JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 9 Number 1 | June 2024: 29-40
DOI: 10.15575/join.v9i1.1249

Analysis of Data and Feature Processing on Stroke Prediction using Wide Range Machine Learning Model

Untari Novia Wisesty[1], Tjokorda Agung Budi Wirayuda[2], Febryanti Sthevanie[3], Rita Rismala[4]
[1,2,3,4]School of Computing, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received December 13, 2023
Revised March 03, 2024
Accepted March 03, 2024
Available Online April 23, 2024

Keywords:

Machine Learning
PCA
Pearson Correlation
Sampling Data
Stroke Prediction

ABSTRACT

Stroke is a disease which cause the death of brain cells, so that the part of the body controlled by the brain loses its function. If not treated immediately, this disease can cause long-term disability, brain damage, and death. In this research, stroke prediction was carried out on the Stroke dataset acquired from the Kaggle dataset using various machine learning models. Then, data sampling techniques are used to handle data imbalance problems in the stroke dataset, which include Random Undersampling, Random Oversampling, and SMOTE techniques. Pearson Correlation and Principal Component Analysis are also used for dimensional reduction and analyzing the important features that are most influential in predicting stroke. Pearson Correlation produces five attributes that have the highest Pearson coefficient, namely age, hypertension, heart disease, blood sugar level, and marital status. Experimental results have demonstrated that the utilization of RUS, ROS, and SMOTE sampling techniques can significantly boost the F1-Score testing by an impressive 43.44%, 34.44%, and 35.55% respectively, as compared to experiments conducted without implementing any data sampling techniques. The highest F1-Score testing was achieved using the Support Vector Machine and Gaussian Naïve Bayes models, namely 0.83.

Corresponding Author:

Untari Novia Wisesty,
School of Computing, Telkom University, Bandung, Indonesia
Jl. Telekomunikasi 1, Terusan Buah Batu, Bandung, 40257, Jawa Barat, Indonesia
Email: untarinw@telkomuniversity.ac.id

1. INTRODUCTION

Stroke is a disease that occurs when a blood vessel to the brain is blocked or ruptures so that the blood supply to the brain is reduced [1]. Lack of blood supply to the brain can cause brain cell death, resulting in the loss of function in the parts of the body controlled by the affected area of the brain. Stroke not only attacks old people but also young people [2]. Strokes are generally caused by high blood pressure, smoking habits, heart disease, genetic factors, obesity, uncontrolled high cholesterol, diabetes, and age factors. Meanwhile, the causes of stroke at a young age include unhealthy lifestyles such as smoking and drinking alcohol, thereby increasing the risk of high blood pressure, cholesterol, and diabetes [2], [3]. Common symptoms of a stroke can include the sudden onset of severe headaches without a known cause, difficulty with balance and coordination of limbs, challenges with walking and speaking, and numbness in the arms, legs, and face, particularly on one side of the body. These symptoms or clinical signs can worsen quickly and last a long time. If stroke symptoms appear, they must be treated immediately to prevent brain damage and long-term disability, as well as death [4].

Electronic Health Record (EHR) is an electronic recording of the health record and illnesses suffered by patients in hospital. The health history is recorded from time to time for each patient. EHR can also be used to record the history of stroke in certain patients along with clinical symptoms. One of the EHRs that contains stroke records is presented in a Kaggle dataset [5], where the data has 10 attributes, namely smoking status, body mass index (BMI), blood glucose level, residence type, work type, marital status, heart disease, hypertension, age, and gender. Based on these attributes, it will be predicted whether someone will suffer a stroke or not. Predicting stroke from the dataset poses challenges due to imbalanced class data and the varying importance of attributes in EHR data for prediction. These difficulties can impact the accuracy and reliability of the predictive model. Therefore, careful consideration and balancing of these factors are essential in developing an effective prediction model for stroke.

Guhdar used SMOTE and Logistic Regression (LR) to predict stroke using this data [6]. In this research, SMOTE was used to balance all datasets. This was also done by other researchers who used SMOTE and several machine learning methods [7]–[10]. Then, in his research, Gupta evaluated several machine learning methods and found that ensemble learning methods, including AdaBoost, XGBoost, and Random Forest, had better performance compared to other machine learning methods. The performance measurements were carried out using accuracy metrics, while the data used included imbalance data because the amount of data included in the stroke class was much less than data in the normal class. In his research, Sailasya employed several machine learning methods to predict stroke, such as Decision Tree Classification, Logistic Regression, Support Vector Machine, Random Forest, K-Nearest Neighbors, and Naïve Bayes. [11]. After completing the training process, they utilize the Flask framework to accurately predict strokes. The research also used undersampling on the entire dataset before the dataset was divided into training and testing datasets. Rahim uses Extreme Gradient Boosting to predict strokes with the Kaggle data [12]. They discovered that the recall value for the stroke class achieved was still very low, which could be caused by unbalanced data. Then Dev analyzes the important attributes contained in the dataset using Learning Vector Quantization [13]. Based on this research, the attributes Age, Heart Disease, Glucose Level, and Hypertension are important attributes that cause stroke.

Based on previous research, this research proposes data sampling analysis to balance imbalance data, feature selection and extraction to analyze important features in predicting stroke attacks using various machine learning methods. The first contribution proposed is data sampling analysis which includes random undersampling, random oversampling, and SMOTE methods to balance the stroke dataset, where the amount of data in the stroke class is much less than the normal class. Learning patterns in minority data becomes challenging when there is an imbalance in the amount of data available. The data sampling process is only carried out on training data so that its effect on stroke prediction can be seen using the testing data. The second contribution is feature selection and extraction analysis for dimensional reduction and analyzing features that are very influential on stroke prediction. The methods used are Pearson Correlation for feature selection and Principal Component Analysis (PCA) for feature extraction. Then in this research, a wide range of machine learning methods were used to predict stroke occurrence including Adaptive Boosting (AdaBoost), Decision Tree (DT), Extreme Gradient Boosting (XGBoost), Gaussian Naïve Bayes, k Nearest Neighbors (k-NN), Neural Network, Quadratic Discriminant Analysis (QDA), Random Forest (RF), and Support Vector Machine (SVM). Machine learning is an essential part of Artificial Intelligence in which it can effectively recognize patterns within training data to make accurate predictions for new data.

2. METHOD

This research proposes an analysis of the influence of data sampling and feature selection and extraction on stroke prediction using machine learning methods. The dataset used is the stroke prediction dataset which is one of the Kaggle datasets [5]. The dataset consists of ten attributes and has two class labels, where the amount of data in the normal class is much greater than the stroke class. Therefore, a method is needed to handle data imbalance in training data, which includes random undersampling, random oversampling, and SMOTE. Then, Pearson Correlation Analysis used to find out the attributes that have highest correlation with the target attribute (stroke) and select the correlated attributes. PCA is also used to obtain principal components from training data and is used to reduce the dimensions of the available dataset. Next, several machine learning methods were used to train and predict stroke on testing data. The machine learning methods used include AdaBoost, Decision Tree, XGBoost, Gaussian Naïve Bayes, k-NN, Neural Network, QDA, Random Forest, and SVM. Figure 1 shows the proposed methodology for predicting stroke.

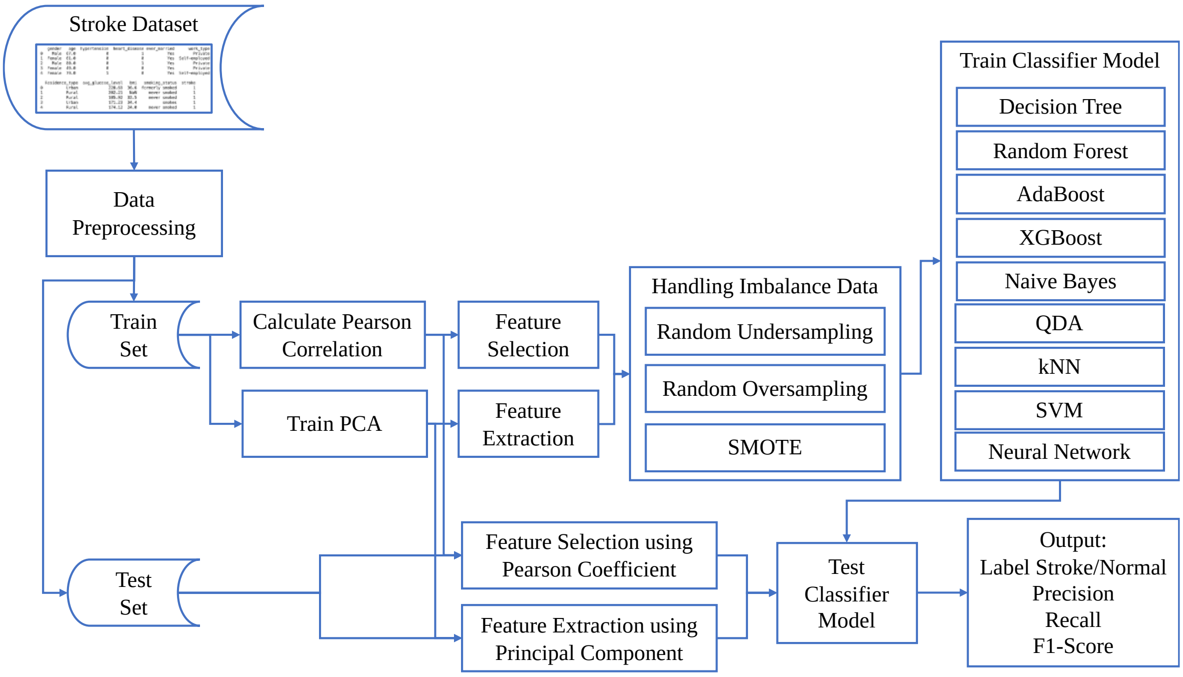


Figure 1. Proposed Methodology for Stroke Prediction.

2.1. *Preprocessing and Exploratory Data*

The stroke prediction dataset used consists of ten input attributes and one output or target in the form of stroke (1) and normal (0) classes. Available input attributes include smoking status, body mass index (BMI), blood glucose level, residence type, work type, marital status, heart disease, hypertension, age, and gender, where the smoking status, residence type, work type, marital status, and gender attributes are string types, and the attributes BMI, blood glucose level, heart disease, hypertension, and age are numeric types. Attributes of the string type will first be processed to become numeric so they can be processed in the machine learning model. The results of the preprocessing process are presented in a histogram in Figure 2 to see the data distribution for each attribute. In the histogram, the amount of data in the stroke class is much less, namely 249 records, while the data in the normal class is 4861 records. In this dataset there are missing values in the bmi attribute and the records that have missing values will be deleted, so that the total data in the stroke class becomes 209 records and the normal class becomes 4700 records. Next, the dataset is divided into two parts, namely training and testing data. Training data is used when train and build the machine learning models, and testing data is used to test previously trained models and obtain the performance of each model. In this research, testing data was selected randomly from the total data, namely 50 data with the stroke label and 50 data with the normal label. The training data amounted to 4809 data with 159 data labeled stroke and 4650 data labeled normal. The data in the training data does not overlap with the testing data, and vice versa.

2.2. *Feature Selection and Extraction*

In the next stage, the Pearson Correlation Coefficient (r) and Principal Component (PC) are calculated using training data which has been previously separated from testing data. Pearson Correlation is used to analyze the linear correlation of two variables. Pearson Correlation Coefficient will range in the range -1 to 1. An r value in the range (0, 1] means that the two variables are positively correlated, where if the value of one variable increases, then the value of the other variable will also increase. Conversely, if the value of r is in the range [-1, 0) means that the two variables are negatively correlated, where if the value of one variable increases, the value of the other variable will decrease.

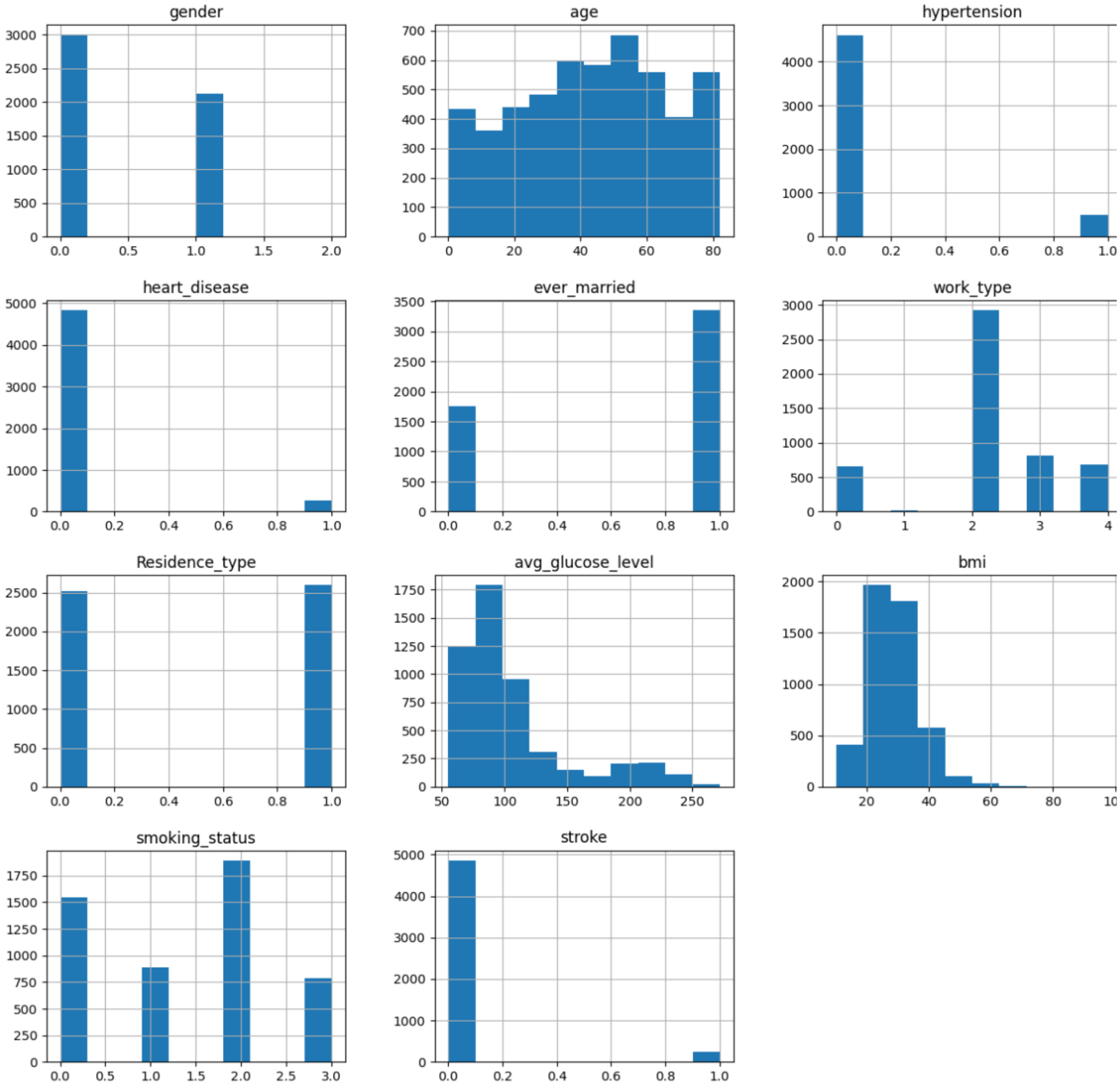


Figure 2. Histogram of each attribute in the stroke prediction dataset.

Meanwhile, if the r value is 0, then the two variables are not correlated. Pearson Correlation Coefficient can be calculated using Equation 1, where r is the Pearson Correlation Coefficient, N is the number of pairs of variables whose correlation is calculated, x is the first variable, and y is the second variable [14], [15]. In this research, each attribute in the training dataset is calculated for its correlation with the Stroke target, so that there will be ten Pearson Correlation Coefficients from this calculation (Figure 3). This Pearson Coefficient value will later be used to select the attributes that are most influential in predicting stroke.

$$r = \frac{N\sum xy - (\sum x)(\sum y)}{\sqrt{(N\sum x^2 - (\sum x)^2)(N\sum y^2 - (\sum y)^2)}} \quad (1)$$

Apart from feature selection, this research also uses feature extraction which will later be compared and analyzed. Feature extraction and feature selection can both be used for dimensionality reduction but in different ways. Feature selection selects attributes that are relevant or have a strong correlation with the target output without changing the value of the selected attributes. Meanwhile, feature extraction reduces dimensions by transforming data into smaller dimensions and extracting important features from a collection of available attributes. The features resulting from the extraction process will be different from the original attribute values. PCA is a feature extraction method that is often used to reduce dimensions [16], [17]. The first step in PCA is to standardize the values of each attribute so that all attributes will have the same value range. After the standardization process, the covariance matrix is calculated to find out whether there is a correlation between the attributes in the dataset. Highly correlated attributes also have the potential to have redundant values. Therefore, it is necessary to calculate the covariance matrix. The third step is to calculate the eigenvector and eigenvalue from the covariance matrix that has been calculated previously to determine the principal component of the available data. Principal components are obtained by sorting the eigenvectors in descending order based on their eigenvalues. Principal components are new features formed from a linear combination of initial attributes. The number of principal components obtained is the same as the number of attributes available, but PCA will maximize important information in the first principal component. So, not much information is lost when dimension reduction is carried out. Dimensionality reduction can be done by selecting several initial vectors from the principal components and multiplying them by an initial standardized dataset.

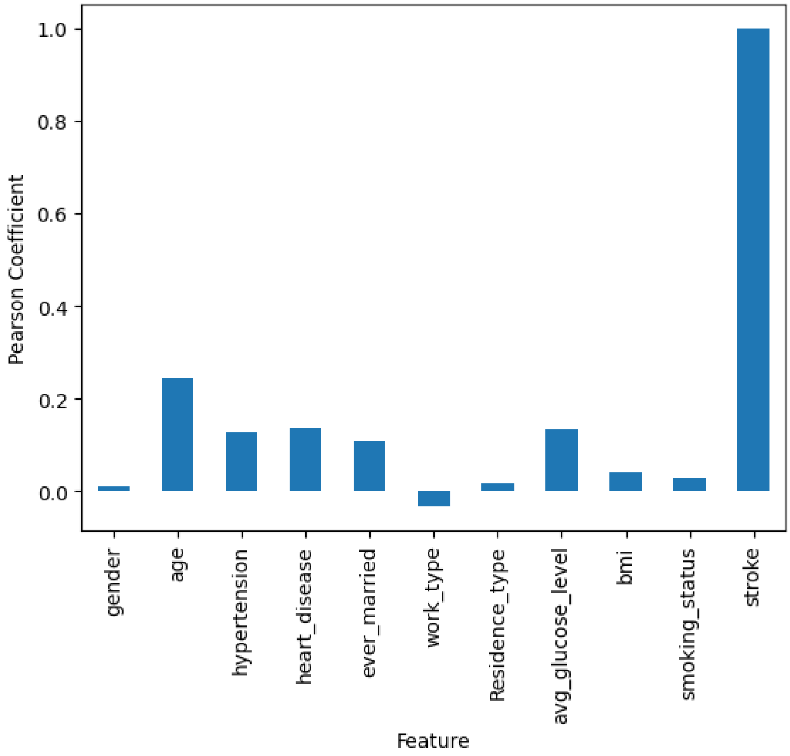


Figure 3. Pearson Correlation Coefficient of each attribute in the stroke prediction dataset.

2.3. *Data Sampling Technique*

The stroke prediction dataset used has much less data on the stroke label than the normal label. This condition will affect the training process of the machine learning model where the machine learning model is more inclined to learn patterns in the normal class. In this research, several data sampling techniques were used to balance the amount of data in the stroke and normal classes in the training data. The sampling techniques used include random undersampling, random oversampling, and Synthetic Minority Oversampling Technique (SMOTE). In the random undersampling technique, data in the majority class is reduced by selecting randomly so that the amount is the same as data in the minority class [18], [19]. Meanwhile, in the random oversampling technique, data in the minority class is duplicated by selecting data randomly so that the amount is the same as data in the majority class [20]. Then, SMOTE is an oversampling technique that generates synthetic data based on differences in randomly selected minority data and its k nearest neighbors [21], [22]. The first step in data sampling in SMOTE is to select randomly one data item in the minority class. Select k-nearest neighbors from the data and calculate the difference, and multiply the difference obtained by a number in the range 0 to 1. The result of the difference between the minority data and its nearest neighbors multiplied by a number in the range 0 to 1 will become synthetic data. Repeat these steps until the amount of minority data is the same as the amount of data in the majority class.

2.4. Model Classifier

After data preprocessing, feature selection or extraction, and data sampling, the next stage is predicting stroke attacks using various machine learning models. Before being used, this model is trained using previously prepared training data. Machine learning methods used include AdaBoost, Decision Tree, XGBoost, Gaussian Naïve Bayes, k-NN, Neural Network, QDA, Random Forest, and SVM. Several machine learning models were selected from various types including tree models, ensemble learning, probabilistic, nearest neighbor, and hyperplane based, to analyze the differences in behavior of several types of machine learning models on the preprocessing and sampling data that has been carried out.

Decision Tree is a machine learning model in the form of a tree. There are several ways to form a decision tree, one of which is by calculating entropy and information gain to select the attribute that is the best split [23], [24]. Entropy indicates the level of heterogeneity of a data set, where the more heterogeneous the data, the higher the entropy. Then Information Gain is a metric used to measure the effectiveness of an attribute in classifying data, which is calculated from the entropy of the entire dataset minus the entropy of each value on an attribute. The attribute that has the highest information gain will be the best split attribute. Entropy and information gain can be calculated using equations 2 and 3, where c is the number of classes, p_i is the sample proportion for class i, A is an attribute, v is a value contained in attribute A.

$$Entropy(S) = \sum_{i-1}^{c} -p_i log_2 p_i \quad (2)$$

$$Information\ Gain(G, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (3)$$

Random Forest, AdaBoost, and XGBoost are machine learning models that are included in the ensemble learning model. Random Forest is a collection of decision trees [25], where each decision tree is built using training data which is sampled using a sampling technique with replacement, namely Bootstrap Aggregating (Boosting). Then, determining the data class in Random Forest is done based on the voting results from the class output of each decision tree. Different from Random Forest, the training process of Adaboost is carried out sequentially to form strong learners from week learners [26], [27]. At the start, each data in the training data is given the same weight to be sampled using the Boosting technique. After the first training process is complete, the model formed is validated against the entire training data. The results of the validation will determine the weight of the training data, where the weight of incorrectly classified data will be increased, while the weight of data that is classified correctly will have its weight reduced. This training process will be repeated until an optimal model is obtained. XGBoost is an end-to-end tree boosting system which is a gradient boosting model. XGBoost uses regularization techniques to overcome overfitting, handles sparse data, and can perform parallel learning using a block structure [28].

Gaussian Naïve Bayes (GNB) is a probabilistic machine learning that applies Bayes' theorem. In continuous data, GNB determines the probability of data falls into a certain class based on the normal distribution and calculates the mean and standard deviation for each attribute in the dataset. This probability can be calculated using equation 4 [29], where $P(x_i|c)$ is the probability that data x_i falls into class c, μ and σ are the average and standard deviation of an attribute. Furthermore, k-nearest neighbors (k-NN) is a supervised learning method where determining the class of data is obtained based on labels from data that have the closest distance to the data in the training data [30]. Distance calculations can be done using Euclidian, Manhattan, or other distance calculation techniques. In k-NN the training process is only carried out to store all training data into certain variables, and distance calculations are carried out during validation and testing.

$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}} \quad (4)$$

Support Vector Machine (SVM) uses hyperplanes to separate data between classes [31], [32]. The hyperplane for linear classification is presented in equation 5. The goal of SVM is to find the optimal hyperplane by maximizing the margin in the training data. The larger the margin produced, the better the hyperplane separates data between classes in the training data and will have better generalization capabilities for classifying testing data. This goal can be achieved by solving equation 6. This equation is a dual form problem and can be solved using Lagrange Multipliers and QP solver, so that we get a hyperplane in equations 7 and 8, where y is the target data, x is the input data, w is the weight or coefficient for each attribute, b is the bias term, and a is the Lagrange multipliers.

$$y_i = w^T x_i + b \quad (5)$$

$$\max_{a} \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i\, y_j x_i . x_j \quad (6)$$

constrain to $a_i \geq 0$ and $\sum_{i=1}^{n} a_i y_i = 0$

$$w = \sum_{i=1}^{n} a_i y_i \, x_i \tag{7}$$

$$b = y_i - w^T x_i \tag{8}$$

Same as SVM, Neural Network also uses hyperplanes to separate data between classes. The hyperplane is obtained from the weights and biases in the Neural Network model. In this research, a Multi-Layer Perceptron is used, where the Neural Network model will have a hidden layer [33], [34]. The number of input neurons in a Neural Network will amount to the data attributes used, and the number of output neurons will also adjust to the number of classes contained in the data. Then, the training process on the Neural Network is carried out using the Backpropagation algorithm which has three phases, namely the forward phase, backward phase, and weight update phase. In its development, several optimization methods have been developed to speed up training process, one of which is the Adam optimization algorithm [35]. Quadratic Discriminant Analysis (QDA) is a discriminant analysis method which aims to find quadratic hypersurfaces to separate classes between data [36]. QDA can handle heteroscedasticity data, namely data that can have heterogeneous variance values. The initial step in the QDA algorithm is calculating the mean and covariance matrix values for each class. Next, calculate the QDA discriminant function (equation 9), so that data can be classified using equation 10 [37], where $\delta(x)$ is the QDA discriminant, x is the input data, $\sum_1$ and $\sum_2$ are the covariance matrices for class 1 and class 2, μ_1 and μ_2 are the average values for classes 1 and 2, and π_1 and π_2 are the prior probabilities for classes 1 and 2.

$$\delta(x) = x^T(\textstyle\sum_1 - \sum_2)^{-1}x + 2(\sum_2^{-1}\mu_2 - \sum_1^{-1}\mu_1)^T x + ({\mu_1}^T \sum_1^{-1}\mu_1 - {\mu_2}^T \sum_2^{-1}\mu_2) + ln\left(\frac{\sum_1}{\sum_2}\right) + 2\ln\left(\frac{\pi_2}{\pi_1}\right) \tag{9}$$

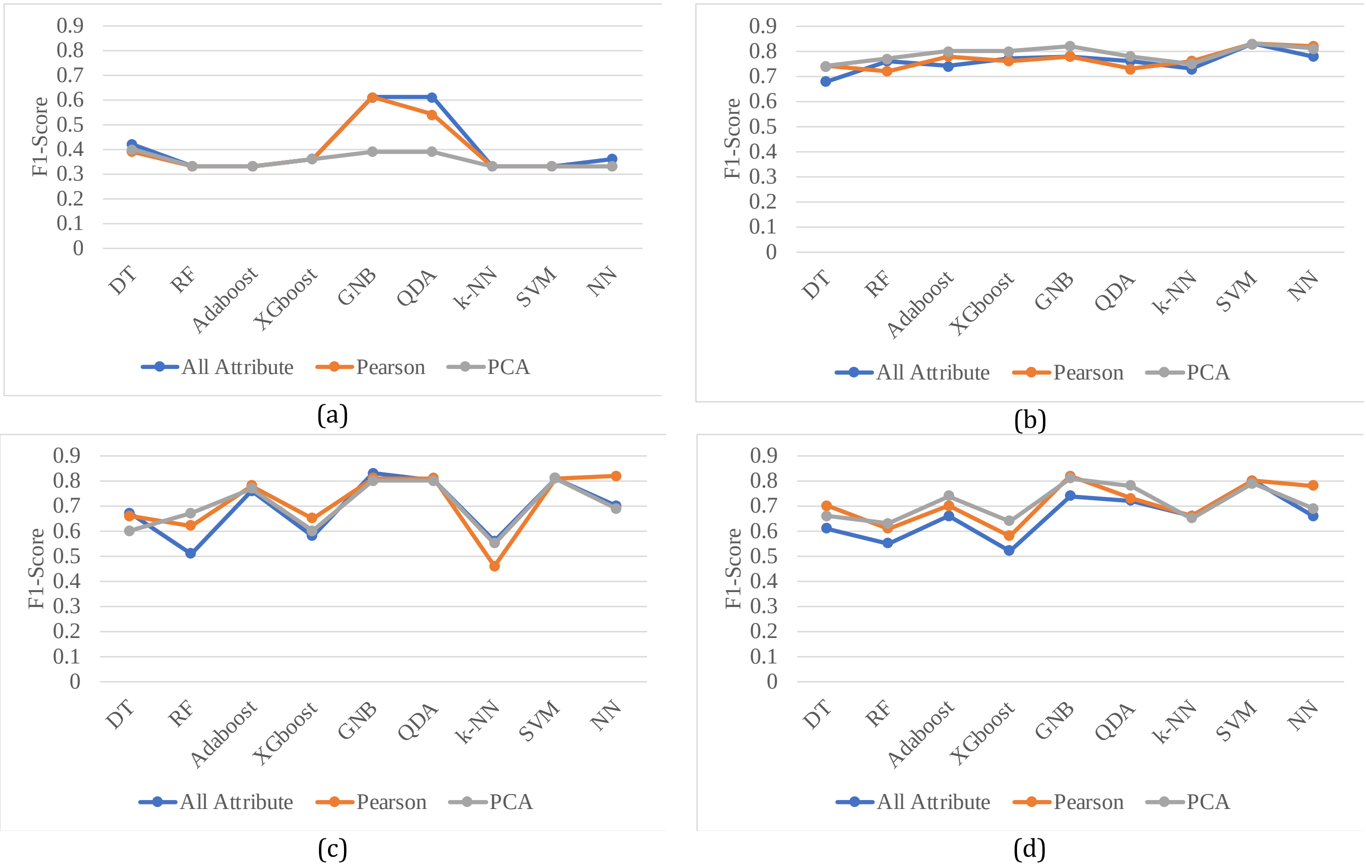
$$c(x) = \begin{cases} 1, & \delta(x) < 0 \\ 2, & \delta(x) > 0 \end{cases} \tag{10}$$

3. RESULT AND DISCUSSION

This research aims to analyze the effect of feature extraction, feature selection, and data sampling on stroke prediction using various machine learning models. The feature extraction method used is PCA, while for feature selection used is the Pearson correlation which calculated the correlation between each input attribute and the target stroke. Then the sampling techniques used include Random Undersampling, Random Oversampling, and SMOTE, as well as the machine learning models used include AdaBoost, Decision Tree, XGBoost, Gaussian Naïve Bayes, k-NN, Neural Network, QDA, Random Forest, and SVM. The hyperparameters in the machine learning model use default hyperparameter values, including the Decision Tree model using a maximum depth of 10, Random Forest using a maximum depth of 10 and number estimators of 100, AdaBoost using number estimators of 100, XGBoost using a learning rate of 0.08, k-NN with 5 nearest neighbor, SVM with a Radial Basis Function (RBF) kernel, and Neural Network using 1 hidden layer with 100 neurons.

3.1. Feature Selection and Extraction Observation

The first observation is to analyze the effect of feature extraction and selection on stroke prediction results using various machine learning models. The testing scheme carried out is by comparing the use of PCA as a feature extraction method, Pearson Correlation for feature selection, and using all attributes to predict stroke, with four variations of training data for the training process, namely overall training data without data sampling and training data resulting from sampling data using RUS, ROS and SMOTE. When using feature selection with Pearson Correlation, five attributes were selected that had the highest Pearson coefficient to the target data (stroke), namely age, hypertension, heart disease, blood sugar level, and marital status attribute. Therefore, in feature extraction using PCA, five



(a)

(b)

(c)

(d)

Figure 4. Testing F1-score comparison of feature extraction and selection on stroke prediction using machine learning models with training data: (a) without data sampling, (b) with RUS, (c) with ROS, and (d) with SMOTE.

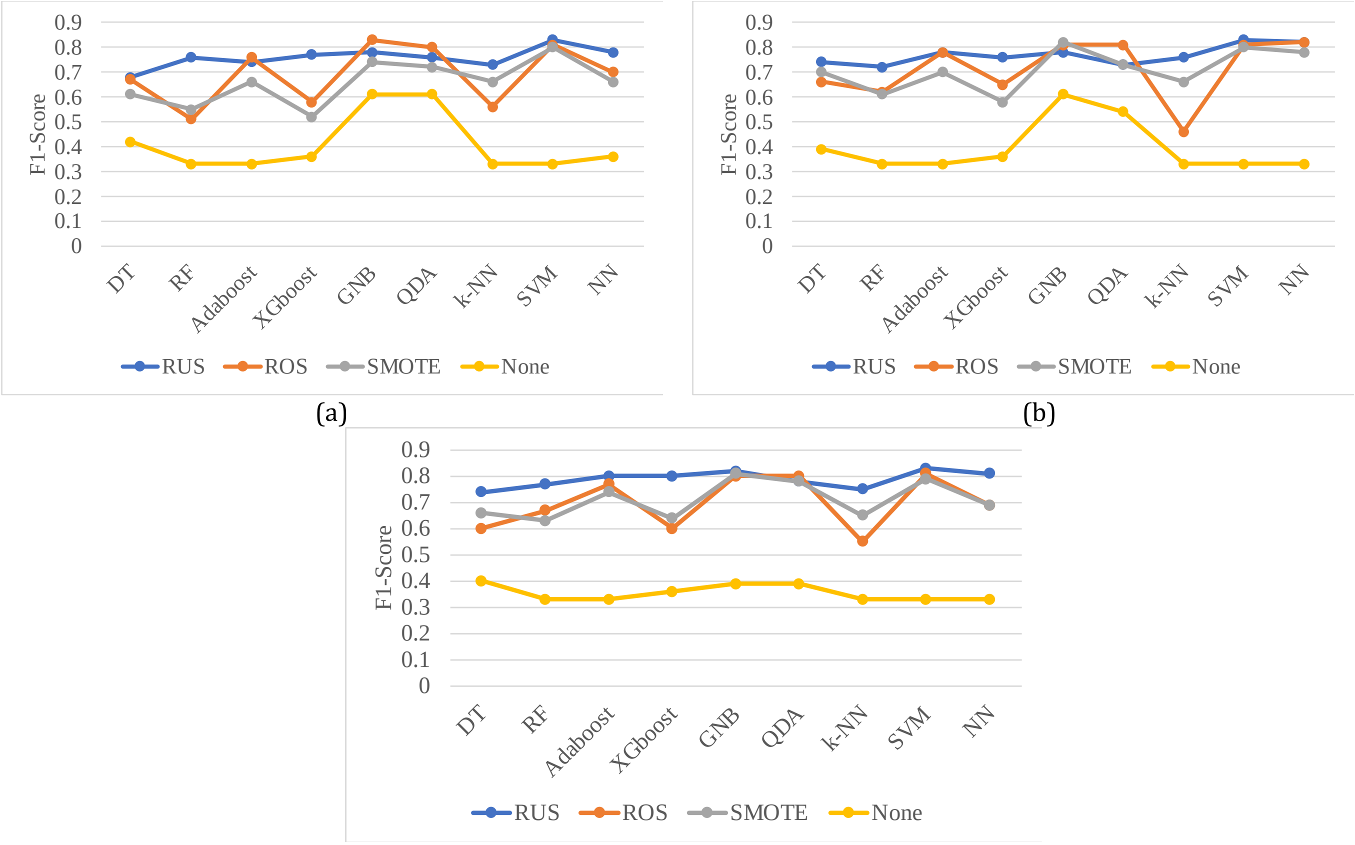
principal components are also taken so that the data resulting from dimensional reduction using PCA also produces five new attributes.

Figure 4 illustrates the results of observations and testing performance of feature extraction and selection for stroke prediction, where Figure 4 (a) uses training data without data sampling, Figure 4 (b) uses training data with RUS, Figure 4 (c) uses training data with ROS, and Figure 4 (c) uses training data with SMOTE. In the test results with training data from RUS, ROS, and SMOTE, the use of feature extraction and selection does not have a significant effect on stroke prediction using machine learning models. In Figure 4(a), applying feature selection with Pearson Correlation leads to a decrease of 0.0144 in the average F1-Score during testing. Similarly, utilizing PCA feature extraction results in a reduction of 0.0544 in the average F-Score value during testing. Figure 4(b) reveals that incorporating feature selection enhances the average F1-Score testing value by a marginal increase of 0.01. However, choosing feature extraction leads to a more substantial improvement, boosting the average F-Score testing value by 0.03. In Figure 4 (c), incorporating feature selection results in a mere 0.0222 increase in the average F1-Score testing value, while the utilization of feature extraction yields a slight 0.0077 boost in the average F-Score testing value. In Figure 4 (d), the average F1-Score testing value increases by 0.051 with feature selection. Additionally, the use of feature extraction further enhances the average F-Score testing value by 0.0522.

3.2. Data Sampling Observation

The second test carried out was observing the effect of sampling data on stroke prediction using a machine learning model. The testing scheme carried out is by comparing the testing performance of machine learning models resulting from training using all training data and training data resulting from RUS, ROS, SMOTE data sampling. Observations of the effect of using data sampling were applied to three types of datasets, namely training data using all attributes, training data with five attributes resulting from feature selection, and training data using five attributes resulting from feature extraction. The total training data amounted to 4809 data with 159 stroke labelled data and 4650 normal labelled data. Then the training data was sampled using RUS, ROS, and SMOTE. In the RUS training data, the resulting data is 159 stroke labelled data and 159 normal labelled data, while in the ROS and SMOTE training data the resulting data is 4650 stroke labelled data and 4650 normal labelled data.

Figure 5 presents the performance comparison results of data sampling on stroke prediction using various machine learning models. Based on Figures 5 (a), 5 (b), and 5 (c), the use of data sampling provides a significant increase in F1-Score testing compared to the performance of models trained with training data without data sampling. This is because the stroke dataset used is very imbalanced so that if training is carried out on imbalanced training data, the machine learning model will more focus on studying data patterns in majority classes. Meanwhile, when the training data is sampled, the amount of data in each class becomes balanced and the machine learning model can learn data patterns in each class. Then, based on the test results, it was also found that the use of the RUS sampling technique could provide a more stable performance increase for all machine learning models used, especially on datasets with five attributes resulting from feature extraction. In Figure 5 (a), namely a dataset with all attributes, the use of the RUS sampling technique can increase the average F1-Score testing by 0.35, ROS increases the average F1-score testing by 0.2822, and SMOTE increases the average F1-score testing of 0.2488 when compared to training data without sampling. In Figure 5 (b), namely a dataset with five attributes resulting from feature selection, the use of RUS, ROS, and SMOTE sampling techniques can increase the average F1-Score testing by 0.3744, 0.3188, and 0.3144. Meanwhile, in the dataset resulting from feature



(a)

(b)

(c)

Figure 5. F1-Score testing comparison of data sampling on stroke prediction using various machine learning model with a dataset using: (a) all attributes, (b) five attributes resulting from feature selection, and (c) five attributes resulting from feature extraction.

extraction (Figure 5 (c)), the use of RUS, ROS, and SMOTE sampling techniques can increase the average F1-Score testing by 0.4344, 0.3444, and 0.3555.

3.3. Machine Learning Model Performance

The final observation involves comparing machine learning models for predicting strokes. The machine learning models used include AdaBoost, Decision Tree, GNB, k-NN, Neural Network, QDA, Random Forest, SVM, and XGBoost. Figure 6 presents a comparison of F1-Score for training and testing data of each machine learning model used on the dataset using all attributes. In Figure 6, it can be seen that the Decision Tree, XGBoost, Random Forest, and k-NN have a very high training F1-score (above 0.9) but the testing F1-Score obtained is moderate. This shows that these models experience overfit, where the model is very good at predicting training data but has low generalization ability to new data. In contrast, GNB and SVM have relatively high and stable performance for predicting stroke. Next, Table 1 presents the best F1-Score testing results for each machine learning model along with the optimal feature analysis and data sampling techniques for each classifier model. SVM has the highest F1-Score testing, namely 0.83, using all attributes or five selected attributes from the Pearson Correlation, as well as the RUS sampling technique. GNB also has the highest F1-Score testing, namely 0.83 using all attributes and ROS sampling techniques.

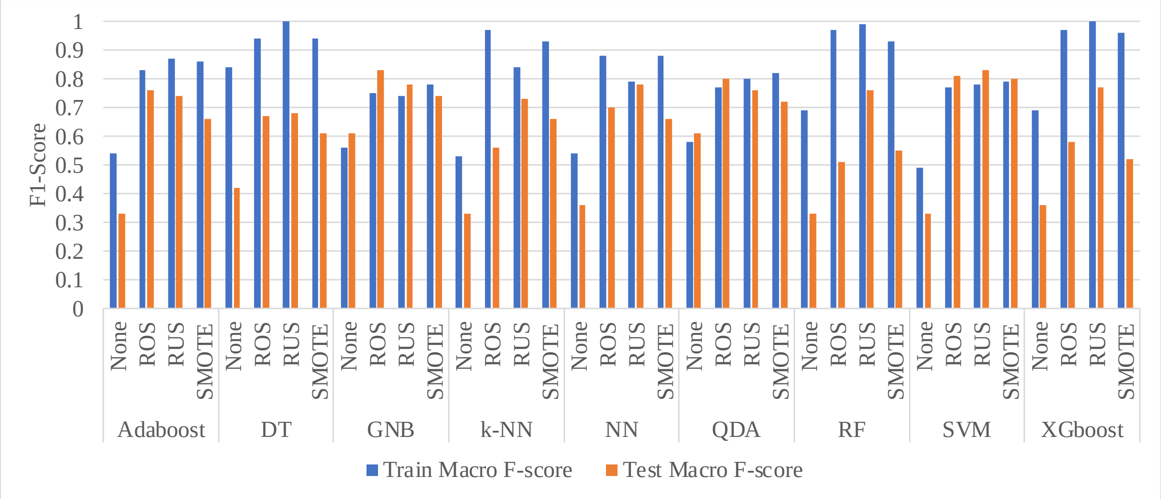


Figure 6. Comparison of training and testing F1-scores of each machine learning model on the dataset with all attributes.

Table 1. The best F1-Score testing for each machine learning model.

Classifier	Feature Analysis	Sampling Data	Test F1-Score
Adaboost	PCA	RUS	0.8
DT	PCA	RUS	0.74
GNB	All	ROS	**0.83**
k-NN	Pearson	RUS	0.76
NN	Pearson	ROS/RUS	0.82
QDA	Pearson	ROS	0.81
RF	PCA	RUS	0.77
SVM	All Attribute/Pearson	RUS	**0.83**
XGBoost	PCA	RUS	0.8

4. CONCLUSION

This research employed various machine learning models, which consists of AdaBoost, Decision Tree, XGBoost, Gaussian Naïve Bayes, k-NN, Neural Network, QDA, Random Forest, and SVM, to predict stroke using the stroke dataset obtained from Kaggle. This data has ten input attributes which are clinical symptoms of stroke which can be used to diagnose stroke. Then the data has a much smaller amount of data with stroke labels than normal labels. This condition is called data imbalance which can be a problem when using machine learning models. Therefore, in this research, Random Undersampling, Random Oversampling, and SMOTE data sampling techniques were used to handle the data imbalance problem. According to the conducted experiments, it has been discovered that the sampling technique utilized can significantly enhance the F1-Score testing. Specifically, the average testing F1-Score can be increased by 0.35, 0.2822, and 0.2488 using the RUS, ROS, and SMOTE sampling techniques, respectively. Then, Pearson Correlation and Principal Component Analysis techniques were also used to analyze important features of the available stroke dataset. Furthermore, the highest testing F1-Score was achieved using the Support Vector Machine model of 0.83 using all attributes or Pearson Correlation feature selection with the RUS technique. Also, the Gaussian Naïve Bayes model also has the highest F1-Score testing, namely 0.83 using all attributes and ROS techniques.

ACKNOWLEDGEMENTS

This research was supported by Telkom University.

REFERENCES

[1] B. W. Negasa, T. W. Wotale, M. E. Lelisho, L. K. Debusho, K. Sisay, and W. Gezimu, “Modeling Survival Time to Death among Stroke Patients at Jimma University Medical Center, Southwest Ethiopia: A Retrospective Cohort Study,” *Stroke Res. Treat.*, vol. 2023, pp. 1–10, Nov. 2023, doi: 10.1155/2023/1557133.

[2] “Acute Ischemic Stroke: Management Approach,” *Indian J. Crit. Care Med.*, vol. 23, no. S2, pp. 140–146, Jun. 2019, doi: 10.5005/jp-journals-10071-23192.

[3] D. Kuriakose and Z. Xiao, “Pathophysiology and Treatment of Stroke: Present Status and Future Perspectives,” *Int. J. Mol. Sci.*, vol. 21, no. 20, p. 7609, Oct. 2020, doi: 10.3390/ijms21207609.

[4] G. Fekadu, L. Chelkeba, and A. Kebede, “Risk factors, clinical presentations and predictors of stroke among adult patients admitted to stroke unit of Jimma university medical center, south west Ethiopia: prospective observational study,” *BMC Neurol.*, vol. 19, no. 1, p. 187, Dec. 2019, doi: 10.1186/s12883-019-1409-0.

[5] fedesoriano, “Stroke Prediction Dataset.” 2020. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset/data

[6] M. Guhdar, A. Ismail Melhum, and A. Luqman Ibrahim, “Optimizing Accuracy of Stroke Prediction Using Logistic Regression,” *J. Technol. Inform. JoTI*, vol. 4, no. 2, pp. 41–47, Jan. 2023, doi: 10.37802/joti.v4i2.278.

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

[8] Md. M. Islam, S. Akter, Md. Rokunojjaman, J. H. Rony, A. Amin, and S. Kar, “Stroke Prediction Analysis using Machine Learning Classifiers and Feature Technique,” *Int. J. Electron. Commun. Syst.*, vol. 1, no. 2, pp. 57–62, Dec. 2021, doi: 10.24042/ijecs.v1i2.10393.

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

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

[11] 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, doi: 10.14569/IJACSA.2021.0120662.

[12] A. M. A. Rahim, A. Sunyoto, and M. R. Arief, “Stroke Prediction Using Machine Learning Method with Extreme Gradient Boosting Algorithm,” *MATRIK J. Manaj. Tek. Inform. Dan Rekayasa Komput.*, vol. 21, no. 3, pp. 595–606, Jul. 2022, doi: 10.30812/matrik.v21i3.1666.

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

[14] F. Zinzendoff Okwonu, B. Laro Asaju, and F. Irimisose Arunaye, “Breakdown Analysis of Pearson Correlation Coefficient and Robust Correlation Methods,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 917, no. 1, p. 012065, Sep. 2020, doi: 10.1088/1757-899X/917/1/012065.

[15] E. I. Obilor and E. C. Amadi, “Test for Significance of Pearson’s Correlation Coefficient (r),” *Int. J. Innov. Math. Stat. Energy Policies*, vol. 6, no. 1, pp. 11–23, 2018.

[16] E. Elhaik, “Principal Component Analyses (PCA)-based findings in population genetic studies are highly biased and must be reevaluated,” *Sci. Rep.*, vol. 12, no. 1, p. 14683, Aug. 2022, doi: 10.1038/s41598-022-14395-4.

[17] L. Peng, G. Han, A. Landjobo Pagou, and J. Shu, “Electric submersible pump broken shaft fault diagnosis based on principal component analysis,” *J. Pet. Sci. Eng.*, vol. 191, p. 107154, Aug. 2020, doi: 10.1016/j.petrol.2020.107154.

[18] M. Saripuddin, A. Suliman, S. Syarmila Sameon, and B. N. Jorgensen, “Random Undersampling on Imbalance Time Series Data for Anomaly Detection,” in *2021 The 4th International Conference on Machine Learning and Machine Intelligence*, Hangzhou China: ACM, Sep. 2021, pp. 151–156. doi: 10.1145/3490725.3490748.

[19] M. Bach, A. Werner, and M. Palt, “The Proposal of Undersampling Method for Learning from Imbalanced Datasets,” *Procedia Comput. Sci.*, vol. 159, pp. 125–134, 2019, doi: 10.1016/j.procs.2019.09.167.

[20] R. G, A. K. Tyagi, and V. K. Reddy, “Performance Analysis of Under-Sampling and Over-Sampling Techniques for Solving Class Imbalance Problem,” *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3356374.

[21] D. Elreedy and A. F. Atiya, “A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance,” *Inf. Sci.*, vol. 505, pp. 32–64, Dec. 2019, doi: 10.1016/j.ins.2019.07.070.

[22] B. S. Raghuwanshi and S. Shukla, “SMOTE based class-specific extreme learning machine for imbalanced learning,” *Knowl.-Based Syst.*, vol. 187, p. 104814, Jan. 2020, doi: 10.1016/j.knosys.2019.06.022.

[23] I. D. Mienye, Y. Sun, and Z. Wang, "Prediction performance of improved decision tree-based algorithms: a review," *Procedia Manuf.*, vol. 35, pp. 698–703, 2019, doi: 10.1016/j.promfg.2019.06.011.

[24] C. Zhang, C. Hu, S. Xie, and S. Cao, "Research on the application of Decision Tree and Random Forest Algorithm in the main transformer fault evaluation," *J. Phys. Conf. Ser.*, vol. 1732, no. 1, p. 012086, Jan. 2021, doi: 10.1088/1742-6596/1732/1/012086.

[25] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata J. Promot. Commun. Stat. Stata*, vol. 20, no. 1, pp. 3–29, Mar. 2020, doi: 10.1177/1536867X20909688.

[26] Y. Ding, H. Zhu, R. Chen, and R. Li, "An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification," *Appl. Sci.*, vol. 12, no. 12, p. 5872, Jun. 2022, doi: 10.3390/app12125872.

[27] Y. Zhang *et al.*, "Research and Application of AdaBoost Algorithm Based on SVM," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China: IEEE, May 2019, pp. 662–666. doi: 10.1109/ITAIC.2019.8785556.

[28] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[29] J. Shen and H. Fang, "Human Activity Recognition Using Gaussian Naïve Bayes Algorithm in Smart Home," *J. Phys. Conf. Ser.*, vol. 1631, no. 1, p. 012059, Sep. 2020, doi: 10.1088/1742-6596/1631/1/012059.

[30] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci. Rep.*, vol. 12, no. 1, p. 6256, Apr. 2022, doi: 10.1038/s41598-022-10358-x.

[31] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.

[32] B. Gaye, D. Zhang, and A. Wulamu, "Improvement of Support Vector Machine Algorithm in Big Data Background," *Math. Probl. Eng.*, vol. 2021, pp. 1–9, Jun. 2021, doi: 10.1155/2021/5594899.

[33] J. Singh and R. Banerjee, "A Study on Single and Multi-layer Perceptron Neural Network," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India: IEEE, Mar. 2019, pp. 35–40. doi: 10.1109/ICCMC.2019.8819775.

[34] H. Alla, L. Moumoun, and Y. Balouki, "A Multilayer Perceptron Neural Network with Selective-Data Training for Flight Arrival Delay Prediction," *Sci. Program.*, vol. 2021, pp. 1–12, Jun. 2021, doi: 10.1155/2021/5558918.

[35] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014, doi: 10.48550/ARXIV.1412.6980.

[36] R. Wu and N. Hao, "Quadratic discriminant analysis by projection," *J. Multivar. Anal.*, vol. 190, p. 104987, Jul. 2022, doi: 10.1016/j.jmva.2022.104987.

[37] A. Araveeporn, "Comparing the Linear and Quadratic Discriminant Analysis of Diabetes Disease Classification Based on Data Multicollinearity," *Int. J. Math. Math. Sci.*, vol. 2022, pp. 1–11, Sep. 2022, doi: 10.1155/2022/7829795.