DOI: 10.15849/IJASCA.240330.06

Int. J. Advance Soft Compu. Appl, Vol. 16, No. 1, March 2024
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Forecasting Hospital Length of Stay for Stroke Patients: A Machine Learning Approach

Imam Tahyudin, Siti Alvi Solikhatin, Ades Tikaningsih, Puji Lestari, Hidetaka Nambo, Eko Winarto, Nazwan Hassa

Information System Department of Universitas Amikom Purwokerto
imam.tahyudin@amikompurwokerto.ac.id
Informatic Department of Universitas Amikom Purwokerto
sitialvi@amikompurwokerto.ac.id
Informatic Department of Universitas Amikom Purwokerto
Adestikaningsih92@gmail.com
Information System Department of Universitas Amikom Purwokerto
Puji24318@gmail.com
Graduate School of Electrical Engineering and Computer Science of Kanazawa University, Kakuma Campus, Kanazawa and 920-1192, Japan
nambo@blitz.et.c.kanazawa-u.ac.jp
Research and Development Department of RSUD Banyumas
eko_win77@yahoo.co.id
Research and Development Department of RSUD Banyumas
nazwanhassa@gmail.com

Abstract

A major problem for healthcare systems is stroke, which is one of the main causes of disability and death globally. The accurate estimation of stroke patients' length of stay (LOS) is essential for the effective use of resources and the provision of individualized care. One useful strategy is to make use of machine learning technology. Nonetheless, the selection of a machine learning method is a crucial and little-studied aspect when it comes to the intricacy of stroke, wherein every situation might differ dramatically. This research endeavors to investigate alternative machine learning methodologies to identify the most appropriate model for the heterogeneous attributes of stroke patient data. By contrasting five algorithms XGB, Extra Tree, CatBoost, Decision Tree, and Random Forestwe want to create a prediction model for stroke patients' length of stay within the hospital. The XGBoost model increased its accuracy rate to 82% and its area under the curve (AUC) to 79% by carefully adjusting its hyperparameters. The AUC-ROC score of 0.79 indicates superior performance in determining the length of hospital stay for stroke patients when compared to alternative models. The aforementioned findings suggest that XGBoost the most suitable option for accurately estimating the duration of hospitalization for stroke patients.

Keywords: *Stroke, length of stay.,Machine learning, prediction.*

Received 11 December 2023; Accepted 4 February 2024

1 Introduction

Stroke is a leading cause of mortality and adult disability worldwide [1]. The number of stroke cases and deaths caused by stroke has increased significantly every year from 1990 to 2019 globally. In Indonesia, stroke is the main cause of death and disability, contributing as the main cause of death with the number of fatalities reaching 328.5 thousand people in 2019, equivalent to 21.2% of total deaths [2]. With constant medical advances in the treatment of acute strokes, it is increasingly important to understand the factors that affect recovery and long-term outcomes for stroke survivors. One of the crucial aspects of stroke management is predicting long stays (LOS) in hospitals for stroke patients. Accurate LOS prediction not only helps healthcare providers with efficient resource allocation and patient management but also plays a significant role in optimizing the overall health system. In recent years, Machine Learning (ML) techniques have become increasingly important in business and health care [3]. ML is the key to supporting medical professionals facing the challenges of these innovative scenarios [4]. Today's technology enables insights from unstructured texts that were previously difficult to do and can be applied on a large scale. With the new wealth of knowledge from ML, doctors and administrators have the ability to make timely and relevant decisions related to patient care. By leveraging the huge amount of patient data available in electronic health records (EHRs), machine learning algorithms can identify complex patterns and relationships that may be difficult for human experts to understand [5].

The study aims to utilize machine learning algorithms to predict the length of hospital stays for stroke patients. By exploring a range of clinical, demographic, and medical history variables, the study aims to develop accurate and reliable predictive models. These models have the potential to improve the quality of stroke care by helping healthcare providers estimate the duration of hospitalization for stroke patients, which will ultimately optimize the use of resources, improve patient outcomes, and ultimately create more efficient and effective health systems. In this study, we will discuss the methodologies used, the data sources used, and the predictive models developed to project long stays for stroke patients. The insights gained from this study can make a significant contribution to personalized patient care, enabling healthcare providers to provide tailored intervention and support to stroke survivors, thereby improving their overall quality of life.

2 Related Work

The application of artificial intelligence (AI) in the world of health has made a significant contribution to the understanding and management of health conditions. AI is not just a tool but a crucial instrument in facing complex challenges in the medical field. In the context of predicting the length of stay for stroke patients, recent studies show that AI can improve the precision and accuracy of predictions. Machine learning algorithms such as XGBoost, Random Forest, and neural networks have successfully processed patient data to provide more accurate predictions, helping medical teams in treatment planning and resource allocation. Previous research has demonstrated the effectiveness of machine learning algorithms in predicting LOS in stroke patients. Among them is research on ischemic stroke in China. The algorithm implemented is XGBoost, a machine learning algorithm known for its accuracy and efficiency. The study involved analyzing data from 18,195 ischemic stroke patients treated in large and comprehensive hospitals in China. The XGBoost model in this study achieved an Area Under the Curve (AUC) of 0.92, demonstrating its ability to predict the actual outcome very well. This model is capable of

predicting, with approximately 85% accuracy, whether a patient will be hospitalized for more than one day [6].

Alternatively, another study addressed the application of hyperparameter settings to improve the performance of machine learning models. The focus of this research is on developing a new framework for hyperparameter optimization known as Optuna. Optuna was designed to overcome the limitations of an existing hyperparameter optimization framework. Experimental results show that in the context of classification, hyperparameter optimization can be used to find optimal values for parameters such as tree size, tree depth, and number of trees in a random forest model [7].

Another study aims to predict the duration of hospitalization for stroke patients, taking into account the potential for significant financial impact on patients, families, and medical institutions. The tests were performed using nerve tissue with three different-size sub-sets of attributes. The best results were achieved on subsets with fewer features, resulting in a root mean square error (RMSE) of 5,9451 and a mean absolute error of 4,6354 [8].

Further research was undertaken to develop a predictive model of old stroke surgery (LOS) in patients with acute stroke using a variety of approaches, including conventional regression techniques, artificial intelligence such as interactive decision trees, neural tissues, and model ensembles. Evaluation of models using the Root Absolute Error Index (ASE) showed that neural networking techniques, as a representation of artificial intelligence, showed superior performance compared to other methods. Evaluations showed ASE values of 23.7 for double regression, 23.7 in the interactive decision tree, 22.7 in neural networks, and 22.7 for ensemble techniques. This indicates that the application of artificial intelligence has proven useful in the development of LOS predictive models [9].

On the other hand, a study aimed to evaluate the improved performance of LOS prediction in hospitals using machine learning when considering clinical signs written in text, compared to traditional approaches that only take into account structured information such as age, gender, and the main diagnosis of ICD. Two random forest models were used to predict LOS. The first model included unstructured text extracted from electronic health records (EHRs). A word-embedding algorithm based on UMLS terminology with precise matches limited to patient-relevant affirmative sentences was used to assess EHR data. The second model was mainly based on structured data, namely diagnoses encoded from the International Classification of Disease, 10th Edition (ICD-10) and triase codes (CCMU/EMGEMSA classifications). The model with unstructured data has an accuracy of 75.0%, while the model with structured information has a precision of 74.1%. Both models provide similar predictions in 86.6% of cases. In a secondary analysis focused on intensive care patients, the accuracy of both models is also comparable, at 76.3% vs. 75.0% [10].

Further research was conducted at one of Iran's educational and therapeutic centers on the duration of hospitalization and related factors in stroke patients. In this descriptive-analytical study, using 253 stroke patient data trained using a linear regression model adjusted to predict the duration of hospitalization in SPSS 21, The multivariate regression models showed that non-working subjects compared with self-employed (β=0.74), hemorrhagic strokes compared to ischemic stroke (β = 0.84), strokes with moderate volumes (β=0.61) and large volumes (β=1.22) compared with small volumes, infectious complications, and the presence of a particular doctor have independent and significant correlations with increased durations of hospital treatment [11].

Previous research has discussed the use of various algorithms in predicting the length of stay of stroke patients. However, this research is different from previous studies by focusing on comparing five algorithms at once, namely XGBoost, Random Forest, Decision Tree, Extra Tree, and CatBoost. This approach provides deeper insight into the relative performance of various models. The choice to use multiple algorithms not only enriches the analysis but also reflects the state of the art in efforts to understand and predict stroke patient length of stay (LOS). Furthermore, we comprehensively explored hyperparameter tuning for each algorithm to improve prediction accuracy in the context of predicting length of stay for stroke patients. By tuning hyperparameters for each algorithm, including XGBoost (XGB), RF, DT, Extra Tree, and CatBoost, this research not only implements different algorithms but also seeks to maximize the potential of each algorithm through careful parameter tuning.

3 The Proposed Method

This research adopts the framework of Rui Chen et al. (2023) [6]. Discuss long-standing forecasts of hospitalization of ischemic stroke in China. The stages on this system are as follows in Fig.1

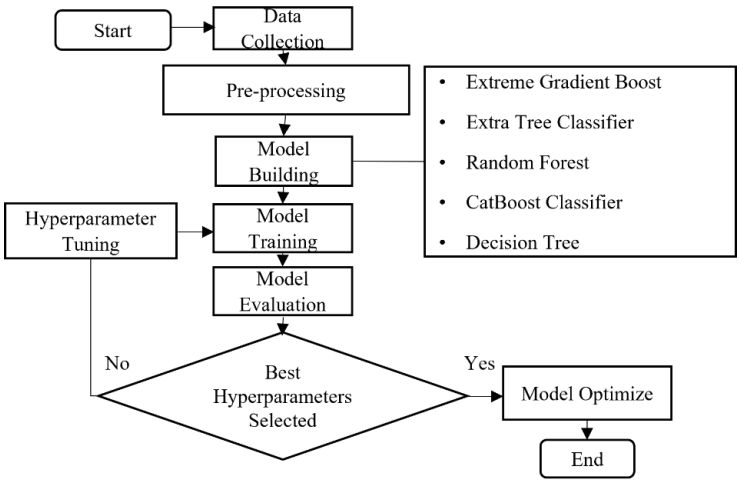

Fig 1. Flowchart

3.1 Data Description

The data used in this study included 106 ischemic stroke patients who were undergoing treatment at the Neural Clinic of the Regional General Hospital (RSUD) in Banyumas, Indonesia. The data collection was carried out by the RSUD research institute team in collaboration with doctors and nurses during the period from January to May 2023. All patients were observed and recorded, including demographic information, medical history, and other clinical records. It is important to note that a detailed description of the variables and the grouping of patients can be found in Table 1 presented in the study. Table 1 not only provides a brief overview of the data but also provides a comprehensive picture of each original attribute present in the patient's data, complete with an in-depth description of each of its attributes.

Table 1. Original stroke dataset

Attribute name	Description
Patient Condition	Patient Health Status
Age	Age of Stroke Patient

Gender	Refers to the category or identity of a person's gender
Debtors	Source or type of health insurance held by the patient
Patient Admission Date	The date when a patient started hospital treatment
Patient discharge Date	The date when a patient finishes treatment and is allowed to leave the hospital
History Of CVD	Information on cardiovascular disease history, such as previous heart disease or stroke
Prior Disease History	Information about other previous medical conditions suffered by the patient
Previous Stroke History	Information on whether the patient has a history of stroke prior to the observed case
Stroke Location	Information about the location of the stroke attack in the patient
HB	Hemoglobin, a component of red blood cells that binds to oxygen
HT	Hematocrit, the proportion of blood volume filled with red blood cells
LEU	Leukocytes, a type of white blood cell
TR	Platelets, a type of blood cell involved in blood clotting
NLR	Neutrophil-Lymphocyte Ratio, the ratio of neutrophils to lymphocytes in the blood
CHOL Total	Total Cholesterol, the aggregate cholestrol level in the bloodstream
HDL	High-Density Lipoprotein, a type of cholesterol known for its role in reducing excess cholesterol in the blood.
TG	Triglycerides, a type of fat in the blood
LDL	Low-Density Lipoprotein, a type of cholesterol considered "bad" because it can clog arteries.

3.2 Proposhed Methodology

3.2.1 Preprocessing

The main focus of this research is to develop a predictive model of the long-term hospitalisation (LoS) of stroke patients as a classification problem [12]. In this case, the main target variable used to train and measure the performance of the model is 'length of stay_inpatient', which is obtained by calculating the difference between admission date and patient discharge. It is important to note that the patient categories that emerge as a result of the variable 'length of stay_inpatient' have a basic categorization into two main groups: first, patients who experience hospitalisation for less than 7 days, and second, patients who experience hospitalisation for more than 7 days. The determination of the 7-day time limit for categorising patients into two groups is based on information from journal literature. There is a concept that the length of stay for stroke patients is considered a multi-classification problem, where the length of stay (LOS) is divided into categories, and one of them is the inpatient category. for 1–7 days [6]. Therefore, we adopted these standards as a basis for dividing patients into these two groups, with the aim of understanding differences in length of stay and categorising patients based on standard and nonstandard conditions. A new variable called the length of hospitalization. This variable has two values: standard and non-standard. In the standard category, there are 54 stroke patient data points, while in the non-standard category, there are 52 stroke patient data points.

Specifically, the proportion of these two values is 54 for stroke patients with standard hospitalization duration (less than 7 days) and 52 for stroke people with unstandard hospital duration. (lebih dari 7 hari). This variable will be the focus of further analysis to evaluate the potential relationship between the duration of hospitalization and other variables that may have an impact on the results of the study.

3.2.2 Handling Missing Data

Filling up missing data is an important challenge in machine learning. One simple and understandable way is to use zero imputation, which means considering the missing value as zero [13]. In this study, some attributes like "Cardiovascular Disease History," "Medical History," and "Previous Stroke History" have lost value. It is important to ensure that the data remains consistent and relevant before carrying out further analysis. Choosing a zero imputation is due to medical interests, especially in predicting hospitalization time for stroke patients. The history of the disease is a crucial factor that affects the outcome of the treatment and the prediction of the patient. Replacing the missing value with 0 allows the separation between patients with a history of illness and those without, providing valuable information for medical professionals who analyze the results of this study. In addition, using 0 as a replacement for missing values also ensures that the data remains consistent in numerical format.

3.2.3 Feature Encoding

At this stage, preprocessing is carried out on categorical attributes of several variables to prepare the dataset to meet the requirements of the predictive model. Encoding categorical variables is an important step in data preparation for predictive models because most machine learning models require data in numerical form [14]. This research applies two encoding methods, namely one-hot encoding and label encoding. One-hot encoding is a technique used to convert categorical variables into a numerical representation. In this process, integer-coded variables are removed, and new binary variables are added for each unique integer value. Each category is represented by a binary variable, with a value of 1 indicating the presence of that category and a value of 0 for other categories [15].

One-hot encoding is applied to the variables length of stay, gender, and patient condition because the values for these variables have two categories that need to be represented separately. The choice of one-hot encoding for these variables is based on the consideration that each variable has two categories that need to be represented separately, allowing the machine learning model to understand and process information more effectively. Meanwhile, the label encoding method is a technique for converting labels into numerical form so that they can be entered into a machine learning model, where each category is given a unique integer value based on alphabetical order [16]. In this study, the variables stroke location, history of cardiovascular disease, previous medical history, and previous history of stroke were coded using label encoding. The choice of encoding label for this variable is based on the nature of the ordinal value it has. By using the label encoding method, sequence information and relationships between categories can be maintained in a numerical representation, making it suitable for variables with ordinal values.

3.2.4 Data Partition

The process of separating datasets is done to divide the data into two parts, namely the data that will be used to train the model and the data to be tested to see how well the model can

predict The most accurate ratio for separating data in health research is still a debate, and there is no definitive agreement as to the optimal ratio of each dataset. The most commonly used ratio is 80:20, which means 80% of the data is used for training and 20% for testing [17]. This study uses a separation configuration of 80:20%, where 80% of the data is used by the machine to learn and identify patterns, while the remaining 20% is used to test to what extent a model can make accurate predictions.

3.3 Machine Learning Classifier

3.3.1 Extreme Gradient Boosting Classifier

Extreme Gradient Boosting (XGB) is an open-source library that provides efficient and effective implementation of gradient enhancement algorithms [18]. XGB uses advanced regularization (L1 and L2), which enhances model generalization capabilities [19]. The basic concept of boosting is to build more accurate models by combining hundreds of simple tree models with low accuracy, in which each iteration will produce a new tree for the model. The next thing to do is pay attention to the complexity of the tree [20].

3.3.2 Category Boosting Classifier

Category Boosting Classifier (CatBoost) uses sequential target statistics and sequential improvements that make it good for heterogeneous data categorical values and has a strong performance compared to other gradient improvement decision tree implementations [21]. CatBoost has an overfitting detector that stops training when observing overfitting. This feature helps improve the generalization performance of the model and makes it more resilient to new health data [22].

3.3.3 Random Forest Classifier

The Random Forest (RF) ensemble model was developed to overcome the drawbacks of standard decision tree algorithms and offers a broad range of applications. In order to reduce the model's bias and variance, RF approaches include training numerous decision-tree learners at once [23]. Reliability of some class families, such as Random Forest (RF), for classification performance is supported by extensive research in general literature [24]. Research in medicine has shown that random forest algorithms are highly accurate in predicting diseases [25]. Random forests work well in complicated data sets, are resistant against unimportant features, and are simple to understand, quantify, and train rapidly.

3.3.4 Extra Tree Classifier

ExtraTree is a machine learning ensemble method that instructs a large number of decision trees and collects results from the tree ensemble to obtain estimates. However, there are a few differences between extra trees and random forests. They work in a slightly different way. Another important difference is in the way they choose where to share information within the decision tree. Extra Trees does this in a random way, which means he chooses a random value to divide the feature and create a new branch in the tree. On the other hand, Random Forest uses a more intelligent algorithm to find and select the best value for dividing the feature [26].

3.3.5 Decision Tree Classifier

The Decision Tree (DT) technique assembles the data pieces based on the traits that most differentiate the class until the termination requirements are satisfied by recursively splitting the data into many portions [27]. a popular kind of supervised machine learning

technique for analyzing different variables Its capacity to split data into branches or segments defines it. The decision tree's branches are oriented upward, with the ultimate outcome represented by the branches at the top [28].

3.4 Hyperparameter Tuning

In machine learning, there is a set of parameter values that are thought to improve model performance, called hyperparameters. Hyperparameter plays an important role in improving algorithm performance and has a major impact on model testing. Hyperparameter adjustment can be done manually or by trying a variety of predefined combinations of hyperparameters [29]. The study used the Optuna library to optimize the performance of predictive long-term hospitalization models for stroke patients. (validasi). We used the Bayesian sample algorithm and the Parzen estimator (TPE) structured tree by default. Optuna also provides shrinkage, i.e., an automatic initial stop to unpromising trials [28].

Table 2. Tuning parameters of the algorithms

Parameters	Description	Range
n_estimators	This controls how many trees will be used in the ensemble.	100-300
max_depth	Controls the maximum depth of trees in the model.	3 – 12
learning_rate	controls how big steps the model will take during the learning process.	0.1 - 1.0
subsample	The proportion of data samples used to train each tree.	0.5 - 1.0
gamma	The minimum threshold for splitting a node.	0.6 - 1.0
colsample_bytree	Proportion of features each tree.	0.0 - 1.0
reg_alpha	Regulation L1 to prevent overfitting	0.0 - 1.0
reg_lambda	Regulation L2 to control complexity	0.0 - 1.0
min_samples_split	Minimum number of samples to divide the node	0.1 – 1.0
min_samples_leaf	Minimum amount of sample in each leaf	0.1 – 0.5
max_features	Maximum characteristics for node separation	0.1 – 0.5
min_weight_fraction_leaf	Total weight ratio for one leaf Criteria for evaluation of node segregation	0.0 – 0.4
criterion	The criterion used to measure the quality of splitting at each node in the decision tree.	“gini”, “entropy
iterations	The number of iterations or steps taken by the model during the learning process.	100-500
12_leaf_reg	A specific parameter for setting trees.	1-10
border_count	The number of bins or categories for use on categorical data.	10-255
thread_count	The number of threads or threads to be used by the model during training.	-1
loss_function	The loss function to be optimized during the learning process.	Logloss
random_seed	Setting to ensure that the model's results can be reproduced.	123
verbose	The level of noise or information to be displayed during the training or model evaluation process.	False

3.5 Performance Evaluation Metrics

In evaluating the performance of the classification model in this study, the data is divided into two sets 80% for training and 20 for testing. The confusion matrix shows how often our models predict accurately and how often we predict inaccurately. False positives and false negatives are allocated to poor predictive values, while true positives and negatives are actually placed on the correctly anticipated values [30]. Precision metrics, recalls, F1 scores, and accuracy obtained using the confusion matrix are used for performance evaluation. The formula for calculating the value is shown below:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{F1-score} = \frac{2TP}{2TP+FP+FN} \quad (2)$$

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

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

Where TP = True positive, TN = True Negative, FP = False Positive, and FN = False Negative [31].

Furthermore, this work uses a Receiver Operating Characteristic (ROC) curve, which shows a trade-off between sensitivity and specificity at each threshold, to demonstrate how well a model performs [32]. A region score below the area under the ROC curve (AUC) indicates the likelihood that a randomly chosen positive sample will be rated higher than a randomly chosen negative sample. An AUC value more than 0.5 indicates that a model is superior to a random one, while an AUC of 1.0 indicates that a model is ideal [33].

5 Result, Analysis and Discussions

This research applied the 80:20 partition method, in which 80% of the total number of data samples was applied for model training purposes while the remaining 20% was used for testing. The total data in this set amounted to 106 samples. Of this figure, 84 samples were assigned as training data (X_train), with 22 samples as testing data (X_test).

5.1 Performance Evaluation of the Based Model

In this study, predicting the length of stay for stroke patients uses a comparison of five machine learning algorithms, namely XGBoost (XGB), Extra Tree, CatBoost, Decision Tree (DT), and Random Forest (RF), using a number of critical evaluation metrics such as accuracy, precision, recall, F1-score, and ROC AUC. Table 3 summarises the performance results of the five algorithms, where accuracy measures the extent to which the model is able to identify the correct class, and based on the results, the XGBoost (XGB) and Extra Tree models show the highest performance with an accuracy level of 68% each. Despite this, CatBoost only achieved 50% accuracy, indicating that this model may be less effective in classifying stroke patients by length of stay. Precision gives an idea of how many of the model's positive predictions are correct, and here XGBoost (XGB) and Extra Tree again stand out with higher levels of precision, especially in predicting the standard case. However, CatBoost has a low level of precision, especially in non-standard modes, indicating challenges in producing accurate and consistent predictions. When looking at recalls, XGBoost stands out with a score of 85%, demonstrating its expertise in identifying the majority of true positive cases. Meanwhile, CatBoost, despite having low accuracy (50%), showed a recall of 69%, indicating its ability to handle positive cases. The balance

between precision and recall is reflected in the F1 score. XGBoost achieved the highest F1-score of 76%, indicating an optimal balance between the ability to provide accurate positive predictions and the ability to identify the majority of true positive cases.

Meanwhile, additional analysis via the ROC AUC (Receiver Operating Characteristic Area Under the Curve) curve provides a deeper perspective on the algorithm's ability to differentiate between positive and negative classes. As shown in Figure 2, XGBoost shows the highest ROC AUC value with 0.71, indicating that this model has a good ability to differentiate between positive and negative cases while providing a lower false positive rate. Meanwhile, Random Forest also shows strong performance, with an ROC AUC value of 0.68. On the other hand, CatBoost shows the lowest ROC AUC value, namely 0.54, indicating that this model may have challenges in differentiating between the two classes. Extra Tree also has a relatively lower ROC AUC value of 0.58.

Overall, XGBoost (XGB) showed good performance in predicting the length of stay of stroke patients based on the overall evaluation of metrics. Several factors may explain why XGBoost (XGB) excels in this regard. First, XGBoost is an ensemble algorithm that utilises the combination of many decision trees, allowing the model to better capture the complexity of relationships between features. This capability helps in making decisions that are more accurate and adaptive to complex patterns in data. Furthermore, the high recall rate in Additionally, the fairly high precision, especially in standard mode, means that positive predictions from XGBoost are more likely to be truly relevant, reducing the number of false positives that can cause uncertainty and additional costs. In addition, the relatively high ROC AUC value indicates that XGBoost has a good ability to differentiate between stroke and non-stroke patients, indicating strong discrimination between the probability distributions of the two classes.

Table 3. Base Model Algorithm Performance

		Non-standar (1)	**Standar (0)**	**Accuracy**
Precission	XGB	67	69	**XGB** 68 %
	Extra tree	62	71	
	CatBoost	33	56	
	DT	56	69	**Extra Tree** 68 %
	RF	50	62	
Recall	XGB	44	85	
	Extra tree	56	77	**CatBoost** 50 %
	CatBoost	22	69	
	DT	56	69	
	RF	33	77	**DT** 64 %
F-1score	XGB	53	76	
	Extra tree	59	74	
	CatBoost	27	62	**RF** 59 %
	DT	56	69	
	RF	40	69	

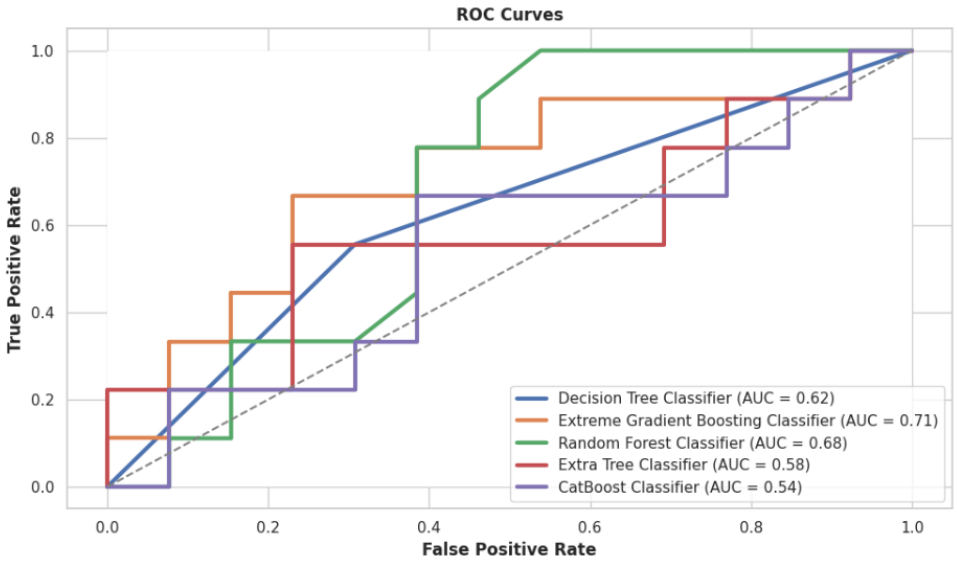

Fig 2. ROC curve for LoS prediction

5.2 Model Enhacement After Hyperparameter Adustment

Hyperparameter settings are an important step in ensuring that the machine learning model achieves optimal performance. The research performed hyperparameter settings on several machine learning algorithms, namely Extreme Gradient Boosting (XGB), Extra Tree, CatBoost, Decision Tree Classifier (DT), and Random Forest Classifiers (RF). The main focus of this initiative is to mobilize model capabilities for predicting the long-term hospitalization of stroke patients. The hyperparameter setting process is well done and involves a variety of parameter values used in the algorithms. The best results of the hyperparameter adjustments that have been made are documented in Table 4, presenting the optimal configuration for each algorithm. The importance of these hyperparameter settings is reflected in our efforts to make more accurate and efficient predictions related to vital parameters.

Table 4. Best hyperparameter values for LoS prediction

Algorithm	Best Hyperparameter Value
XGB	n_estimator : 267, max_depth : 12, learning_rate : 0.82, subsample : 0.92, colsample_bytree : 0.92, gamma : 0.28, reg_alpha : 0, reg_lambda : 0.0
Extra Tree	n_estimator : 392, max_depth : 3, min_samples_split : 0.74, min_samples_leaf : 0.39, max_features : 0.75, min_weight_fraction_leaf : 0.33, criterion : ‘entropy’
CatBoost	max_depth : 10, iterations : 446, learning_rate : 0.04, 12_leaf_reg : 6.54, Border_count : 21
DT	max_depth : 5, min_samples_split : 0.7, min_samples_leaf : 0.6, max_features : 0.38
RF	n_estimator : 91, max_depth : 7, min_samples_split : 0.57, min_samples_leaf : 0.22, max_features : 0.12

Based on the observations in Figure 3, which include the evaluation results related to the duration of hospitalization, it can be noted that there was a significant improvement in the performance of the model after hyperparameter adjustment. Table 5 presents the model evaluation results in two categories, Standard (0) and Non-Standard (1), with the precision,

recall, and F-1 score metrics. In the case of precision, the XGB model stands out with 86% accuracy on non-standard, whereas the DT is superior on standard with 83%. CatBoost shows a decrease in precision from standard (64%) to non-standard (50%). Extra Tree showed lower recalls in both categories. In the F-1 score, XGB dominated with a score of 86% on the standard and 75% on the non-standard. Overall, the XGB model shows good consistency in performance in both categories, with a good balance between precision and recall. There was a significant improvement in the accuracy of the XGB model, which reached 82%, making it the model with the highest precision among others.

On the other hand, there is an improvement in the performance of the model through the Area Under the Receiver Operating Characteristic Curve (ROC AUC) after the tuning process. Figure 5 shows an increase in the AUC value of the ROC, especially in the XGB model, by 79%, showing a significant improvement in performance. Besides, the Extra Tree and Random Forest models also experienced a striking improvement after hyperparameter settings. Through this hyperparameter adjustment process, it makes a real contribution to the ability of the model to provide more accurate predictions related to the duration of patient care, a crucial aspect in the management of hospital patient care. With increased accuracy, precision, recall, and F-1 score values, the models can now provide a more detailed and reliable view of the durations of patient treatment, both standard and non-standard.

XGBoost (Extreme Gradient Boosting) shows significant performance improvement after the hyperparameter tuning process, and this can be explained by several reasons. First, XGBoost has the ability to handle non-linear dependencies and complex feature interactions. By adjusting hyperparameters such as the number of trees (n_estimator), maximum depth of trees (max_depth), and learning rate (learning_rate), the model can be adjusted more precisely to the patterns present in the data. Second, optimal hyperparameter settings in XGBoost can overcome the problem of overfitting or underfitting, thereby increasing the generalisation of the model to new data. Tuning hyperparameters, such as setting subsample and colsample_bytree values, helps control how many subsets of the data are used in each iteration and how many features are used in each tree, optimizing the balance between variance and bias.

Furthermore, the use of hyperparameters such as gamma, reg_alpha, and reg_lambda in XGBoost can help regulate the level of model complexity and reduce overfitting tendencies. This is important because models that are too complex may tend to overfit noise in the data, while models that are too simple may fail to capture complex patterns. Apart from that, XGBoost's performance improvement can also be caused by its ability to handle imbalanced data. By tuning parameters such as scale_pos_weight, XGBoost can give greater weight to the minority class, which generally occurs in the case of predicting the length of stay for stroke patients.

Table 5. Algorithm Performance for LoS Prediction After Tuning

		Non-standar (1)	**Standar (0)**	**Accuracy**
Precission	XGB	86	80	**XGB** 82 %
	Extra tree	75	67	
	CatBoost	50	64	
	DT	70	83	**Extra Tree** 68 %
	RF	10	62	
Recall	XGB	67	92	

	Extra tree	33	44	**CatBoost** 59 %
	CatBoost	44	69	
	DT	78	77	
	RF	11	10	**DT** 77 %
F-1score	XGB	75	86	
	Extra tree	46	77	
	CatBoost	47	67	**RF** 64 %
	DT	74	80	
	RF	20	76	

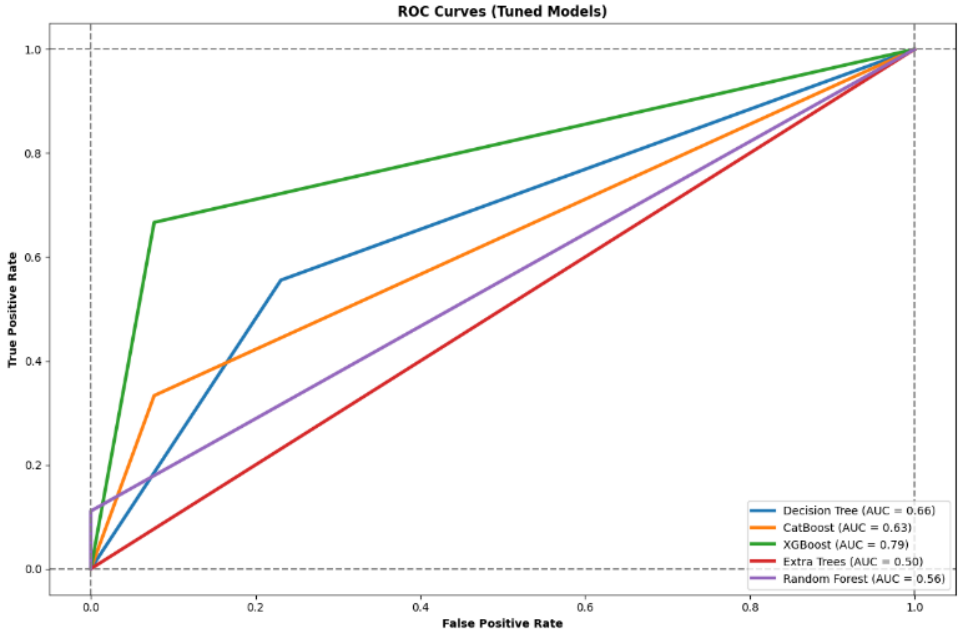

Fig 3. Tuned ROC curve for LoS prediction

5.3 Feature Importance

In this study, the Extreme Gradient Boosting Classifier (XGB) model showed excellent performance, especially after setting the hyperparameter, making it the highest-accuracy algorithm in predicting the hospitalization of stroke patients. In order to determine the characteristics that most influence hospitalization duration, we adopted the feature importance approach using the best algorithm, namely XGBoost. An analysis of the importance of this feature allows us to measure how much the influence of each feature is on the accuracy of the model [34]. In other words, features that have a more significant impact will have a higher level of importance in the prediction process. The evaluation of long-term hospital predictions also highlights key factors, as shown in Table 6. High total cholesterol and low levels of high-density lipoprotein (HDL) increase the risk of serious complications, which in turn affect patient long-term care. In addition, early hospital conditions, high levels of leukocytes, high triglyceride levels, low hemoglobin levels, high low-density lipoprotein levels (LDL), high blood pressure, and a high neutrophil-to-lymphosit ratio (NLR) are also identified as significant factors affecting the duration of hospitalization. This analysis provides in-depth insight into the contribution of each variable to predictive outcomes, providing a solid basis for decision-making in the management of stroke patient care.

Table 6. Feature Importance LoS Prediction

Length of Stay (LoS)	
Feature	*Score*
Chol Total	0.1084
HDL(High-Density Lipoprotein)	0.1006
Patient Condition	0.1002
LEU(Leukocytes)	0.0906
TG(Triglycerides)	0.0878
HB(Hemoglobin)	0.0851
LDL(Low-Density Lipoprotein)	0.0847
TR(Thrombocytes)	0.0742
NLR(Neutrophil-Lymphocyte Ratio)	0.0717
Stroke Location	0.0661

5.4 Implementation of web application

This research implements web applications with the extreme gradient boosting classifier algorithm (XGB). These algorithms have proven to be excellent at predicting the duration of hospitalization for stroke patients, especially after a careful hyperparameter setting process. The purpose of the application is to develop an XGB model that is integrated into the web through the Python Flask framework, so it can help the medical team estimate the hospitalization estimates of stroke patients. It is hoped that this application will facilitate more efficient medical decision-making.

1. An application developed using the XGBoost algorithm to predict hospitalization estimates for stroke patients The web-based interface built with Flask as its shell enables the development and provision of the interface so that users can easily fill in patient data, including relevant demographic and health history information.
2. The application actively processes the data sent by the user. This data is channeled into a machine learning model that has been trained using the XGBoost algorithm. The model training process involves data from previous stroke patients, enabling the model to understand patterns of death and identify risk factors that may affect the prediction outcome.
3. After receiving input from the user, the X GBoost model will analyze the input attributes and predict the duration of the stroke patient's stay based on the data.
4. The results of this prediction will then be sent back to the user via the web interface in the form of an output or report. The outputs or reports generated are not only informative but also provide a more detailed understanding of the duration of patient hospitalization. This data is presented in standard (0) and non-standard (1) categories, with the standard prediction (0) result indicating that patients can be expected to undergo hospital treatment in accordance with the prescribed standard time (<7 days). While the nonstandard prediction (1) result indicates that patients have a tendency to be undergoing hospital treatment beyond the prescriptive standard (>7 days).

The success of this application is directly related to financial issues, where the predictions provided not only provide an overview of the duration of patient care but also provide

valuable information regarding the potential financial impacts that may be faced. Thus, this application not only provides benefits in monitoring patient health but also provides a broader and more strategic view in the context of financial management related to stroke patient care. How it works, Users will be asked to fill in 17 attributes containing information such as the patient's age, gender, history of cardiovascular disease, past medical history, and other information. All input required in the input page form is numeric data. After all these attributes are filled in, the data will be taken by the application and processed through a machine learning model using the XGBoost algorithm. The prediction results, as shown in Figure 4, will be displayed to the user via the web application interface in output form. This output provides information about the expected long-term hospitalisation of stroke patients.

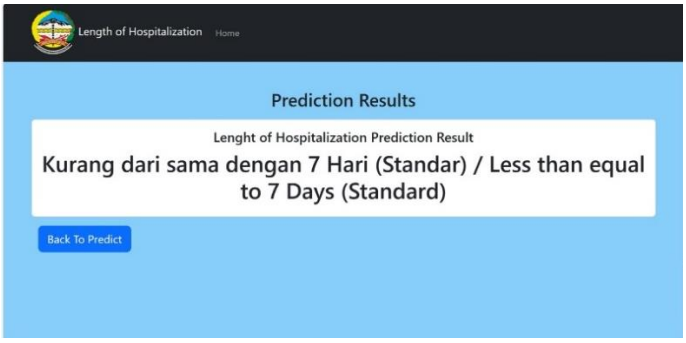

Fig 4. Prediction Result

5.5 Conclusion

The study uses machine learning (ML) algorithms to predict longevity (LoS) in stroke patients. Of the five algorithms, the XG Boost model is the best at predicting hospitalization length. In this study, the hyperparameter adjustment process carefully performed on the Extreme Gradient Boosting (XGBoost) model was able to improve predictive performance, with results achieving an accuracy of 82% and an area under the curve (AUC) of 79%. AUC-ROC score of 0.79, reflecting excellent performance in classifying the duration of hospitalization of stroke patients. In other words, the model is not only superior in predicting the duration of hospitalization but also effective in identifying and classifying patients accurately. Significant improvements in the accuracy and predictive modeling capabilities following the sophisticated hyperparameter setting process confirm that XGBoost remains a highly reliable first choice in the context of predicting long-term hospitalization in stroke patients.

In addition, a web application has been developed by integrating the XGBoost model into it. This web-based interface, built using the Python Flask framework, provides easy access for users to enter patient data, including relevant demographic and health history information. This process allows users, especially the medical team, to quickly and efficiently obtain estimates of the duration of hospitalization for stroke patients. Significant improvements in the accuracy and predictive capabilities of models following the sophisticated hyperparameter setting process emphasize that XGBoost remains a highly reliable primary choice in the context of predicting long-term hospital care for stroke patients.

However, this research has some limitations to bear in mind. First, the relatively small sample size of 106 patients is the main obstacle. Second, the research is being carried out in one of the hospitals in Indonesia, so the possibility of generalizing the findings to other

populations is limited. Thirdly, factors such as socio-economic status and access to health services are not taken into account, which may affect the results related to mortality and LoS. Therefore, future research is expected to address these constraints by using larger and more representative datasets of more diverse populations.

References

[1] H. H. Kyu *et al.*, "Global, regional, and national disability-adjusted life-years (DALYs) for 359 diseases and injuries and healthy life expectancy (HALE) for 195 countries and territories, 1990-2017: A systematic analysis for the Global Burden of Disease Study 2017," *Lancet*, vol. 392, no. 10159, pp. 1859–1922, 2018, doi: 10.1016/S0140-6736(18)32335-3.

[2] V. W. B, F. F. Rahman, and V. Ningrum, *Proceedings of the 3rd International Conference on Cardiovascular Diseases (ICCvD 2021)*, vol. 1. Atlantis Press International BV, 2023. doi: 10.2991/978-94-6463-048-0.

[3] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, and A. Saboor, "Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare," *IEEE Access*, vol. 8, no. Ml, pp. 107562–107582, 2020, doi: 10.1109/ACCESS.2020.3001149.

[4] Z. Ahmed, K. Mohamed, S. Zeeshan, and X. Q. Dong, "Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine," *Database*, vol. 2020, pp. 1–35, 2020, doi: 10.1093/database/baaa010.

[5] V. J. Pawar, "Lung Cancer Detection System Using Image Processing and Machine Learning Techniques," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 5956–5963, 2020, doi: 10.30534/ijatcse/2020/260942020.

[6] R. Chen *et al.*, "A study on predicting the length of hospital stay for Chinese patients with ischemic stroke based on the XGBoost algorithm," *BMC Med. Inform. Decis. Mak.*, vol. 23, no. 1, pp. 1–10, 2023, doi: 10.1186/s12911-023-02140-4.

[7] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 2623–2631, 2019, doi: 10.1145/3292500.3330701.

[8] C. Neto, M. Brito, H. Peixoto, V. Lopes, A. Abelha, and J. Machado, "Prediction of Length of Stay for Stroke Patients Using Artificial Neural Networks," *Adv. Intell. Syst. Comput.*, vol. 1159 AISC, no. Dm, pp. 212–221, 2020, doi: 10.1007/978-3-030-45688-7_22.

[9] B. K. Choi, S. W. Ham, C. H. Kim, J. S. Seo, M. H. Park, and S. Kang, "Development of a prediction model for length of stay for acute stroke patients using artificial intelligence," vol. 16, no. 1, pp. 231–242, 2018.

[10] J. Chrusciel, F. Girardon, L. Roquette, D. Laplanche, and A. Duclos, "The prediction of hospital length of stay using unstructured data," *BMC Med. Inform. Decis. Mak.*, vol. 4, pp. 1–9, 2021, doi: 10.1186/s12911-021-01722-4.

[11] M. Majidi Shad, A. Saberi, M. Shakiba, and S. Rezamasouleh, "Evaluating the Duration of Hospitalization and Its Related Factors Among Stroke Patients," *Casp. J. Neurol. Sci.*, vol. 4, no. 15, pp. 169–177, 2018, doi: 10.29252/cjns.4.15.169.

[12] R. Vickery, "The Art of Finding the Best Features for Machine Learning," *towardsdatascience.com*, 2020. https://towardsdatascience.com/the-art-of-finding-the-best-features-for-machine-learning-a9074e2ca60d (accessed Nov. 12, 2023).

[13] J. Yi, J. Lee, K. J. Kim, S. J. Hwang, and E. Yang, "Why Not To Use Zero Imputation? Correcting Sparsity Bias in Training Neural Networks," *8th Int. Conf. Learn. Represent. ICLR 2020*, vol. 1, pp. 1–27, 2020.

[14] S. Garg, "How to Deal with Categorical Data for Machine Learning," *https://www.kdnuggets.com/*, 2022. https://www.kdnuggets.com/2021/05/deal-with-categorical-data-machine-learning.html (accessed Oct. 20, 2023).

[15] Jason Brownlee, "Why One-Hot Encode Data in Machine Learning?," *machinelearningmastery.com*, 2020. https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ (accessed Jan. 08, 2024).

[16] Alakh Sethi, "One Hot Encoding vs. Label Encoding using Scikit-Learn,"

www.analyticsvidhya.com, 2023. https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/ (accessed Jan. 08, 2024).

[17] V. R. Joseph, V. R. Joseph, and H. M. Stewart, "Optimal ratio for data splitting," no. February, pp. 531–538, 2022, doi: 10.1002/sam.11583.

[18] D. Tarwidi, S. R. Pudjaprasetya, D. Adytia, and M. Apri, "An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach," *MethodsX*, vol. 10, no. March, p. 102119, 2023, doi: 10.1016/j.mex.2023.102119.

[19] A. Moore and M. Bell, "XGBoost, A Novel Explainable AI Technique, in the Prediction of Myocardial Infarction: A UK Biobank Cohort Study," *Clin. Med. Insights Cardiol.*, vol. 16, 2022, doi: 10.1177/11795468221133611.

[20] Cheshta Dhingra, "A Visual Guide to Gradient Boosted Trees (XGBoost)," *https://towardsdatascience.com/*, 2020. https://towardsdatascience.com/a-visual-guide-to-gradient-boosted-trees-8d9ed578b33 (accessed Oct. 20, 2023).

[21] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00369-8.

[22] N. Safaei *et al.*, *E-CatBoost: An efficient machine learning framework for predicting ICU mortality using the eICU Collaborative Research Database*, vol. 17, no. 5 May. 2022. doi: 10.1371/journal.pone.0262895.

[23] M. Ghazwani and M. Y. Begum, "Computational intelligence modeling of hyoscine drug solubility and solvent density in supercritical processing: gradient boosting, extra trees, and random forest models," *Sci. Rep.*, vol. 13, no. 1, pp. 1–11, 2023, doi: 10.1038/s41598-023-37232-8.

[24] R. S. Olson, W. La Cava, Z. Mustahsan, A. Varik, and J. H. Moore, "Data-driven advice for applying machine learning to bioinformatics problems," *Pacific Symp. Biocomput.*, vol. 0, no. 212669, pp. 192–203, 2018, doi: 10.1142/9789813235533_0018.

[25] P. C. K. b Trung Pham Dinh a c, Cuong Pham-Quoc a c, Tran Ngoc Thinh a c, Binh Kieu Do Nguyen a b, "A flexible and efficient FPGA-based random forest architecture for IoT applications," *Internet of Things*, vol. 22, 2023, doi: https://doi.org/10.1016/j.iot.2023.100813.

[26] K. Thankachan, "What? When? How?: ExtraTrees Classifier," *https://towardsdatascience.com/*, 2022. https://towardsdatascience.com/what-when-how-extratrees-classifier-c939f905851c (accessed Oct. 20, 2023).

[27] N. S. Chauhan, "Decision Tree Algorithm, Explained," *https://www.kdnuggets.com/*, 2022. https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html (accessed Oct. 20, 2023).

[28] B. O. Ogunleye, "Statistical Learning Approaches to Sentiment Analysis in the Nigerian Banking Context A thesis submitted in partial fulfilment of the requirements of Sheffield Hallam University for the degree of Doctor of Philosophy Bayode Oluwatoba Ogunleye October 2021," no. October, 2021.

[29] I. Muslim Karo Karo, "Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan," *J. Softw. Eng. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 11–18, 2020.

[30] 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, 2021, doi: 10.1155/2021/7633381.

[31] I. Imantoko, A. Hermawan, and D. Avianto, "Comparative analysis of support vector machine and k-nearest neighbors with a pyramidal histogram of the gradient for sign language detection," *Matrix J. Manaj. Teknol. dan Inform.*, vol. 11, no. 2, pp. 107–118, 2021, doi: 10.31940/matrix.v11i2.2433.

[32] I. G. Ivanov, Y. Kumchev, and V. J. Hooper, "An Optimization Precise Model of Stroke Data to Improve Stroke Prediction," *Algorithms*, vol. 16, no. 9, p. 417, 2023, doi: 10.3390/a16090417.

[33] D. Teoh, "Towards stroke prediction using electronic health records," *BMC Med. Inform. Decis. Mak.*, vol. 18, no. 1, pp. 1–11, 2018, doi: 10.1186/s12911-018-0702-y.

[34] Jason Brownlee, "How to Calculate Feature Importance With Python,"

https://machinelearningmastery.com/, 2020. https://machinelearningmastery.com/calculate-feature-importance-with-python/ (accessed Oct. 21, 2023).

Notes on contributors

Imam Tahyudin received Bachelor of Science (S.Si) from Universitas Jenderal Soedirman, Magister of Informatics (M.Kom) from Universitas Amikom Yogyakarta, Magister of Management (M.M.) from Universitas Jenderal Soedirman, and Doctor of Engineering (Dr.Eng) from Kanazawa University, Japan. He is a member of Computer Higher Education Association (APTIKOM), a member of Indoceiss and the secretary of Central Java Chapter of Indoceiss. He has been a lecturer since 2009 and now He is a Dean of Computer science faculty of Universitas Amikom Purwokerto. His research field in Artificial Intelligent, Data Mining, Machine Learning, Biological monitoring, IoT, DSS, and Information system. He has published more than 20 scientific articles in Scopus indexed, has 2 patents, has more than 9 copyright of software, and has published more than 15 scientific books.

Siti Alvi Solikhatin completed her Bachelor of Computer Science (S.Kom.) at STMIK Amikom Purwokerto and Master of Informatics Engineering (M.Kom) at Amikom University Yogyakarta. She is a member of the Indonesian Association of Informaticians (IAII) and has an International Certificate in Artificial Intelligence Associate (CAIA). Her research interests are artificial intelligence, cyber security, and information security. She has published 3 books, several accredited national journals, and IEEE indexed proceedings.

Ades Tikaningsih is a student in the Informatics Study Program at AMIKOM University, Purwokerto, who has a concentration in data and machine learning.

Puji Lestari is a student in the Information System Study Program at AMIKOM University, Purwokerto, who has a concentration in data Analyst and web application.

Hidetaka Nambo, Received his Ph.D. degree in 1999 from Kanazawa University. He was a Research Associate in 1999 in the Department of Electrical Information, Engineering Faculty of Kanazawa University. He has been a lecturer in the Graduate School of Natural Science and Technology since 2015. He is an associate professor in the College of Science and Engineering. He is engaged in research on monitoring systems by living plants and data mining. He is a member of the IEEE, the IEE of Japan, the IEICE of Japan, and the IPSJ of Japan.

Eko Winarto, graduated a bachelor's degree in nursing from the Gadjah Mada University, Yogyakarta. Completed a master's degree in nursing and medical surgical nursing specialist from the University of Indonesia, Jakarta. Currently working at the Banyumas District Hospital and as a contract lecturer at Karya Husada University, Semarang. Active in the Indonesian Medical Surgical Nurses Association and the islamic youth organization GP Ansor. Currently actively conducting research on degenerative diseases, especially on the immune system, heart and blood vessels and the nervous system.

Nazwan Hassa, graduated as a doctor at Sebelas Maret University, Surakarta. His education as a neurologist was completed at Diponegoro University, Semarang. Currently working at the Banyumas Regional General Hospital. He is active as an administrator of the Indonesian Association of Neurology Specialists (PERDOSNI).