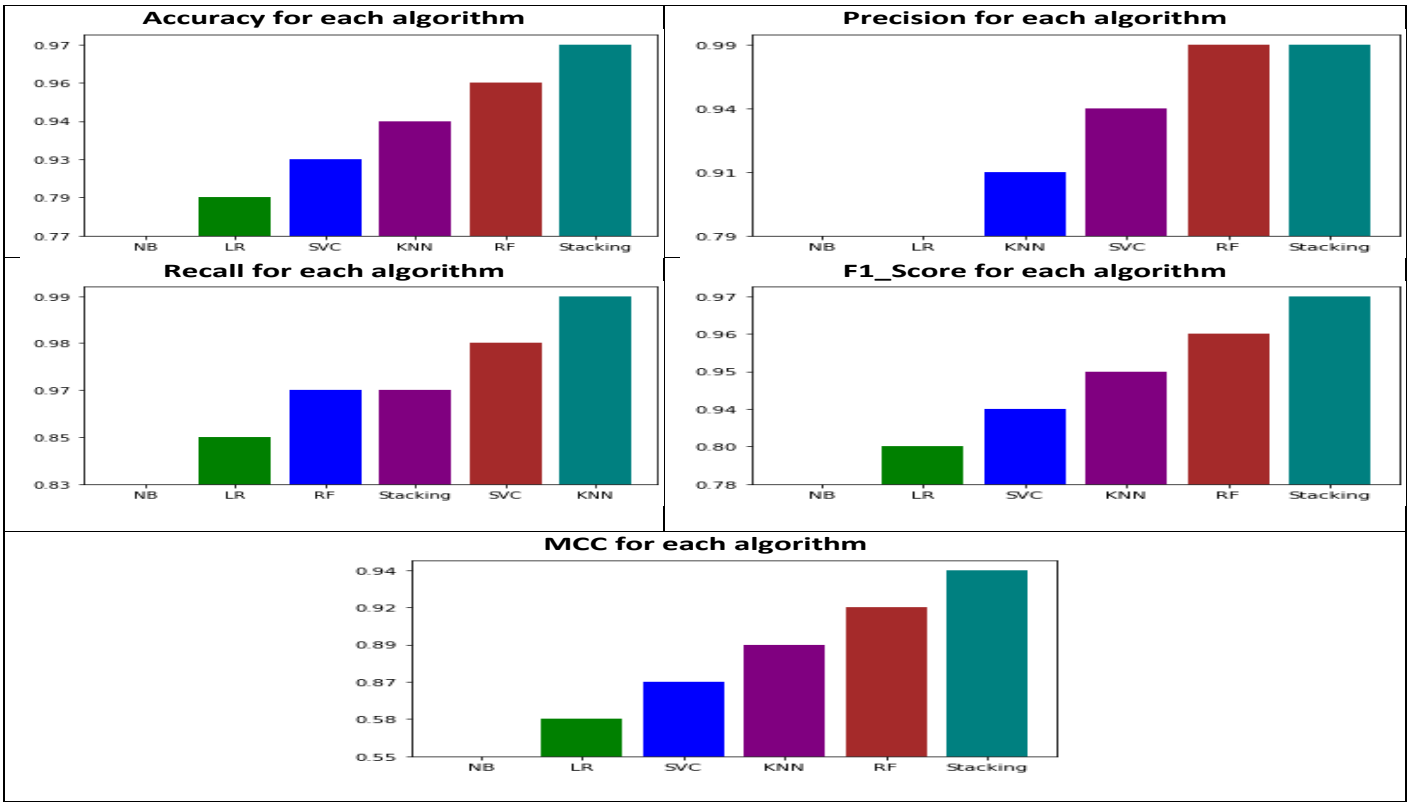

Fig. 8. Comparison between the base classifiers and the stacking model.

From III, the stacking algorithm has achieved the highest accuracy compared with the other standalone classifiers as it achieved the highest accuracy about 97%, and this result shows the efficiency of the ensemble methods. It also achieved the highest MCC value of 94% which means that the stacking model provides an accurate prediction as it well predicts the actual and the predicted values.

Table IV shows the difference between using SMOTE technique to balance the data and without it in comparison with the model accuracy.

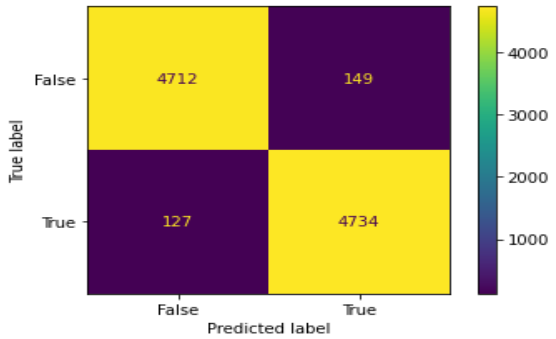

Fig. 9. Stacking confusion matrix.

TABLE IV. ALGORITHMS ACCURACY BETWEEN USING SMOTE AND WITHOUT USING SMOTE

	Accuracy without SMOTE	Accuracy with SMOTE
K-NN	0.91	0.95
Random Forest	0.95	0.96
Gaussian Naïve Bayes	0.87	0.78
Logistic Regression	0.95	0.79
Support Vector Machine	0.95	0.94
Stacking	0.95	0.97

V. DISCUSSION

Over an effort to enhance a brain stroke prediction model a combination of classification algorithms and the stacking ensemble classifier was used to create the prediction model. The outcomes of this research supported the advantages of employing the ensemble method when establishing predictive models.

The prediction model procedures have included the following steps: data preprocessing, in which dataset issues are addressed using various techniques such as column's average for filling in missing values, Label Encoder to convert categorical features into numerical features, Data Scaling to align the data values, and Smote technique to balance this medical dataset. Then, to improve the accuracy of each classifier, hyperparameter tuning was used to determine the most appropriate hyperparameter. The stacking model's level one consists of the Random Forest, K-Nearest Neighbors, Logistic Regression, Support Vector Machine, and Naive Bayes while level two is mainly composed of the Random Forest classifier, which serves as a meta-learner by combining the level one prediction results to generate the final prediction.

Comparing the proposed stacking prediction model with other standalone classifiers, it demonstrated higher classification performance as shown in Table III and Fig. 9. It obtained the best accuracy result of approximately 97%, demonstrating the efficiency of the ensemble methods. As well, it achieved the maximum MCC value of 94%.

VI. CONCLUSION AND FUTURE WORK

This paper intended to demonstrate a stacking ensemble classifier-based prediction of brain stroke disease. Traditional classifiers served as the stacking model's base models. To enhance the final prediction result, the output of those classifiers was combined using the stacking meta-learner.

According to the experimental findings, using a stacking ensemble classifier can significantly increase prediction accuracy as it achieved about 97% and give the highest MCC measure of about 94%, which ensures that the prediction is correct. It also demonstrated that the MCC value is more trustworthy than the conventional measures in the evaluation of the two-class confusion matrix. Using some risk factors, this model aids in accurately predicting whether someone will suffer a brain stroke or not. The future scope of this study will include using other combinations of the base model classifiers in the stacking model. As well, it may extend to utilizing other effective attributes for building the prediction model or employing deep learning algorithms.

REFERENCES

[1] "About Stroke", www.stroke.org, 2022. [Online]. Available: https://www.stroke.org/en/about-stroke.

[2] "The top 10 causes of death", Who. int, 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[3] CDC, "About stroke," Centers for Disease Control and Prevention, 06-May-2022. [Online]. Available: https://www.cdc.gov/stroke/about.htm.

[4] "Stroke",nhs.uk,2022.[Online].Available:https://www.nhs.uk/conditions/stroke/.

[5] A. Roy, A. Kumar, K. Singh, and D. Shashank, "Stroke Prediction using Decision Trees in Artificial Intelligence. Stroke Prediction Using Decision Trees in Artificial Intelligence," IJARIIT, vol. 4, pp. 1636–1642, 2018.

[6] B. Khalid and N. Abdelwahab, "A model for predicting Ischemic stroke using Data Mining algorithms," *IJISET*, vol. 2, no. 11, 2015.

[7] M. Rajora, M. Rathod, and N. S. Naik, "Stroke prediction using machine learning in a distributed environment," in Distributed Computing and Internet Technology, Cham: Springer International Publishing, 2021, pp. 238–252.

[8] I. D. Mienye, Y. Sun, and Z. Wang, "An improved ensemble learning approach for the prediction of heart disease risk," Inform. Med. Unlocked, vol. 20, no. 100402, p. 100402, 2020.

[9] K. Shilpa and T. Adilakshmi, "Applying ensemble techniques of machine learning to predict heart disease," in Proceedings of the International Conference on Cognitive and Intelligent Computing, Singapore: Springer Nature Singapore, 2022, pp. 775–783.

[10] Z. Asghari Varzaneh, M. Shanbehzadeh, and H. Kazemi-Arpanahi, "Prediction of successful aging using ensemble machine learning algorithms," BMC Med. Inform. Decis. Mak., vol. 22, no. 1, p. 258, 2022.

[11] IBM Cloud Education, "What is bagging?" Ibm.com, 11-May-2021. [Online]. Available: https://www.ibm.com/cloud/learn/bagging.

[12] IBM Cloud Education, "What is boosting?" Ibm.com, 26-May-2021. [Online]. Available: https://www.ibm.com/cloud/learn/boosting.

[13] J. Brownlee, "Stacking ensemble machine learning with python," Machinelearningmastery.com, 09-Apr-2020. [Online]. Available:

https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/.

[14] R. Rosly, M. Makhtar, M. Khalid Awang, M. Isa Awang, M. Nordin Abdul Rahman, and H. Mahdin, "Comprehensive study on ensemble classification for medical applications," Int. j. eng. technol., vol. 7, no. 2.14, p. 186, 2018.

[15] S. Džeroski, P. Panov, and B. Ženko, "Machine Learning, Ensemble Methods in," in Encyclopedia of Complexity and Systems Science, New York, NY: Springer New York, 2009, pp. 5317–5325.

[16] B. Pavlyshenko, "Using stacking approaches for machine learning models," in 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 2018.

[17] A. Gupta, V. Jain, and A. Singh, "Stacking ensemble-based intelligent machine learning model for predicting post-COVID-19 complications," New Gener. Comput., pp. 1–21, 2021.

[18] J. Gu, S. Liu, Z. Zhou, S. R. Chalov, and Q. Zhuang, "A stacking ensemble learning model for monthly rainfall prediction in the Taihu Basin, China," Water (Basel), vol. 14, no. 3, p. 492, 2022.

[19] J. Brownlee, "SMOTE for imbalanced classification with python," Machinelearningmastery.com, 16-Jan-2020. [Online]. Available: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/.

[20] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, p. 6, 2020.

[21] Singh, M. Sheetal, and Prakash Choudhary. "Stroke prediction using artificial intelligence." 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON). IEEE, 2017.

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

[23] Almadani, Ohoud, and Riyad Alshammari. "Prediction of Stroke using Data Mining Classification Techniques." International Journal of Advanced Computer Science and Applications (IJACSA) (2018).

[24] R. S. Jeena and S. Kumar, "Stroke prediction using SVM," IEEE Xplore, Dec. 01, 2016. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7988020.

[25] M. Kunder Akash and S. Srikanth, "Prediction of Stroke Using Machine Learning," ResearchGate, 2020.

[26] G. Sailasya and G. L. A. Kumari, "Analyzing the performance of stroke prediction using ML classification algorithms," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, 2021.

[27] A. Sudha, P. Gayathri, and N. Jaisankar, "Effective Analysis and Predictive Model of Stroke Disease using Classification Methods", International Journal of Computer Applications, vol. 43, no. 14, pp. 26-31, 2012. Available: 10.5120/6172-8599.

[28] T. Tazin, M. N. Alam, N. N. Dola, M. S. Bari, S. Bourouis, and M. Monirujjaman Khan, "Stroke disease detection and prediction using robust learning approaches," Journal of Healthcare Engineering, 26-Nov-2021. [Online]. Available: https://www.hindawi.com/journals/jhe/2021/7633381/.

[29] S. Cheon, J. Kim, and J. Lim, "The use of deep learning to predict stroke patient mortality," Int. J. Environ. Res. Public Health, vol. 16, no. 11, p. 1876, 2019.

[30] Monteiro, M., Fonseca, A. C., Freitas, A. T., Pinho e Melo, T., Francisco, A. P., Ferro, J. M. and Oliveira, A. L., "Using Machine Learning to Improve the Prediction of Functional Outcome in Ischemic Stroke Patients", 2018.

[31] L. Amini, R. Azarpazhouh, M. T. Farzadfar, S. A. Mousavi, F. Jazaieri, F. Khorvash, R. Norouzi, and N. Toghianfar, "Prediction and control of stroke by data mining," International Journal of Preventive Medicine, 2013.

[32] Ali, Abdelmgeid A., "Stroke Prediction using Distributed Machine Learning Based on Apache Spark," Stroke 28(15), pp. 89-97, 2019.

[33] M. M. Islam, S. Akter, M. 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. 17-22, Dec. 2021.

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

[35] P. Govindarajan, R. K. Soundarapandian, A. H. Gandomi, R. Patan, P. Jayaraman, and R. Manikandan, "Classification of stroke disease using machine learning algorithms," Neural Comput. Appl., vol. 32, no. 3, pp. 817–828, 2020.

[36] O. Rado, M. Al Fanah, and E. Taktek, "Ensemble of Multiple Classification Algorithms to Predict Stroke Dataset," Advances in Intelligent Systems and Computing, vol. 998, pp. 93–98, 2019, DOI: https://doi.org/10.1007/978-3-030-22868-2_7.

[37] Fedesoriano, "Stroke prediction dataset," Kaggle, 26-Jan-2021. [Online]. Available: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset.