applied sciences

MDPI

Article

A-Tuning Ensemble Machine Learning Technique for Cerebral Stroke Prediction

Meshrif Alruily [1], Sameh Abd El-Ghany [2], Ayman Mohamed Mostafa [2,*], Mohamed Ezz [1] and A. A. Abd El-Aziz [2]

[1] Computer Science Department, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Aljouf, Saudi Arabia; mfalruily@ju.edu.sa (M.A.); maismail@ju.edu.sa (M.E.)
[2] Information Systems Department, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Aljouf, Saudi Arabia; saabdelwahab@ju.edu.sa (S.A.E.-G.); aaeldamarany@ju.edu.sa (A.A.A.E.-A.)
* Correspondence: amhassane@ju.edu.sa

Abstract: A cerebral stroke is a medical problem that occurs when the blood flowing to a section of the brain is suddenly cut off, causing damage to the brain. Brain cells gradually die because of interruptions in blood supply and other nutrients to the brain, resulting in disabilities, depending on the affected region. Early recognition and detection of symptoms can aid in the rapid treatment of strokes and result in better health by reducing the severity of a stroke episode. In this paper, the Random Forest (RF), Extreme Gradient Boosting (XGBoost), and light gradient-boosting machine (LightGBM) were used as machine learning (ML) algorithms for predicting the likelihood of a cerebral stroke by applying an open-access stroke prediction dataset. The stroke prediction dataset was pre-processed by handling missing values using the KNN imputer technique, eliminating outliers, applying the one-hot encoding method, and normalizing the features with different ranges of values. After data splitting, synthetic minority oversampling (SMO) was applied to balance the stroke samples and no-stroke classes. Furthermore, to fine-tune the hyper-parameters of the ML algorithm, we employed a random search technique that could achieve the best parameter values. After applying the tuning process, we stacked the parameters to a tuning ensemble RXLM that was analyzed and compared with traditional classifiers. The performance metrics after tuning the hyper-parameters achieved promising results with all ML algorithms.

Keywords: cerebral stroke; stoke prediction; oversampling; fine-tuning

Citation: Alruily, M.; El-Ghany, S.A.; Mostafa, A.M.; Ezz, M.; El-Aziz, A.A.A. A-Tuning Ensemble Machine Learning Technique for Cerebral Stroke Prediction. *Appl. Sci.* **2023**, *13*, 5047. https://doi.org/10.3390/app13085047

Academic Editors: István Vassányi, István Kósa and László Balkányi

Received: 30 March 2023
Revised: 15 April 2023
Accepted: 16 April 2023
Published: 18 April 2023

1. Introduction

The central nervous system, in particular the cerebrum, plays a critical role in regulating numerous physiological functions such as memory, movement, cognition, and speech, as well as autonomic control of the operation of vital organs. However, severe disruptions or pathologies affecting the brain can have fatal consequences. A stroke is an acute medical condition that requires prompt attention [1,2]. According to the Global Stroke Organization [3], 5.5 M will die from strokes, and 13 M will suffer from a stroke annually. Stroke is the leading cause of death and functional impairment globally, so it significantly affects all facets of life [4]. Anyone at any age, regardless of gender or physical condition, can be affected by stroke [5]. There are two types of cerebral strokes: ischemic and hemorrhagic, where the damage can be temporary or extreme, ranging from mild to severe. Hemorrhages are rare and entail a brain hemorrhage occurring by the blood vessel rupture. The most frequent kind of stroke, an ischemic stroke, occurs when an artery becomes narrowed or blocked, preventing blood from reaching a specific brain part. Cerebral strokes are the sixth most common cause of mortality in the United States and the fourth most common cause of death in India. The 795,000 cases/year frequently result in permanent disability in the United States [6,7]. The India Collaborative Acute Stroke

Study (ICASS) statistical report indicates that more than two thousand persons in India had strokes in 2004 [8]. According to statistics from the Stroke Association, five out of every 100,000 children in the United Kingdom experienced a stroke in 2012 [9].

On the other hand, the overall stroke mortality rate in Canada was more than 15 thousand in 2000 [10]. Myocardial infarction increases the risk of stroke, such as heart conditions that increase the probability of stroke cases [11–13]. A healthy and balanced lifestyle can help avoid stroke by keeping an appropriate weight index, an optimum glucose level, and good heart and kidney function. Additional good practices include abstaining from alcohol and tobacco, maintaining a healthy weight, and exercising regularly. Prompt detection and proper treatment are necessary to prevent brain injury and associated consequences in other body areas. Magnetic resonance imaging (MRI) and computed tomography (CT) investigations are frequently used in early stroke identification.

The paper contains five sections, including the introduction section. Section 2 summarizes the recent studies of cerebral stroke with their results. The proposed methodology with machine learning algorithms is explained in Section 3. Section 4 provides the implementation of the methodology and the tuning ensemble for cerebral stroke prediction with the experimental results. Finally, the paper concludes with an explanation of the results in Section 5.

2. Literature Review

As presented in [14], different machine learning techniques can predict a patient with a high stroke risk. The research will use three models of machine learning algorithms: RF, DT, and NB. After evaluating each approach, the prediction uses the patient's health history as the attribute in each mode. The RF method has the highest accuracy, at 94.781%, followed by the DT method, which recorded an accuracy of 91.906%, and the NB method recorded the lowest accuracy, at 89.976%. The RF method outperforms the other methods in terms of accuracy based on these findings. Using a variety of elements that record the participant profiles, the authors of [15] investigated the efficacy of various ML algorithms, RF, LR, K-NNs, SGD, DT, stacking, and majority voting, to identify the accurate algorithm to predict the cases of stroke. Based on the results that exceed 98%, stacking classification obtains high results. As a result, the stacking technique effectively identifies those who are eventually likely to suffer a stroke. The authors of [16] trained four distinct models for accurate stroke prediction with machine learning algorithms and numerous physiological parameters. With an accuracy of approximately 96%, the RF performed the best for this task. As proposed in [17], the authors used two datasets. The first dataset is a list of medical checks, changes in the environment, and changes in the body. The dataset was balanced using the SMOTE method, missing values were replaced with KNN Imputer values, and outliers were removed. According to the corresponding features values, diabetes and obesity were created as two features. The five ML algorithms that ultimately fed these features were SVM, KNN, DT, RF, and MLP. The MRI scans comprise the second dataset, where the average filter has been used to optimize each image and eliminate noise.

The dataset has been balanced using a data augmentation technique to prevent overfitting of data. Relatively 80% of the dataset was used for training, validation, and 20% for testing. The AlexNet model and SVM algorithm were used to extract the deep features. The deep learning model (AlexNet) performed worse than the hybrid algorithm that combined machine learning and deep learning. AlexNet and the SVM achieve 99.9% accuracy, 100% sensitivity, 99.80% specificity, and 99.86% AUC, respectively. The authors of [18] demonstrate how several machine-learning systems may correctly predict stroke cases based on physiological parameters. The NB Classification exceeds the alternative algorithms with an accuracy of 82%.

As explained in [19], the authors proposed a comprehensive assessment of patient characteristics in the digital health record. They used systematic analysis to look at various features. They carried out feature correlation and stepwise analysis to select the best features. In addition, they performed a principal component analysis (PCA), and the

results explained that the principal components are required to account for more variance. Finally, various features and principal component configurations were used to test three machine-learning algorithms, neural network (NN), DT, and RF. They discovered that a feature combination of A, HD, HT, and AG works best for a NN, with an accuracy and miss rate of 78% and 19%, respectively. As presented in [20], the authors used a stroke prediction algorithm and an improvised RF ensemble technique to find the risk factor. The prediction model has a 0.03% of error rate and an accuracy of 96.97%. As presented in [21], the authors proposed a hybrid framework to predict cerebral stroke disease using classification and clustering. The authors implemented clustering using an enhanced hierarchal clustering technique and then implemented LR, RF, SVM, NN, and XGBoost classifiers. According to accuracy and AUC, all the classifiers produced satisfactory results, and the RF classifier achieved 97%, which was the best result.

We can conclude from the above literature review of the most recent research that none accurately classified cerebral stroke. However, our proposed tuning ensemble RXLM achieved 95.29%, 99.13%, 96.38%, 94.36%, 95.35%, 90.59%, and 90.63%, respectively.

3. Materials and Methods

We proposed a tuning ensemble RXLM composed of RF, XGBoost, and LightGBM to predict cerebral stroke diseases. The designed methodology and associated algorithms, which comprise the aim of this study, were achieved successfully. The dataset used in the proposed classification algorithms and methods is described in this section.

3.1. Stroke Prediction Dataset

The dataset was sourced from the famous benchmark dataset repository Kaggle [22]. We concentrated on participants over the age of 18 from this dataset. The dataset has 11 features and 5110 rows. Table 1 shows the descriptions for each feature, where the input to the ML is ten features, and one feature is applied for the target class.

Table 1. Stroke prediction dataset features.

Feature Name	Description	Range
Gender	The gender of the participant is the focus of this feature. There are 1260 men and 1994 women in the population.	Male–Female
Age	Participants over the age of 18 are the subject of this feature.	Float
Hypertension	This characteristic indicates whether this participant has hypertension. 12.54% of participants have high blood pressure.	0 → No Hypertension 1 → Hypertension
Heart Disease	If this individual has heart disease, it is indicated by this feature. The participants' prevalence of heart disease was 6.33%.	0 → No Heart Disease 1 → Heart Disease
Ever Married	The number of participants who are married, which is 79.84%, is represented by this feature.	Yes–No
Work Type	This feature has four categories: private, self-employed, government, and finally, never worked.	Never_worked, Children, Private, Self-employed, or Govt_job
Residence type	This feature has two categories: urban and rural, representing the participant's living situations.	Urban or Rural
Average glucose level (mg/dL)	This feature tracks the participants' average blood glucose level.	Float
BMI (Kg/m^2)	This feature records the participants' BMI.	Float
Smoking Status	There are three categories for this feature: smoker (22.37%), never smoker (52.64%), and formerly smoker (24.99%).	Never smoked, smoked, or Formerly Smoked
Stroke	This attribute identifies if the participant has had a stroke in the past. 5.53% of participants have experienced a stroke.	0 → No Stroke 1 → Stroke

Most characteristics are categorical, except age, average blood glucose level, and BMI, which are considered numerical. The dataset includes 2994 females, 2115 males, and one other, and it includes 249 stroke patients and 4861 normal patients. The dataset was divided into a training set with 4088 rows (3901 no-stroke and 187 strokes) (80%) and a test set with 1022 rows (20%). Class 0 represents the no-stroke class in the dataset, and class 1 represents the stroke class.

3.2. Methodology

In this research, we proposed a robust tuning ensemble RXLM from RF, XGBoost, and LightGBM for predicting cerebral stroke, where an open-access dataset for stroke prediction is applied. The dataset was pre-processed by handling missing values using the K-nearest neighbor (KNN) Imputer technique, eliminating outliers, applying the one-hot encoding method, and normalizing the features with different ranges of values. After splitting the dataset, we applied the SMOTE to the training set only to balance the samples of the two classes. Furthermore, we applied the random search technique to tune the hyper-parameters of RF, XGBoost, and LightGBM to get the best parameter values. After tuning, we stacked the three ML models to propose the tuning RXLM. Traditional classifiers and the proposed tuning ensemble RXLM were assessed and compared. The results of the ensemble's accuracy, AUC, recall, precision, F1-score, Kappa, and MCC recorded 95.29%, 99.13%, 96.38%, 94.36%, 95.35%, 90.59%, and 90.63%, respectively. The evaluation of the proposed ensemble showed that the proposed ensemble achieved superior outcomes. The primary achievements of the research are:

1. We proposed a novel tuning ensemble RXLM from RF, XGBoost, and LightGBM to predict cerebral stroke diseases.
2. We used the open-access Stroke Prediction dataset and the KNN Imputer technique to handle the missing values.
3. We applied the SMOTE on the training set to balance the two classes' samples due to the imbalance of the training subset.
4. We tuned the hyper-parameters of the RF, XGBoost, and LightGBM using the random search technique to get the best parameter values and their high performance.
5. Achieving 95.29%, 99.13%, 96.38%, 94.36%, 95.35%, 90.59%, and 90.63% for accuracy, AUC, recall, precision, F1-score, Kappa, and MCC, respectively, for stroke disease prediction.

As depicted in Figure 1, we trained three ML methods (RF, XGBoost, and LightGBM) and stacked them to propose a tuning ensemble RXLM. Each ML model of RF, XGBoost, and LightGBM has some limitations. However, the proposed tuning ensemble achieved a high binary classification success because it combined the advantages of the three ML models to improve the overall accuracy. Figure 1 depicts the steps of the proposed tuning ensemble model: (1) pre-processing of the Stroke Prediction dataset, (2) splitting the dataset, (3) applying the SMOTE on the training set, (4) tuning the three ML techniques, (5) stacking the three tuning ML models, (6) utilizing the measured metrics to evaluate the proposed tuning ensemble model.

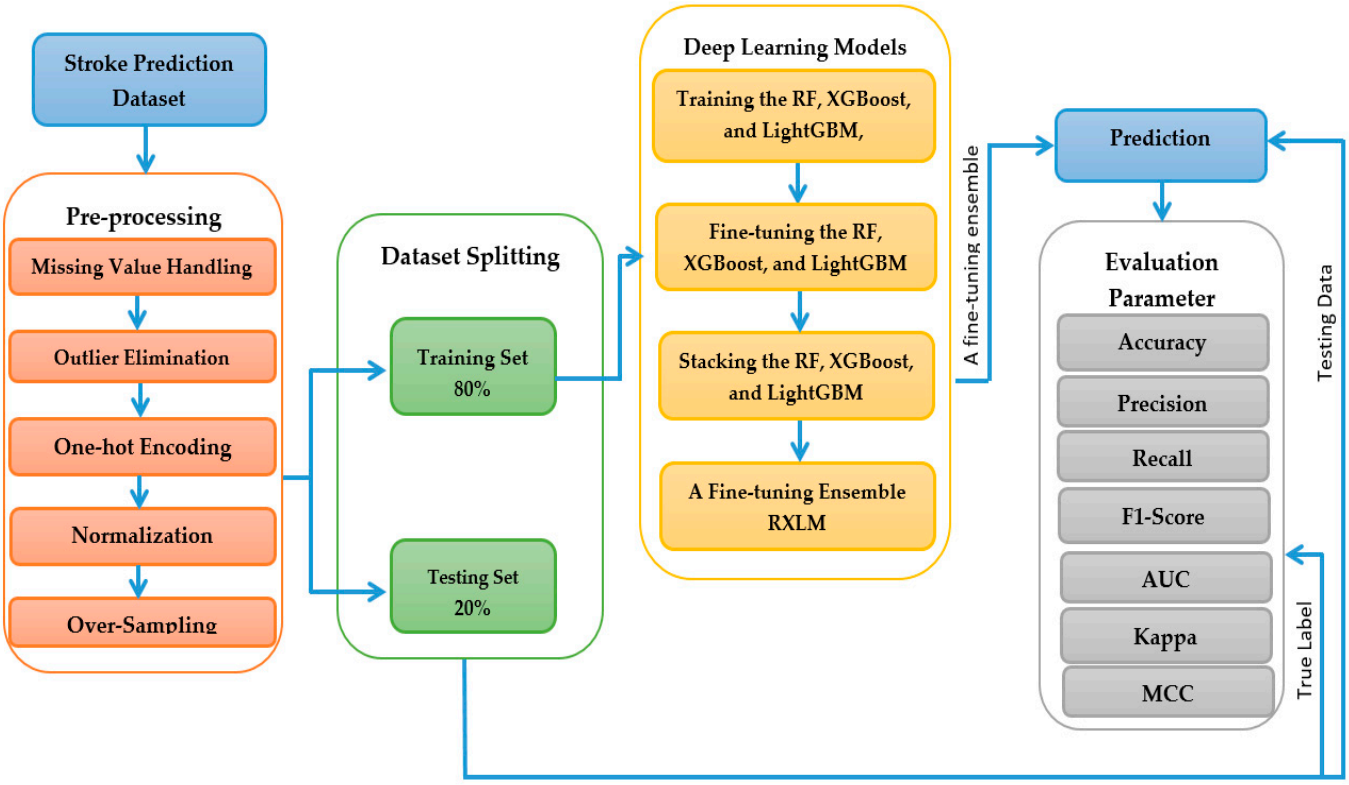

Figure 1. The methodology for the tuning ensemble RXLM for cerebral stroke prediction.

At the beginning of our experiment, we pre-processed the dataset by replacing missing values, eliminating outliers, applying one-hot encoding, and normalizing the data. In the second step, we split the stroke dataset into a training set with 4088 rows (3901 no-stroke and 187 strokes) (80%) and a test set with 1022 rows (20%). We applied the ten-fold cross-validation technique that used more than one train–test split of the data to evaluate the RF, XGBoost, LightGBM, and the proposed ensemble models. The ten-fold cross-validation uses one fold as a test set and nine folds as train sets. It selected various values for the train–test split function's random state seed parameter. By running multiple training–test splits and then averaging the results, cross-validation provided estimates of how the four models were likely to perform, on average, more stably and reliably than those based solely on a single training set. In the third step, we oversampled the training set using the SMOTE. The SMOTE increased the stroke class to 3901 samples. In the fourth step, we trained the three ML models: RF, XGBoost, and LightGBM, on the training set, and we tuned their parameters using the random search method. In the fourth step, we stacked the three tuning ML techniques, proposed the tuning ensemble RXLM, and trained it on the training set. The random search method tunes the hyper-parameters to get the best parameter values. Finally, we applied the test set to evaluate the effectiveness of the proposed tuning ensemble by calculating the evaluation metrics. We compared its performance with the performance of the three ML models.

3.3. Data Pre-Processing of Stroke Dataset

The final prediction may be less accurate than the raw data because of missing values or noisy data. The pre-processing stage in the experiment must be applied to make the data more suitable for mining and analysis. It includes replacing the missing values, outliers' elimination, over-sampling, one-hot encoding, and normalization.

3.3.1. Missing Values Handling

In the stroke prediction dataset, 201 missing BMI feature values are 3.93% of the total BMI feature values. The missing BMI values were predicted using the KNN Imputer

technique. The mean value from the parameter n_neighbors replaces the missing data in the KNN Imputer technique, which refers to the training set's closest neighbors. KNN Imputer uses of the KNN method. It imputes the missing values using the Euclidean distance metric by default.

3.3.2. Outlier Elimination

Calculating the top and lower boundaries of the feature allowed the outliers to be removed. The interquartile range was determined, as well as the first and third quartiles, for each feature. The avg_glucose_level feature had 166 outliers removed, meaning 166 rows were lost. Therefore, the dataset contains approximately 4944 rows, where 4717 rows reflect the normal cases with no stroke, while 227 rows reflect the stroke cases.

3.3.3. One-Hot Encoding

In one-hot encoding, k binary features that can only take the values 0 or 1 replace the categorical feature with k possible values and $k > 2$. One of these k features, the hot feature, is exactly equal to 1, hence the name one-hot encoding. If the categorical feature only has two possible values, 0 or 1 takes their place. The dataset that implements the one-hot encoding algorithm is shown in Table 2.

Table 2. The stroke prediction dataset after applying the one-hot encoding.

Age	Avg Glucose Level	BMI	Gender Male	Hypertension_1	Heart_Disease_1	Ever_Married_Yes	Work_Type Never_Worked
67	5.432367	3.600048	1	0	1	1	0
61	5.309307	5.309307	0	0	0	1	0
80	4.662684	4.662684	1	0	1	1	0
49	5.143008	5.143008	0	0	0	1	0
79	5.159745	5.159745	0	1	0	1	0

3.3.4. Normalization

There are several features in the stroke prediction dataset, including various numerical values. A good model has the potential to learn to choose a relatively small weight when a feature's possible range of values is large. Similarly, a weight's reasonable value will be relatively large when the feature's possible values are small. As a result, the prediction process is impacted by these various feature values, and the training process moves slowly to attain the cost functions.

To find a much more direct path to the global minimum of the cost functions, we need to rescale the features so that their ranges of values are comparable. Additionally, the training process needs to run quickly. We used Z-score normalization to implement feature scaling, which allows features with very different ranges of values to have similar ranges of values.

In Z-score normalization, we calculate the feature's standard deviation and mean value, subtract each feature value from the mean, and divide it by the standard deviation. Equation (1) defines the Z-score normalization [23]. Consequently, if the feature value matches all feature values' mean, the stroke prediction dataset is normalized to 0. A number below the mean will be negative, while a number above the mean will be positive. Table 3 shows the dataset after applying the Z-score normalization.

$$X_{norm} = \frac{(value - \mu)}{\sigma} \quad (1)$$

where: μ refers to the mean of the features, and σ refers to the standard deviation of the features calculated from the Z-score normalization.

Table 3. The stroke prediction dataset after applying the normalization.

Age	Avg Glucose Level	BMI	Gender Male	Hypertension_1	Heart_Disease_1	Ever_Married_Yes	Work_Type Never_Worked
67	2.320709	1.027679	1	0	1	1	0
61	1.980714	0.781547	0	0	0	1	0
80	0.194204	0.574693	1	0	1	1	0
49	1.521257	0.791320	0	0	0	1	0
79	1.567499	−0.581283	0	1	0	1	0

3.3.5. Over-Sampling

Since the training set of the Stroke Prediction dataset had 3901 rows for the no-stroke class and 187 rows for the stroke class, it was an imbalanced set because there was a significant skew in the distribution of the two classes. Many ML algorithms, where the ratio of positive to negative samples is very skewed and very far from 50:50, can be influenced by this bias in the training set, causing the usual error metrics such as accuracy not to work very well. Therefore, class balancing through an over-sampling technique should be applied to the training set. We adjusted the unbalanced participant distribution between the two-stroke and non-stroke in the training set using the SMOTE to balance the samples of the two classes. In the SMOTE, the stroke class in the training set was oversampled to 3901 rows. Therefore, after applying the SMOTE, the training set had 3901 rows for the class no-stroke (normal) and 3901 for the class stroke. Hence, the training set became balanced, and the ratio of stroke cases and no strokes was 50:50.

3.4. XGBoost

XGBoost is scalable, fast, and the most widely used ML method for implementing decision tree ensembles because it can handle large datasets and achieve cutting-edge efficiency in many tasks of classification and regression. It runs quickly, the open-source implementations are simple to use, and it has been used to win numerous ML competitions and in numerous commercial applications with great success. It is an ensemble learning technique that produces a stronger prediction by combining the predictions of multiple weak models. In XGBoost, we will examine the trained decision trees thus far and the samples we are still struggling with. When we build the next decision tree, we will pay more attention to the examples where we are failing. Therefore, rather than examining all of the training examples, we concentrate more on the subset that still needs to perform well, resulting in the new decision tree. XGBoost has built-in regularization to prevent overfitting.

3.5. LightGBM

It is a gradient-boosting ML method. Parallel training, sparse optimization, early stopping, multiple loss functions, regularization, and bagging are just a few of Light GBM's many advantages over XGBoost. How the trees are constructed is a major distinction between the two. The LightGBM does not develop a tree level-by-level, row-by-row. Instead, it branches into trees and selects the leaf with the highest reduction. In addition, the sorted-based decision tree learning technique looks for the ideal split point based on sorted feature values and is not used by LightGBM. A significantly improved histogram-based decision tree learning calculation is carried out by LightGBM, which has a remarkable impact on both productivity and memory usage. In order to operate more quickly while maintaining high precision, the LightGBM algorithm uses two special approaches called Exclusive Feature Bundling (EFB) and Gradient-Based One-Side Sampling (GOSS).

4. Implementation and Evaluation

4.1. Tuning Parameter Using Random Search Optimization

Hyper-parameters are used in ML models. An ML model's hyper-parameters are points of choice or configurations that can be tailored to a specific task or dataset. In

addition, ML models have parameters, which are the internal coefficients set when the model is trained or optimized using a training dataset. Hyper-parameters are not the same as parameters; automatically learned parameters direct the learning process, and hyper-parameters are manually set. The process of optimizing involves defining a search space. Geometrically, this can be considered an n-dimensional volume, with each hyper-parameter representing a different dimension and the values the hyper-parameter can take on, such as real-valued, integer-valued, or categorical, based on the scale of the dimension.

Random search is the simplest and most widely used optimization technique. Random search is a method for training a model that selects and applies random hyper-parameter combinations. The best possible combinations of random hyper-parameters are used. For each hyper-parameter, we select values from a statistical distribution. For a random search, a sampling distribution is established for each hyper-parameter. We can identify the models' numbers to train using a random search. We used the random search algorithm to tune the hyper-parameter in this step. The best parameters for the RF, XGBoost, and LightGBM models are presented in Tables 4–6. These tables show the meaning and best value for each parameter.

Table 4. Tuned parameters for RF.

Parameter	Meaning	Best Value
bootstrap	Bootstrapping generates simulated datasets by resampling the original dataset with replacement many thousands of times.	True
ccp alpha	Minimal cost-complexity pruning (ccp) employs a complexity parameter.	0.0
class weight	Weights that are related to classes.	None
criterion	The capability to assess a split's quality.	gini
Max depth	The leaf and root nodes are measured by the maximum number of levels in the tree.	None
Max features	The aspect number must be taken into consideration when selecting the best data split.	Auto
Max leaf nodes	The trees are grown using the maximum nodes of the leaf.	None
Max samples	How many samples from X need to be taken to train each base estimator	None
Min impurity decrease	A split node will experience an impurity reduction greater than or equal this amount.	0.0
Min samples leaf	The smallest number of required samples at each leaf node.	1
Min samples split	This setting instructs the decision tree in a random forest to divide any node with less observation than necessary.	2
Min weight fraction leaf	A leaf node must contain a minimum weighted percentage of the total weights from all of the input samples.	0.0
N estimators	This parameter refers to the number of trees required to be built before taking the maximum voting or averages of predictions.	100
Random state	Random number seed.	123
N jobs	The number of concurrent jobs to run	−1
Oob score	Score from an out-of-bag estimate of the training dataset.	False
verbose	Regulates the amount of jargon used in fitting and predicting.	0
Warm start	If True is set, use the previous fit call's solution and add more estimators to the ensemble; if not, just fit a new forest.	False

Table 5. Tuned parameters for LightGBM.

Parameter	Meaning	Best Value
Boosting type	Gradient boosting methods.	gbdt
Class weight	Weights that are related to classes.	None
Colsample bytree	Column subsample ratio used to construct each tree.	1.0
Importance type	The kind of importance of a feature that should be entered into feature_importances.	split
Learning rate	The learning rate of boosting.	0.4
Max depth	The leaf and root nodes are measured by the maximum number of levels in the tree.	−1
Min child samples	The minimum level of data required for a child.	6
Min child weight	Minimum required child weight.	0.001
Min split gain	The minimum loss reduction is required to construct a second partition on a tree leaf node.	0.3
N estimators	How many boosted trees can fit?	20
N jobs	The number of concurrent jobs to run.	−1
Num leaves	Maximum number of base learners' tree leaves.	150
Objective	Choose a custom objective function or the associated learning objective and task.	None
Random state	Random number seed.	123
Reg alpha	Represents L_1 regularization parameter on weight.	0.005
Reg lambda	Represents L_1 regularization parameter on weight.	0.0005
Subsample	The ratio of the subsamples in the training instance.	1.0
Subsample for bin	A number of samples are needed to make bins.	200,000
Subsample freq	A frequency of 0 indicates that there is no enablement.	0
Verbosity	Whether messages are printed during construction.	warn

Table 6. Tuned parameters for XGBoost.

Parameter	Meaning	Best Value
Objective	Output probability, logistic regression for binary classification.	binary:logistic
Base score	Global bias, the overall initial prediction score.	0.5
booster	Gradient boosting methods.	gbtree
Colsample bylevel	Ratio of columns in the subsample for each level.	1
Colsample bynode	The column-to-node subsample ratio (split).	1
Colsample bytree	Column subsample ratio used to construct each tree.	1
Learning rate	The learning rate of boosting.	0.300000012
Max bin	The maximum number of discrete bins can be used to group continuous features.	256
gamma	The least loss reduction is needed for a subsequent part on a tree leaf node.	0
Max cat to onehot	A threshold for determining whether categorical data should be split using a one-hot encoding-based split in XGBoost.	4
Max delta step	Utilized to safeguard optimization.	0

Table 6. *Cont.*

Parameter	Meaning	Best Value
Max depth	The leaf and root nodes are measured by the maximum number of levels in the tree.	6
Min child weight	Minimum required child weight.	1
N estimators	How many boosted trees can fit?	100
N jobs	The number of active jobs at once.	−1
Num parallel tree	The number of parallel trees that are built in each iteration.	1
Predictor	Which predictor algorithm to employ? It allows the use of a GPU or CPU but delivers the same results.	auto
Eval metric	Metrics for evaluation of validation data.	None
Random state	Random number seed.	123
Reg alpha	Represents L_1 regularization parameter on weight.	0
Reg lambda	Represents L_1 regularization parameter on weight.	1
Sampling method	How to sample the training scenarios using this method.	Uniform
Tree method	The XGBoost tree construction algorithm.	Auto
Scale Pos weight	Adjust the balance between positive and negative weights, which is helpful for unbalanced classes.	1
Subsample	The ratio of the training instances' subsamples.	1
Validate parameters	XGBoost will validate input parameters to determine whether a given parameter is utilized when the value is True.	1

4.2. Evaluation Metrics

The Kaggle environment was used to implement and evaluate the three ML models and the proposed tuning ensemble RXLM. Data scientists and developers can host their datasets, share their code, and compete in ML competitions with Kaggle. The three ML models and the suggested tuning ensemble RXLM were assessed by applying the different predictions of the confusion matrix, as follows [23,24]:

- The true positive (TP_x) of class x means that the observed value output and predicted value output for class x are true or correct.
- The true negative (TN_x) of class x means that the output of the testing dataset is negative compared to the predicted output of class x.
- The false positive (FP_x) of class x means that the output of the testing dataset is positive, whereas the predicted output of the experiments is false.
- The false negative (FN_x) of class x means that the observed value output and predicted value output for class x are false or incorrect.

In addition, various evaluation metrics are used to evaluate the overall performance of the experimental results, as follows [23]:

- Accuracy is the degree to which the result matches the required value. It is calculated by measuring the overall number of correct expectations or predictions on the dataset by the total number of expectations.

$$\text{Accuracy} = \frac{(TP_x + TN_x)}{(TP_x + FP_x + TN_x + FN_x)} \tag{2}$$

- Precision is the percentage of positive samples that the model properly predicted.

$$\text{Precision} = \frac{TP_x}{(TP_x + FP_x)} \tag{3}$$

- The recall is the percentage that correctly identifies the positive samples from the overall number of samples.

$$\text{Recall} = \frac{TP_x}{(TP_x + FN_x)} \tag{4}$$

- Sensitivity is the proportion of true positives to actual positive values of the samples.

$$\text{Sensivity} = \frac{TP_x}{(TP_x + FN_x)} \tag{5}$$

- Specificity is the ratio of the number of actual negatives to the number of true negative values.

$$\text{Specifity} = \frac{TN_x}{(TN_x + FP_x)} \tag{6}$$

- F1-score is the harmonic average of precision and sensitivity.

$$\text{F1-score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

- AUC is a number used to summarize a classifier's performance by determining the total area beneath the receiver operating characteristic (ROC) curve. The Mathew correlation coefficient (MCC) is a statistical method for assessing models. It performs the same function as chi-square statistics for a 2 × 2 contingency table, measuring the difference between expected and actual values.

$$\text{MCC} = \frac{(TP_x \times TN_x) - (FP_x \times FN_x)}{\left(\left(TP_x + FP_x\right)(TP_x + FN_x)(TN_x + FP_x)(TN_x + FN_x)\right)^{0.5}} \tag{8}$$

- The Kappa Coefficient, or Cohen's Kappa score, is used with two raters, but can also be modified to work with more than two raters. One of the raters takes on the role of the classification model in ML binary classification models. In contrast, the other rater assumes the role of the real-world observer who is aware of the true categories of each record or dataset. Cohen's Kappa can be used to calculate overall agreement and agreement after chance has been considered, and it considers the number of agreements (TP_x and TN_x) and disagreements (FP_x and FN_x) between the raters.

$$\text{Kappa} = \frac{2 \times (TP_x \times TN_x) - (FP_x \times FN_x)}{((TP_x + FP_x) \times (FP_x + TN_x) \times (TP_x + FN_x) \times (TN_x + FN_x))} \tag{9}$$

4.3. Model Evaluation

This section discusses the two experiments' outcomes regarding the prediction of cerebral stroke. The first experiment was performed before hyper-parameter optimization, and the second, after hyper-parameter optimization. The accuracy of cerebral stroke prediction has been improved by using the proposed tuning ensemble RXLM based on the tuning RF, XGBoost, and LightGBM. The evaluation of the proposed tuning ensemble RXLM and the tuning three RF, XGBoost, and LightGBM was performed using the ten-fold cross-validation method that uses more than one train–test split of the data to evaluate the RF, XGBoost, LightGBM, and the proposed ensemble models. The ten-fold cross-validation uses one-fold as a test set and nine folds as train sets. It selected various values for the train–test split function's random state seed parameter.

Figures 2–5 showed the first experiment's results. In this experiment, we evaluated the RF, XGBoost, LightGBM, and RXLM with measurement metrics before hyper-parameters optimization. The ensemble RXLM had 96.08%, 99.2%, 95.5%, 96.65%, 96.06%, 92.16%, and 92.2% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The RF

had 94.49%, 98.84%, 96.56%, 92.73, 94.6%, 88.98%, and 89.06% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The XGBoost achieved 95.29%, 99.13%, 96.38%, 94.36%, 95.35%, 90.59%, and 90.63% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The LightGBM achieved 94.82%, 98.89%, 96.19%, 93.64%, 94.89%, 89.64%, and 89.69% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The accuracy of the RF, XGBoost, and LightGBM was 94.49%, 95.29%, and 94.82 %, respectively. The accuracy, precision, F1-score, Kappa, and MCC of ensemble RXLM were the highest, at 96.08%, 96.65%, 96.06%, 92.16%, and 92.2%, respectively. The XGBoost achieved the highest AUC, at 99.13%. The RF achieved the highest recall, at 96.56%. The RF had the lowest accuracy, AUC, precision, F1-score, Kappa, and MCC, at 94.4%, 98.84%, 92.73%, 94.6%, 88.98%, and 89.06%. The RXLM had the lowest recall, at 95.5%.

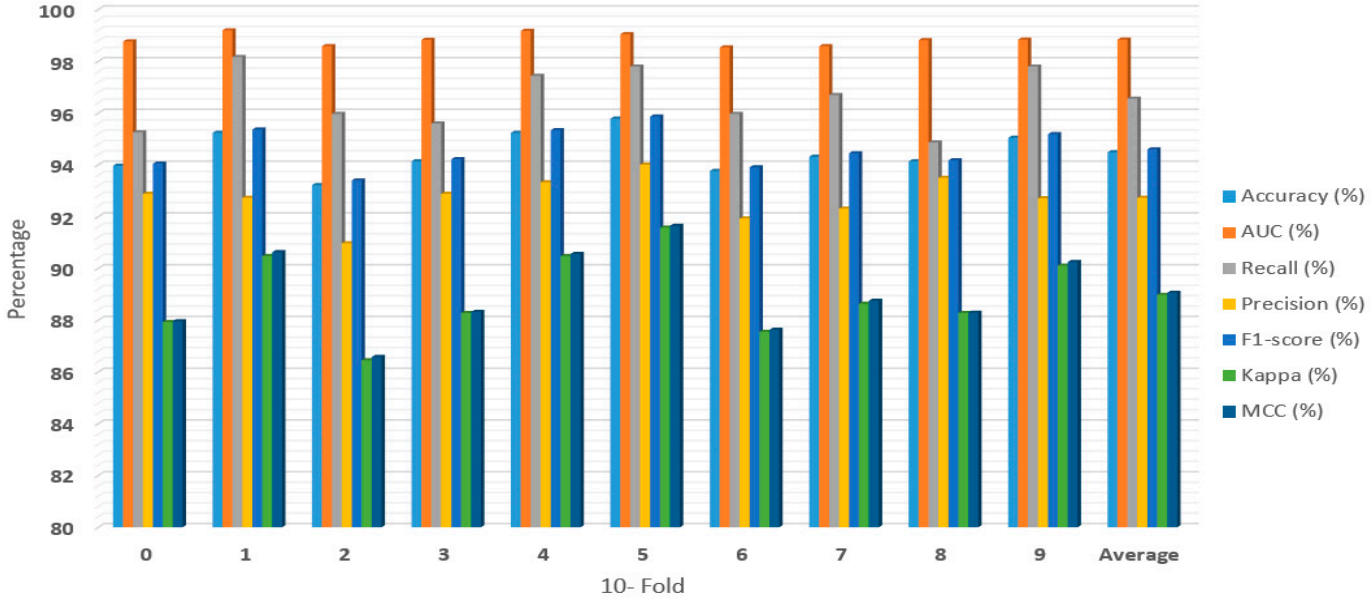

Figure 2. The metrics of the RF model before hyper-parameters tuning.

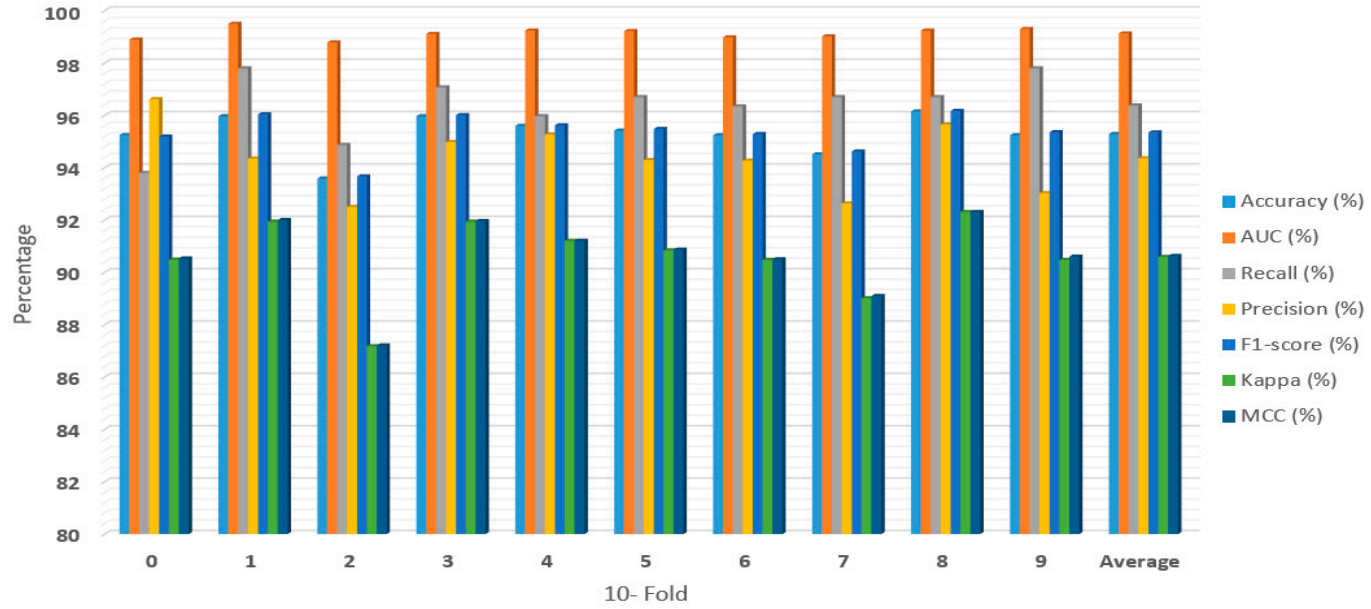

Figure 3. The metrics of the XGBoost model before hyper-parameters tuning.

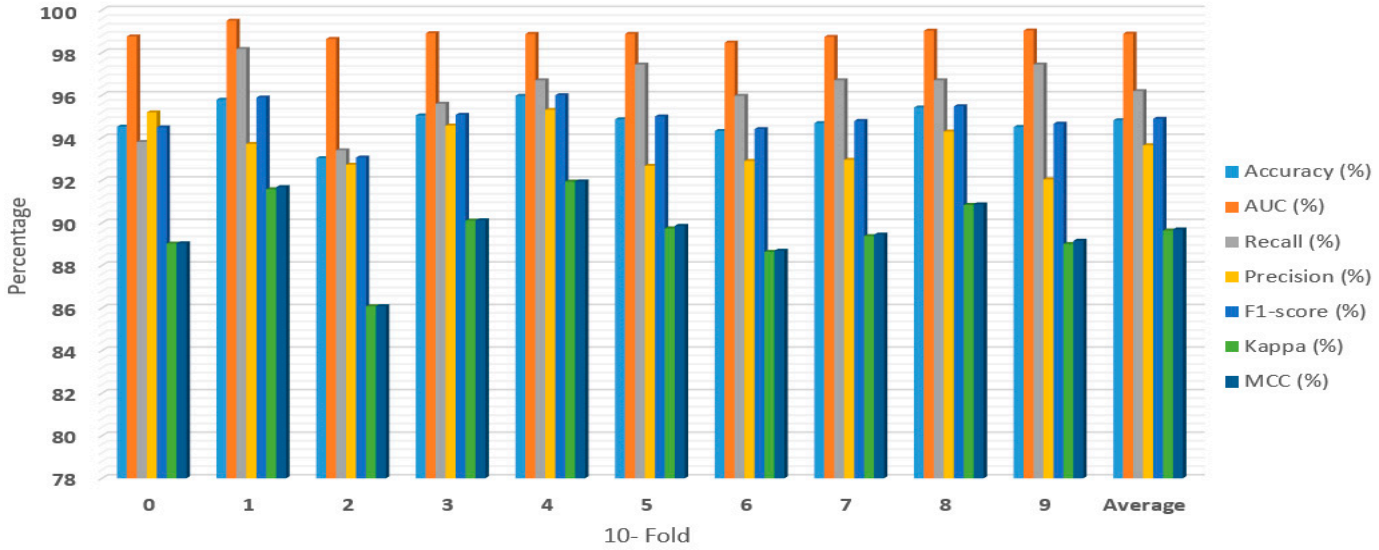

Figure 4. The metrics of the LightGBM model before hyper-parameters tuning.

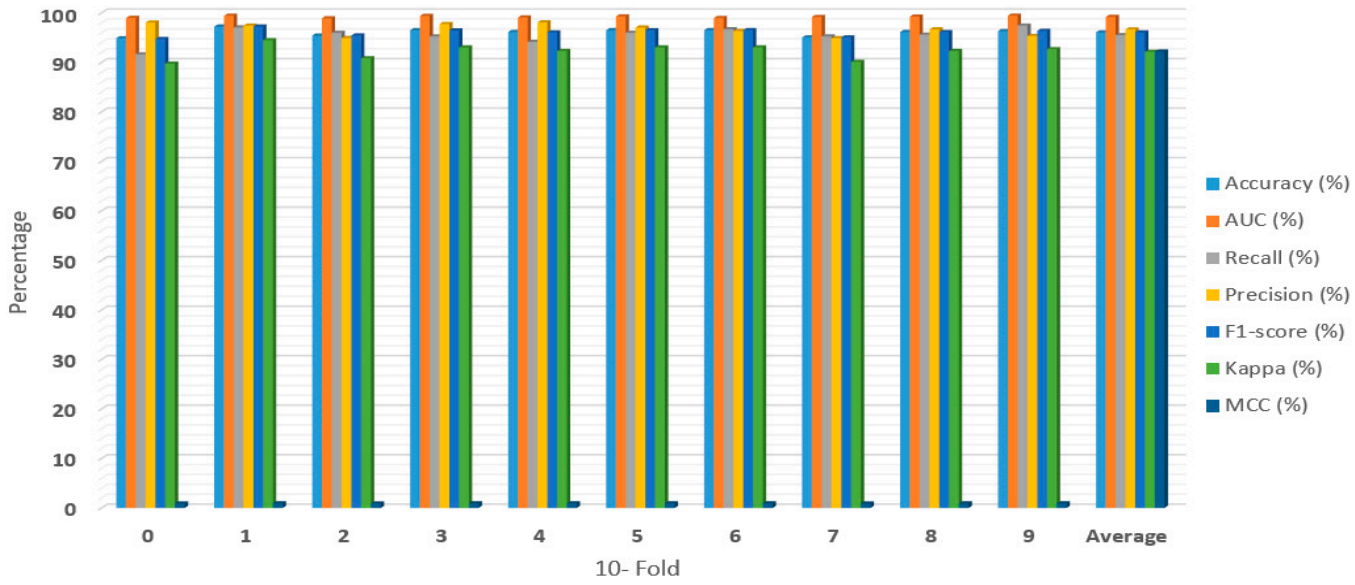

Figure 5. The metrics of the RXLM model before hyper-parameters tuning.

Figures 6–9 showed the results of the second experiment. In the second experiment, we used the random search technique to evaluate the RF, XGBoost, LightGBM, and RXLM with measurement metrics after tuning the hyper-parameters on the training set. The proposed ensemble RXLM achieved 96.34%, 99.38%, 96.12%, 96.55%, 96.33%, 92.68%, and 92.69% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The RF had 84.73%, 92.75%, 89.27%, 81.88%, 85.4%, 69.46%, and 69.78% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The XGBoost had 92.42%, 99.12%, 98.65%, 87.74%, 92.87%, 84.84%, and 85.52% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The LightGBM had 95.18%, 98.86%, 94.91%, 95.45%, 95.17%, 90.37%, and 90.39% for accuracy, AUC, recall, precision, F1-score, Kappa, MCC, respectively. The accuracy, AUC, precision, F1-score, Kappa, and MCC of ensemble RXLM were the highest, at 96.34%, 99.38%, 96.55%, 96.33%, 92.68%, and 92.69%, respectively. The XGBoost achieved the highest recall, at 98.65%. The RF had the lowest accuracy, AUC, recall, precision, F1-score, Kappa, and MCC, at 84.73%, 92.75%, 89.27%, 81.88%, 85.4%, 69.46%, and 69.78%, respectively.

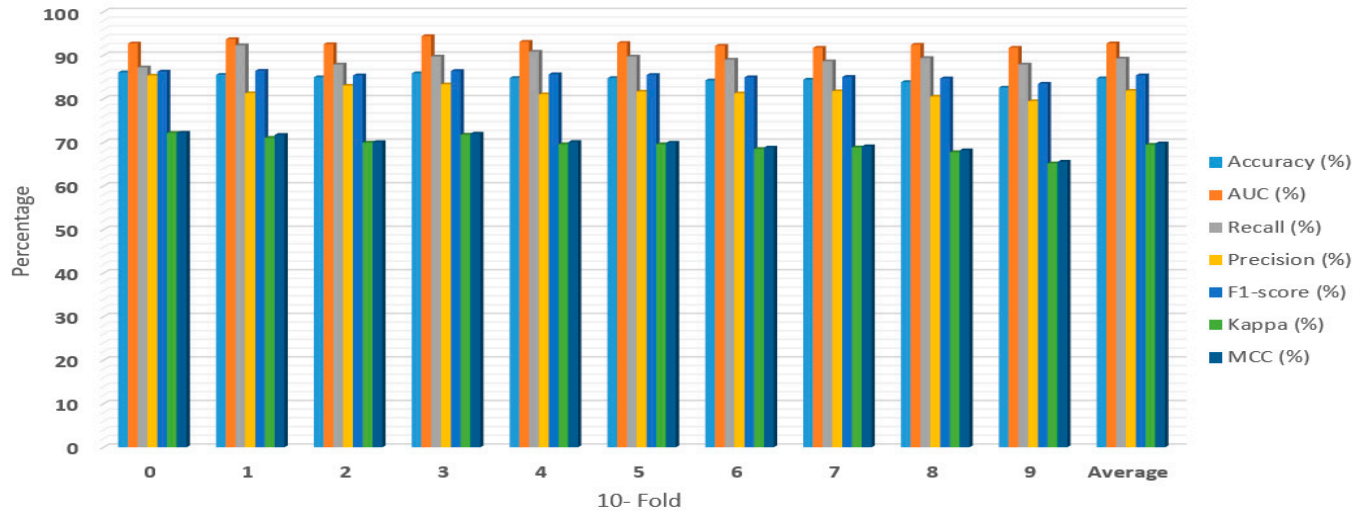

Figure 6. The metrics of the RF model after tuning the hyper-parameters.

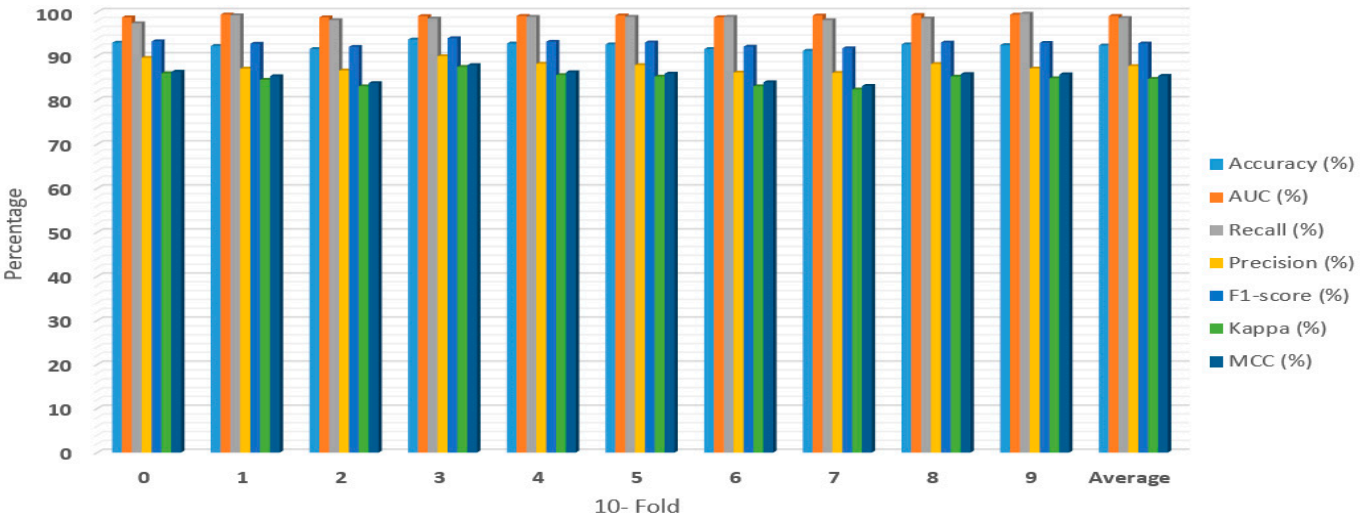

Figure 7. The metrics of the XGBoost model after tuning the hyper-parameters.

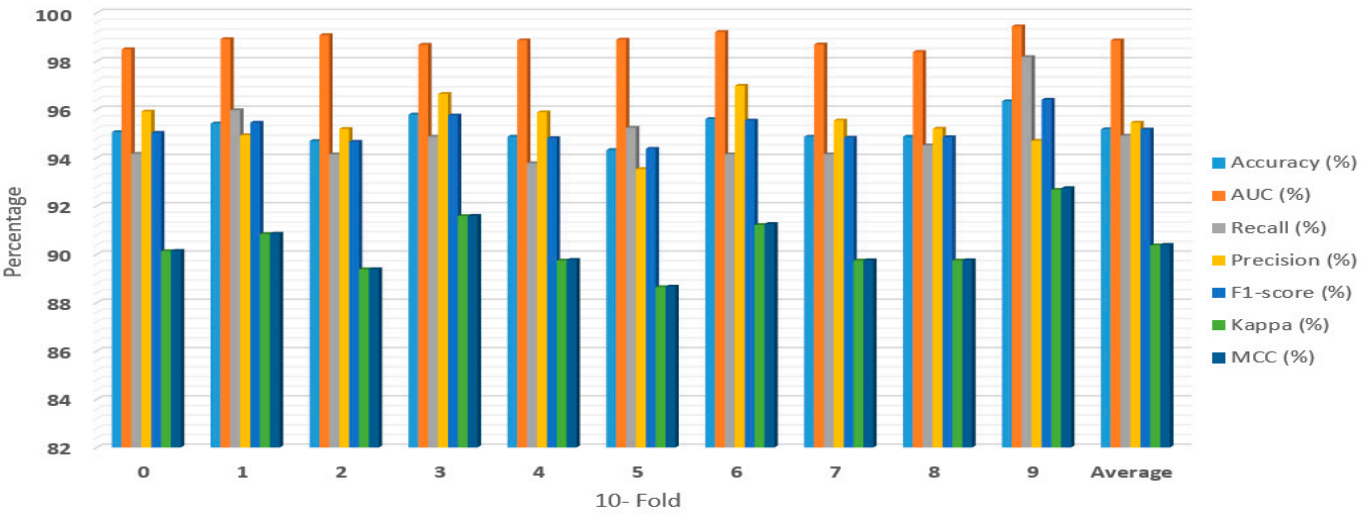

Figure 8. The metrics of the LightGBM model after tuning the hyper-parameters.

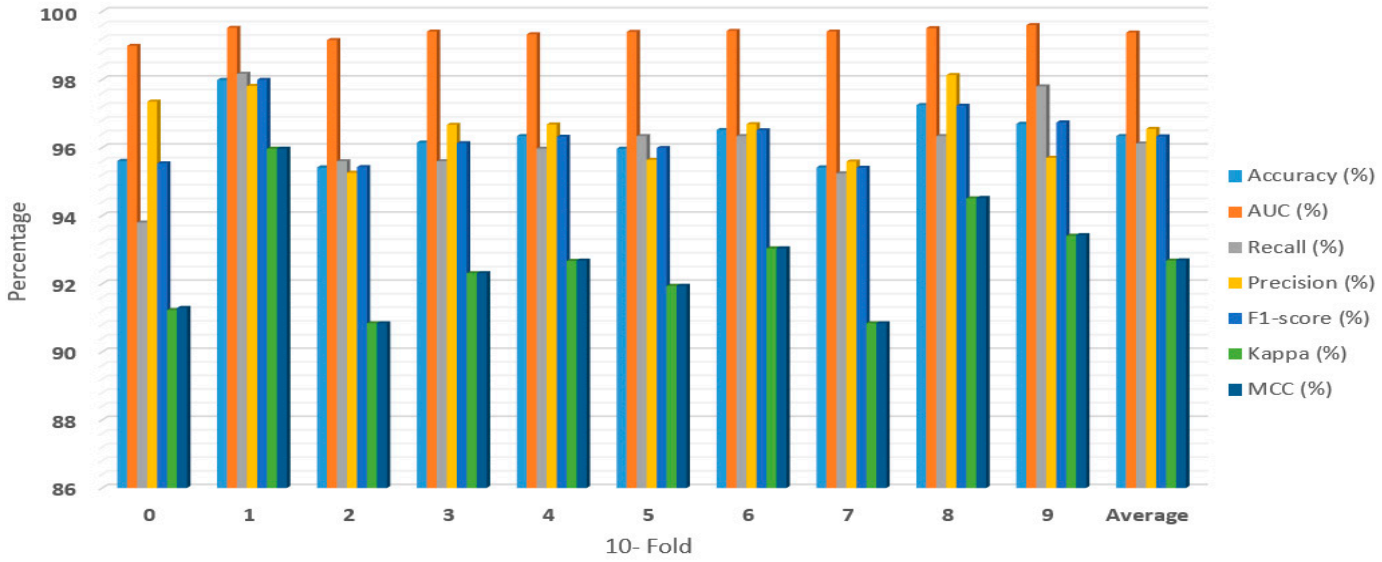

Figure 9. The metrics of the RXLM model after tuning the hyper-parameters.

Using the confusion matrix (CM) of the first experiment, an examination of the proposed RBSVM and the five ML models is depicted in Figures 10–13 by contrasting the actual and predicted labels of the stroke and no-stroke classes of the test dataset. The stroke class had 968 rows, and the no-stroke class had 976. The KNNs, the RF, the CatBoost, the XGBoost, and the proposed ensemble RBSVM correctly predicted 986 cases for class stroke, achieving 100% accuracy. With 880 examples accurately predicted, the SVM model has a 90.9% accuracy rate. For the no-stroke class, the RBSVM model was 94.1% accurate in predicting 919 samples. The KNNs model correctly predicted 838 samples out of 976, giving it an accuracy of 85.8%. The SVM model correctly predicted 755 samples, giving it a 77.3% accuracy. With 961 samples properly predicted, the RF model has a 98.4% accuracy rate. While predicting 912 samples, the CatBoost model had a 93.4% accuracy rate, and the XGBoost model had a 95.5% accuracy rate.

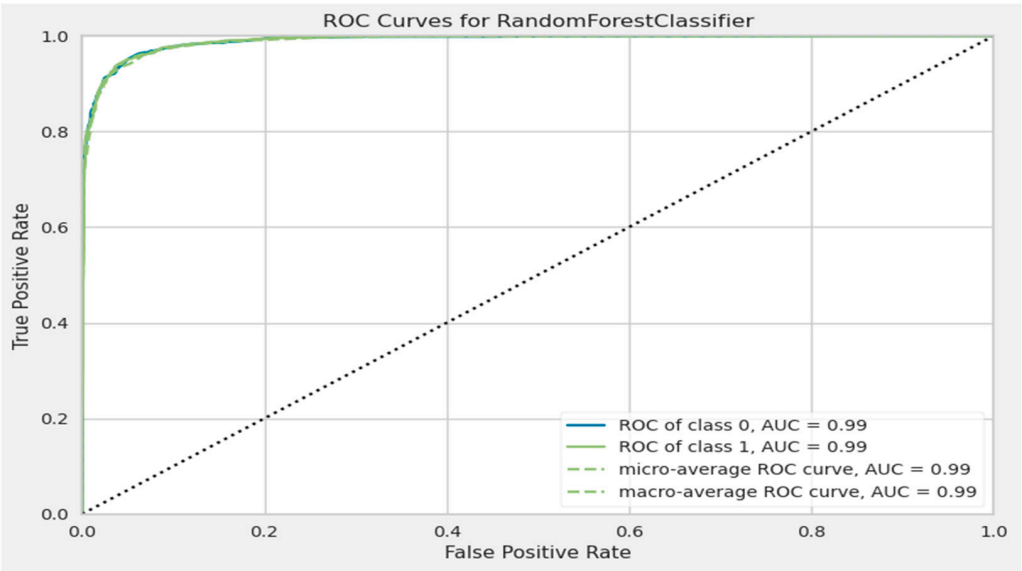

Figure 10. The ROC curve for the RF model after the hyper-parameter tuning.

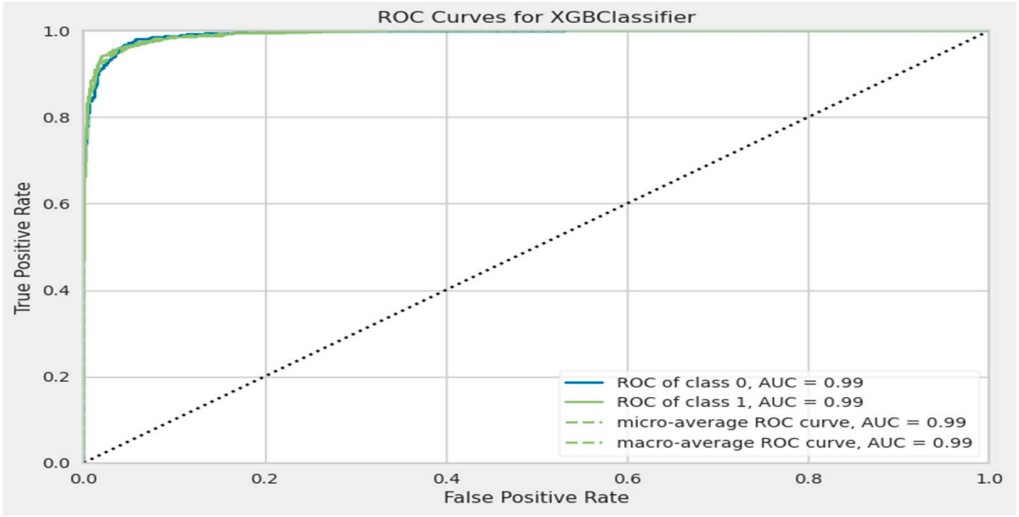

Figure 11. The XGBoost model's ROC curve after the hyper-parameter tuning.

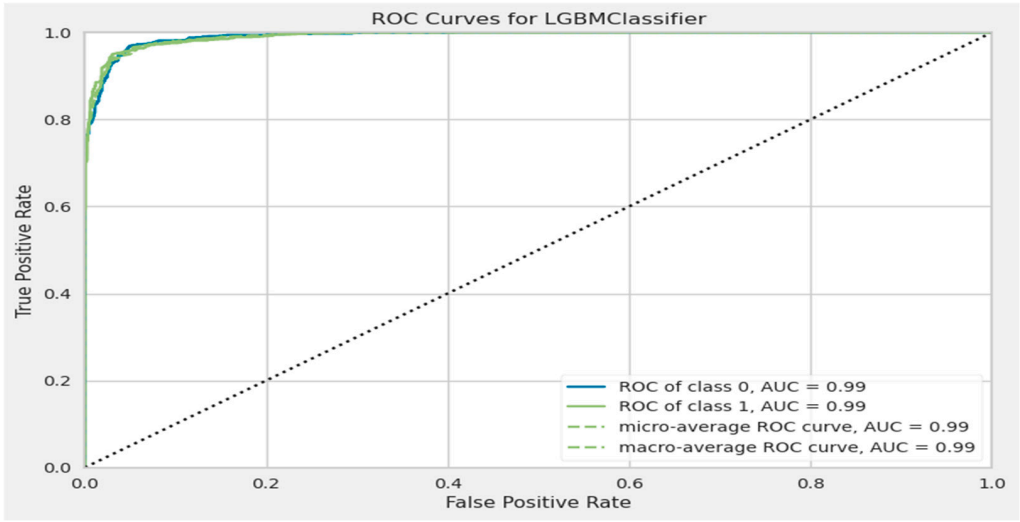

Figure 12. The lightGBM model's ROC curve after the hyper-parameter tuning.

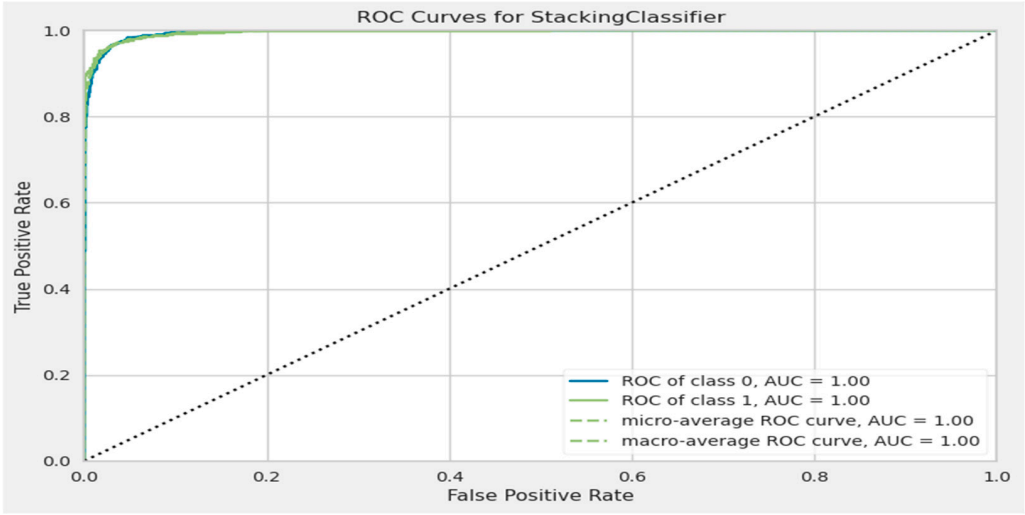

Figure 13. The RXLM model's ROC curve after the hyper-parameter tuning.

4.4. Ensemble's Result Comparison with the Literature

Current methods and the proposed ensemble are contrasted to demonstrate the novelty of the proposed ensemble. The various ML and deep learning classification strategies for cerebral stroke are presented in Table 7. The two most recent studies are [15,17], with 98% and 99% accuracy rates, respectively. Since the accuracy of the proposed RXLM was 96.97%, our proposed ensemble RXLM outperformed the most current methods. Hence, the proposed ensemble RXLM will be used by doctors to detect early cerebral stroke effectively.

Table 7. Comparative results of the proposed RBSVM and recent ML algorithms.

Ref.	Methodology	Accuracy	Feature Selection	Datasets
[14]	RF, DT, and NB	94.781%	No	Stroke prediction dataset
[15]	NB, RF, LR, KNN, SGD, DT, MLP, Majority Voting, and Stacking, which was the best.	98%	No	Stroke prediction dataset
[16]	LR, DT, Voting, and RF, which was the best.	96%	No	Stroke prediction dataset
[17]	SVM, KNN, DT, MLP, and RF which was the best	99%	Recursive Feature Elimination	Dataset of medical record
[18]	LR, DT, KNN, SVM, RF, and NB, which was the best	82%	No	Stroke prediction dataset
[19]	SVM, LASSO, and NN, which was the best	79%	Perceptron Neural Network and (PCA)	Electronic health records dataset
[20]	NB, LR, Logistic R, KNN, DT, AdaBoost, Improvised RF	96.97%	NIHSS	Medical record
[21]	LR, SVM, ANN, XGBoost, and RF, which was the best	97%	No	Stroke prediction dataset
Proposed RXLM	RF, XGBoost, and LightGBM	96.34%	No	Stroke prediction dataset

Accelerating the diagnosis process and delaying the disease's progression will assist the patient in lowering the cost of diagnosis. The evaluation of the suggested ensemble RXLM showed that it reaches an unmatched level of perceptiveness with hyper-parameters optimization compared to previous models. As presented in Table 7, the proposed RXLM provided high results with 96.34% accuracy, which is considered relatively lower than other proposed methodologies. The proposed methods presented in [15,17,20,21] applied oversampling techniques used in datasets to address class imbalance issues by increasing the number of samples in the minority class. However, if we use oversampling before splitting the data into training and testing sets, the same data points may end up in both. As a result, the model may overestimate its performance on the testing set, resulting in overly optimistic results. The problem with oversampling data before splitting can cause data leakage, leading to model overfitting and biased results. Oversampling was exclusively applied to the training data after splitting it into training and testing sets to prevent data leakage in the research. This approach ensures that the model can effectively generalize to unseen data, leading to a more precise estimation of its performance on new data. However, this model may perform slightly worse than other research studies that utilize oversampling before splitting, as such methods can lead to optimistic performance estimates due to data leakage.

5. Conclusions

In this research, we proposed a robust and tuned ensemble RXLM using the random search algorithm. The proposed ensemble improves the diagnosis of cerebral stroke disease. It will speed up diagnosis and halt the progression of the cerebral stroke disease, while also assisting doctors in efficiently detecting early cerebral stroke disease. We began by

pre-processing the Stroke Prediction dataset by employing the KNN Imputer technique for handling missing values, eliminating outliers, using one-hot encoding, and normalizing the features with different ranges of values. After splitting the dataset, we implemented the SMOTE to achieve a balance between the samples of the two classes of the training dataset.

Additionally, a random search technique was used to fine-tune the hyper-parameters of the three ML models (RF, XGBoost, and LightGBM) to select the optimal parameter values. The proposed ensemble RXLM was evaluated and compared to the recent ML models. The proposed ensemble RXLM's accuracy, AUC, recall, precision, F1-score, Kappa, and MCC were 96.34%, 99.38%, 96.12%, 96.55%, 96.33%, 92.68%, and 92.69%, respectively. The proposed tuned ensemble outperformed the most recent classifiers in terms of accuracy.

Author Contributions: Data curation, M.A., M.E.; formal analysis, S.A.E.-G., M.E. and A.A.A.E.-A.; investigation, A.M.M., M.A. and M.E.; conceptualization, S.A.E.-G., A.M.M., A.A.A.E.-A.; methodology, M.E., M.A.; supervision, M.A. and A.M.M.; writing—original draft, S.A.E.-G. and AA; writing—review and editing, A.M.M., A.A.A.E.-A. and S.A.E.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the DEANSHIP OF SCIENTIFIC RESEARCH—JOUF UNIVERSITY, grant number IF_JU_2_205.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Furnished on request.

Acknowledgments: The authors acknowledge the Deanship of Scientific Research at Jouf University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Katan, M.; Luft, A. Global burden of stroke. *Semin. Neurol.* **2018**, *38*, 2018–2211. [CrossRef] [PubMed]
2. Bustamante, A.; Penalba, A.; Orset, C.; Azurmendi, L.; Llombart, V.; Simats, A.; Pecharroman, E.; Ventura, O.; Ribó, M.; Vivien, D.; et al. Blood Biomarkers to Differentiate Ischemic and Hemorrhagic Strokes. *Neurology* **2021**, *96*, 1928–1939. [CrossRef] [PubMed]
3. Learn about Stroke. Available online: https://www.world-stroke.org/world-stroke-day-campaign/why-stroke-matters/learnabout-stroke (accessed on 25 February 2023).
4. Li, G.; Cheng, L.; Gao, Z.; Xia, X.; Jiang, J. Development of an Untethered Adaptive Thumb Exoskeleton for Delicate Rehabilitation Assistance. *IEEE Trans. Robot.* **2022**, *38*, 3514–3529. [CrossRef]
5. Elloker, T.; Rhoda, A. The Relationship between Social Support and Participation in Stroke: A Systematic Review. *Afr. J. Disabil.* **2018**, *7*, a357. [CrossRef]
6. Concept of Stroke by Health Line. Available online: https://www.cdc.gov/stroke/index.htm (accessed on 7 January 2023).
7. Statistics of Stroke by Centers for Disease Control and Prevention. Available online: https://www.cdc.gov/stroke/facts.htm (accessed on 14 March 2023).
8. Banerjee, T.; Das, S. Fifty years of stroke researches in India. *Ann. Indian Acad. Neurol.* **2016**, *19*, 1–8. [CrossRef]
9. Stroke Association. Available online: https://www.stroke.org.uk (accessed on 25 January 2023).
10. Stroke in Canada. Available online: https://www.canada.ca/en/public-health/services/publications/diseases-conditions/stroke-canada-fact-sheet.html (accessed on 9 March 2023).
11. Xia, X.; Yue, W.; Chao, B.; Li, M.; Cao, L.; Wang, L.; Shen, Y.; Li, X. Prevalence and Risk Factors of Stroke in the Elderly in Northern China: Data from the National Stroke Screening Survey. *J. Neurol.* **2019**, *266*, 1449–1458. [CrossRef]
12. Alloubani, A.; Saleh, A.; Abdelhafiz, I. Hypertension and Diabetes Mellitus as a Predictive Risk Factor for Stroke. *Diabetes Metab. Syndr. Clin. Res. Rev.* **2018**, *12*, 577–584. [CrossRef] [PubMed]
13. Boehme, A.; Esenwa, C.; Elkind, M. Stroke risk factors, genetics, and prevention. *Circ. Res.* **2017**, *120*, 472–495. [CrossRef] [PubMed]
14. Adi, N.; Farhany, R.; Ghina, R.; Napitupulu, H. Stroke Risk Prediction Model using Machine Learning. In Proceedings of the International Conference on Artificial Intelligence and Big Data Analytics, Bandung, Indonesia, 27–29 October 2021. [CrossRef]
15. Dritsas, E.; Trigka, M. Stroke Risk Prediction with Machine Learning Techniques. *Sensors* **2022**, *22*, 4670. [CrossRef] [PubMed]
16. Tazin, T.; Alam, M.; Dola, N.; Bari, M.; Bourouis, S.; Khan, M. Stroke Disease Detection and Prediction using Robust Learning Approaches. *J. Healthc. Eng.* **2021**, *2021*, 7633381. [CrossRef] [PubMed]
17. Al-Mekhlafi, Z.; Senan, E.; Rassem, T.; Mohammed, B.; Makbol, N.; Alanazi, A.; Almurayziq, T.; Ghaleb, F. Deep learning and machine learning for early detection of stroke and haemorrhage. *Comput. Mater. Contin.* **2022**, *72*, 775–796. [CrossRef]

18. Sailasya, G.; Kumari, G. Analyzing the Performance of Stroke Prediction using ML Classification Algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 539–545. [CrossRef]
19. Dev, S.; Wang, H.; Nwosu, C.; Jain, N.; Veeravalli, B.; John, D. A predictive analytics approach for stroke prediction using machine learning and neural networks. *Healthc. Anal.* **2022**, *2*, 100032. [CrossRef]
20. Bandi, V.; Bhattacharyya, D.; Midhunchakkravarthy, D. Prediction of Brain Stroke Severity using Machine Learning. *Rev. D'intelligence Artif.* **2020**, *34*, 753–761. Available online: https://www.iieta.org/download/file/fid/48077 (accessed on 18 February 2023). [CrossRef]
21. Alhakami, H.; Alraddadi, S.; Alseady, S.; Baz, A.; Alsubait, T. A Hybrid Efficient Data Analytics Framework for Stroke Prediction. *Int. J. Comput. Sci. Netw. Secur.* **2020**, *20*, 240–250. Available online: http://paper.ijcsns.org/07_book/202004/20200429.pdf (accessed on 7 March 2023).
22. Stroke Prediction Dataset. Available online: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset (accessed on 25 February 2022).
23. Tan, P.; Steinbach, M.; Karpatne, A.; Kumar, V. *Introduction to Data Mining*; Computers; Pearson: New York, NY, USA, 2018; pp. 1–864. Available online: https://www-users.cse.umn.edu/~kumar001/dmbook/index.php (accessed on 4 February 2023).
24. Naser, M.; Alavi, A. Insights into Performance Fitness and Error Metrics for Machine Learning. *arXiv* **2020**. [CrossRef]