RESEARCH ARTICLE

High performance of Dengue shock syndrome detection using extreme gradient boosting with ANOVA feature selection

Lailil Muflikhah[1, *], Agustin Iskandar[2], Novanto Yudistira[1], Isbat Uzzin Nadlori[3], Bambang Nur Dewanto[4]

[1]Faculty of Computer Science, Brawijaya University, Malang, Indonesia. [2]Faculty of Medicine, Brawijaya University, Malang, Indonesia. [3]Department of Informatics Engineering and Computer, Politeknik Elektronika Negeri, Surabaya, Indonesia. [4]Faculty of Technology Information, Merdeka University, Malang, Indonesia.

Received: November 1, 2023; **accepted:** December 30, 2023.

Dengue shock syndrome (DSS) is an infectious disease that affects millions of people every year all over the world. Early detection of DSS is essential for providing effective therapy and promoting patient recovery. In this work, we proposed a method to enhance DSS detection by combining the Extreme Gradient Boosting (XGBoost) algorithm with variance (ANOVA) feature selection analysis. We used a clinical dataset that contained important information gleaned from people with dengue virus infection. The dataset used for the research was collected from patients at Syaiful Anwar Hospital and consisted of 501 instances. Of these, 401 cases were related to DSS, while the other instances were unrelated to this specific medical condition. An analysis of variance (ANOVA) evaluated the most significant factors that distinguished persons with DSS from those with other dengue diseases. After that, the XGBoost model gave the characteristics of the selected features. For evaluation, we split the data into 80% and 20% for training and testing, respectively. The experimental result showed that the created model had a high degree of performance evaluation in detecting DSS. The highest performance achieved an accuracy of 0.839, precision of 0.875, recall of 0.92, and f1-score of 0.897. Most importantly, the model could detect DSS early on, enabling more proactive therapy and faster responses for individuals at risk of developing the condition.

Keywords: Dengue shock syndrome; ANOVA; XGBoost; feature selection.

[*]**Corresponding author:** Lailil Muflikhah, Faculty of Computer Science, Brawijaya University, Malang, Indonesia. Tel: +62 8123220198. Email: lailil@ub.ac.id.

Introduction

Dengue Fever, caused by the dengue virus, is a significant global public health concern. The disease has become a major issue with rising mobility and population density in Indonesia. In 2015, there were 126,675 documented cases of Dengue Fever (DF) across 34 provinces, resulting in 1,229 fatalities. Dengue Hemorrhagic Fever (DHF) outbreaks increased from 1,081 in 2014 to 8,030 in 2015 with the affected provinces and districts rising from 21 to 69. Dengue Shock Syndrome (DSS) is an acute manifestation of dengue hemorrhagic fever (DHF), distinguished by circulatory collapse or shock and the advancement of dengue virus infection. The patients have symptoms that exhibit extensive and abrupt signs of DHF or DF, and it is also a matter of clinical significance. According to Rajapakse (2011), a significant proportion of individuals diagnosed with dengue fever, ranging from around 30% to 50%, may develop shock and face potentially deadly consequences. These results are more likely to occur if the patients do not get timely and adequate medical intervention. The management of shock in dengue hemorrhagic fever (DHF) is a matter of utmost significance since the fatality rate escalates at the prompt and sufficient intervention for shock. Dengue shock syndrome (DSS), characterized by substantial plasma leakage, hypotension, and shock, is the most severe and possibly life-threatening manifestation. The fast and accurate diagnosis of DSS is of utmost importance for adequate care and improved patient outcomes [1]. In traditional clinical methodologies, hematocrit values and platelet counts are employed alongside clinical symptoms to diagnose Dengue Shock Syndrome (DSS). However, it does not possess the capability to diagnose problems effectively and expeditiously, perhaps resulting in delayed intervention.

In recent years, the integration of data-driven methodologies with machine learning has yielded significant advancements in the field of medical diagnostics. Research has demonstrated that machine learning algorithms can analyze extensive and intricate medical data, aiding in the early detection of illnesses. Previous research has investigated the application of machine learning techniques in identifying Dengue fever [1, 2]. The data sets utilized encompass several types, such as sequence, geographical, and clinical data. Research on Dengue fever obtained an accuracy rate of 79% using a decision tree algorithm with the patient's diagnosis report, medical history, and symptoms [2]. In Malaysia, the prediction of Dengue outbreaks used time series data of the number of cases with a linear regression method [3] and temporal data [4, 5]. DNA sequence data of the virus was used to classify the patient's Dengue fever and achieve high performance using a sequential pattern mining algorithm [6]. Another study used a spatial data clustering approach for identifying risk households of Dengue virus infection during the insecticide spraying ultra-low volume (ULV) period. It showed that the significant spatial pattern of Dengue vector populations affected high-risk areas of Dengue virus infection after insecticide treatment [7]. Moreover, several research investigations on Dengue, employing data analysis techniques, have utilized a statistical framework. The assessment of these studies encompassed several statistical models, including Logistic regression (59.1%), Linear Regression (17.4%), and General Linear Model with a success rate of 70% [8]. On the other hand, to improve the performance rate, several studies on disease detection applied feature selection hybrid with machine learning strategies to identify the most influential predictor for disease diagnosis [9]. For the other methods for feature selections, several studies applied statistical approaches, such as Chi^2 [10], ANOVA [11], and machine learning approaches including the Particle Swarm Optimization (PSO) algorithm [12], XGBoost algorithm [13], and Principal Component Analysis (PCA) [14].

DSS is a severe manifestation of Dengue fever, and timely detection is crucial for proper and timely medical intervention. Traditional methods of diagnosis may have limitations or may not be as accurate or efficient, hence the need for an improved and high-performance approach. Therefore, the purpose of this study was to develop an accurate and efficient method to identify and detect cases of Dengue Shock Syndrome by a hybrid feature selection method.

Materials and Methods

This research generally followed a structured pipeline, beginning with data acquisition, moving through data preprocessing (handling missing values and normalization), selecting the most important features, and ultimately applying the XGBoost classifier to build a predictive model for Dengue disease detection (Figure 1).

Data acquisition

The data of 501 patients with Dengue fever in Saiful Anwar Hospital, Malang, Indonesia were collected for this study. The proportion of data was distributed in class which detailed 401 patients categorizing as having Dengue fever and

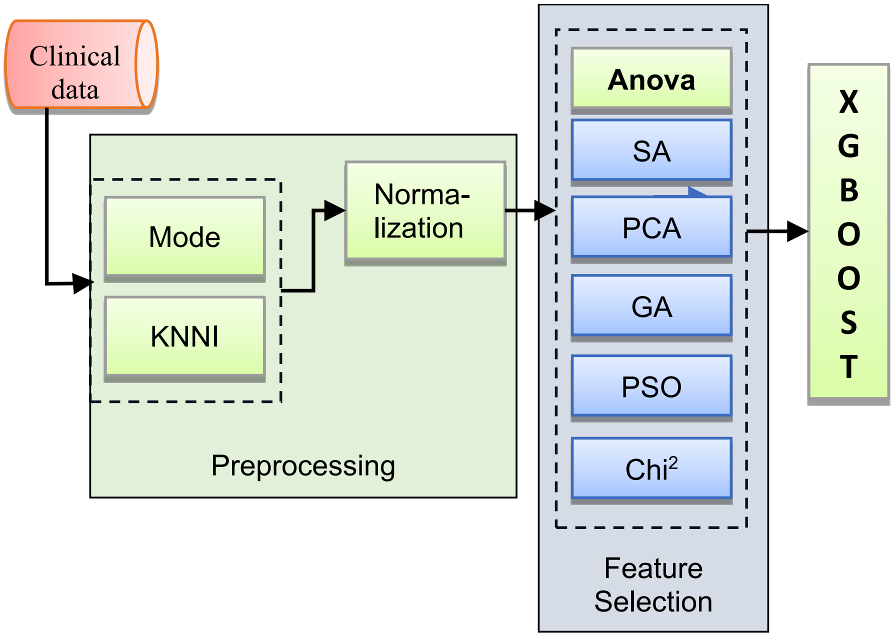

Figure 1. General proposed method.

100 patients experiencing shock syndrome. The clinical data included were as follows:

- Haemoglobin (Hb)
- Haematocrit (Hct)
- Thrombocyte
- Aspartate aminotransferase
- Alanine aminotransferase
- Eosinophil absolute
- Basophil absolute
- Neutrophil absolute
- Lymphocyte absolute
- Monocyte absolute
- Rumple Leede
- Hepatomegaly
- Splenomegaly
- Effuse pleura
- Nausea
- Vomiting
- Headache
- Epigastrium
- Kidney failure
- Hepatic failure

Once the data were obtained, it was imperative to undertake the necessary steps to ready it for analysis. In the raw data, there appeared to be data incompleteness and several values exhibited excessively wide ranges. Therefore, we need to pre-process data by completion of many tasks including handling missing values and normalization.

(1) Handling with missing values

Numerous datasets in practical applications often contain missing or partial data points. During this stage, it is necessary to decide on the handling of missing values, which may be addressed by either imputation, including the replacement of missing values with estimated values, or the removal of rows or features that include an excessive number of missing values. In this study, two methodologies were employed to address the issue, contingent upon the nature of the data. When dealing with numerical attribute values, an effective approach is the utilization of the K-Nearest Neighbors Imputation (KNNI) technique. Using the values of the nearest neighbors to guess missing values is what the K-Nearest Neighbors Imputation (KNNI) method does [15]. This method for finding lost data is still being worked on. For the reproducibility of this method, we utilized sci-kit-learn library in Phyton through the *KNNImputer* class within the *sklearn.impute* module.

$$d(X_i, X_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \qquad (1)$$

where a_i was i^{th} attribute or independent variable. x_i was i^{th} data.

On the other hand, when dealing with missing values in categorical data, the mode imputation approach is frequently employed. This approach prevents bias and maintains the original distribution. Determine the mode for each attribute, find the category attributes that have missing values, and then replace the missing values with the appropriate mode to fix the missing values. For each value that is lacking, repeat the steps.

(2) Normalization

For preparing data, the Min-Max scaling normalization method is employed. The process of scaling or modifying data values to a uniform range is referred to as normalization. This step is particularly important when dealing with data that is measured in multiple units or scales. This study utilized the Min-Max scaling normalization technique as a means of data preprocessing as stated in formulation (2).

$$normalized\ value = \frac{(original\ value - \text{minimum value})}{(maximum - minimum)} \quad (2)$$

Feature Selection

Feature selection is a crucial step in creating predictive models in data science and machine learning. It involves identifying key features from a larger dataset to enhance performance and reduce complexity. Researchers use methodologies like statistical analyses and machine learning algorithms to assess attributes' significance [16].

(1) ANOVA

One statistical method for locating and choosing the most pertinent features in a dataset—especially for tasks involving regression or classification—is ANOVA feature selection. It involves grouping data into different categories based on the target variable, computing variance, calculating an *F*-statistic, and calculating a p-value. Features with lower p-values are considered more important in explaining the target variable. Researchers can set a significance level (*alpha*) to control the trade-off between false positives and false negatives. ANOVA feature selection is useful in analyzing experimental data, clinical trials, and machine learning for selecting the most important features for predictive models, reducing the dimensionality of the dataset, and improving model performance. In general, the ANOVA statistic component is denoted as *F* and computed using formulation (3). For reproducibility, we utilized scikit-learn library in Python programming language.

$$F = \frac{MST}{MSE} \quad (3)$$

where *F* was ANOVA coefficient. MST was the mean sum of squares due to treatment. *MSE* was the mean sum of squares due to errors.

(2) Simulated Annealing (SA) method

A probabilistic optimization method called Simulated Annealing (SA) is used in data analysis and machine learning to identify the ideal subset of characteristics. Motivated by the metallurgical annealing process, SA facilitates the exploration of various combinations and performance evaluation to identify an ideal subset. The process starts with an initial subset of features, which can be randomly chosen or based on domain knowledge. An objective function is defined to measure the model's performance using the current subset of features. SA then perturbs the current solution by adding or removing features, ensuring they are reversible. The new solution is evaluated using the objective function, and if it improves performance, it is accepted as the new current solution. The algorithm gradually converges towards a near-optimal feature subset, and the best solution is returned once the temperature cools. In this study, we utilized *scikit-opt* library in python programming language for the reproducibility of program application for detection. In this method, we did several steps. We initialized the temperature as INIT-TEMP and the placement as INIT-PLACEMENT. While the temperature was greater than FINAL-TEMP, we entered a loop. Within this loop, we had another loop that continued if the inner loop criterion was FALSE.

We generated a new placement by perturbing the current placement, denoted as a new place. We then calculated the change in cost, denoted as ΔC, by subtracting the cost of the current placement from the cost of the new placement. If ΔC was less than zero, we updated the placement to be the new placement. Otherwise, if a randomly generated number between 0 and 1 was less than e raised to the power of -(ΔC/temp), we also updated the placement to be the new placement. Finally, we updated the temperature using the SCHEDULE function.

(3) Principle Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical methodology utilized to reduce the dimensionality of data by projecting it onto a lower-dimensional space, to preserve the maximum amount of original variation. The PCA is a technique that generates new variables, known as principal components, using linear combinations of the original characteristics. The objective of PCA is to optimize the variance among these principal components. It is used to decrease the dimensionality of a dataset by selecting a reduced number of principal components that effectively capture a significant portion of the variation present in the data [17]. Although PCA does not conduct feature selection, it aids in comprehending the primary components' contribution from the original features, hence facilitating an explanation of the variation in the data. Nevertheless, it does not eliminate characteristics seen in conventional methods. For reproducibility, we utilized the scikit-learn library to implement PCA method in the relevant machine learning.

(4) Genetic algorithm

An evolutionary optimization method called the Genetic Algorithm (GA) is widely used in machine learning and data analysis to choose which features to use. To find the set of attributes that are most useful for optimizing a certain objective function, the process acts like natural selection. The process has several important steps, such as starting the population, figuring out the fitness function, choosing individuals, crossing over genetic material, changing genetic material, replacing individuals, figuring out when the process should end, and finding the final solution. A machine-learning model is judged by its fitness function, which looks at certain factors. It is biased toward solutions with higher fitness scores and gives them more weight during the search process. The GA can quickly look through a very large solution space, even if there are a lot of factors to think about. To ensure convergence towards a useful feature subset, it is important to optimize algorithm parameters like population size, crossover rate, mutation rate, and termination conditions. It is implemented as a computational optimization technique for feature selection. For reproducibility, we utilized scikit-learn in Python programming language by customizing code through DEAP library.

(5) Particle Swarm Optimization (PSO)

In the fields of machine learning and data analysis, Particle Swarm Optimization (PSO) is a popular way to find the best solution. Its main objective is to find the best subset of features that will effectively improve a certain objective function. PSO is made up of several important parts, such as setting up particles, evaluating fitness functions, keeping track of personal best positions, and finding the global best position. When particles change their speeds, they consider where they are now, where they would like to be, and where the best positions are around the world. By making this change, they can get closer and closer to the best feature subsets. The algorithm keeps updating particles and the feature subsets that go with them until a certain condition is met. When the method is over, the chosen subset of features corresponds to the global best position. Because it can search the feature space well and handle complex, non-linear interactions between features and the target variable well, Particle Swarm Optimization (PSO) has many benefits. In any case, how well the system works depends on choosing the right parameters and the details of the situation. This method is applied to select the potential predictor in this study. For reproducibility, we

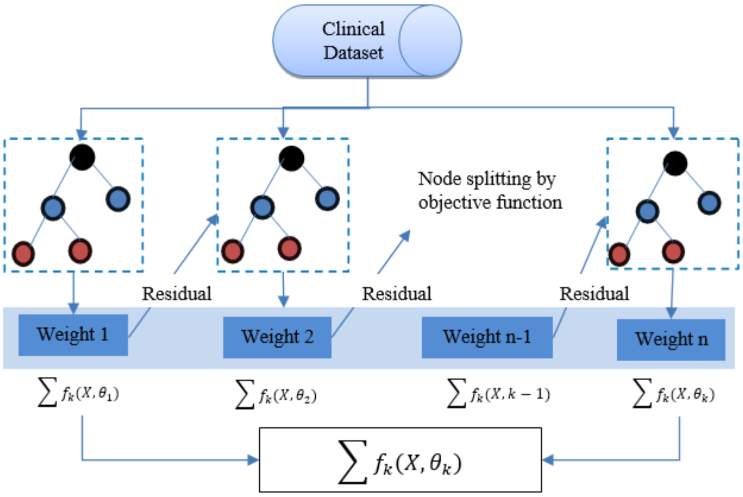

Figure 2. Illustration of XGBoost algorithm.

utilized scikit-opt library in Python language programming.

(6) Chi-squared (χ^2)

Chi-squared feature selection is a statistical method for feature selection in this study. It involves creating a contingency table for each feature and comparing observed and expected frequencies in a cell. The Chi-squared statistic (χ2) is calculated to identify the important feature based on the *p-value*. For reproducibility, we utilized sci-kit-learn in the Python programming language. The degrees of freedom depend on the number of categories in the feature and the number of classes in the target variable. The p-value represents the significance of the relationship between the feature and the target variable. Features are ranked based on their p-values, with smaller p-values indicating more significant associations. However, it may not be as effective for continuous or mixed-type data and may not capture complex non-linear associations.

Extreme Gradient Boosting (XGBoost) algorithm

Extreme Gradient Boosting (XGBoost) is an ensemble method of machine learning algorithms. It is applied to more than one classifier by boosting the decision tree method with updated serial weight (Figure 2). XGBoost is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form. The weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model as follows.

$$obj(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (4)$$

where *obj(θ)* was regularized objective. $\hat{y}_i$ was *i*th prediction. y_i was *i*th target. *n* was the number of samples. *K* was the number of trees. *l* was the difference between prediction $\hat{y}_i$ and the target y_i. Ω was complexity value of the model. *f* was functional space of F.

Furthermore, the prediction $\hat{y}_i$ is obtained from functional space *f* of samples *X* involved into CART's algorithm $\mathcal{F}$ as stated in formulation (5). The first term is the loss function and the second is the regularization parameter.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(X_i), f_k \epsilon \mathcal{F} \qquad (5)$$

Table 1. The details of selected clinical data.

Feature Selection Method	The number of clinical data	The selected clinical data (feature)
None	40	['LOS', 'age', 'body mass', 'dfever', 'HB1', 'HBSEL', 'HCT1', 'HCTSEL', 'TROM1', 'TROMSEL', 'LEU', 'AST', 'ALT', 'UR', 'CR', 'PPT', 'APTT', 'EOSABS', 'BASABS', 'NEUABS', 'LIMABS', 'MONABS', 'gender', 'fever', 'Vomiting', 'Nausea', 'epigastrium', 'cephalgi', 'bleeding', 'Rumpel leede', 'Hepatomegali', 'SPLENOMEGALI', 'ASITES', 'EFUSIPLEURA', 'ENCEPHALOPATI', 'ENCEPHALITIS', 'kidney failure', 'Hepatic failure', 'MYOCARDITIS', 'contact history']
ANOVA	10	['vomiting', 'nausea', 'epigastrium', 'RUMPLELEEDE', 'HEPATOMEGALI', 'ASITES', 'EFUSIPLEURA', 'ENCEPHALOPATI', 'hepatic failure', 'LOS']
PCA	20	['LOS', 'age', 'body mass', 'dfever', 'HB1', 'HBSEL', 'HCT1', 'HCTSEL', 'TROM1', 'TROMSEL', 'LEU', 'AST', 'ALT', 'UR', 'CR', 'PPT', 'APTT', 'EOSABS', 'BASABS', 'NEUABS', 'LIMABS', 'MONABS', 'gender', 'fever', 'Vomiting', 'Nausea', 'epigastrium', 'cephalgi', 'bleeding', 'Rumpel leede', 'Hepatomegali', 'SPLENOMEGALI', 'ASITES', 'EFUSIPLEURA', 'ENCEPHALOPATI', 'ENCEPHALITIS', 'kidney failure', 'Hepatic failure', 'MYOCARDITIS', 'contact history']
GA	24	['age', 'dfever', 'HB1', 'TROM1', 'ALT', 'UR', 'PPT', 'APTT', 'EOSABS', 'NEUABS', 'LIMABS', 'fever', 'epigastrium', 'cephalgi', 'bleeding', 'Rumpel leede', 'Hepatomegali', 'SPLENOMEGALI', 'EFUSIPLEURA', 'ENCEPHALITIS', 'kidney failure', 'Hepatic failure', 'MYOCARDITIS', 'contact history']
SA	21	['LOS', 'age', 'body mass', 'HCT1', 'HCTSEL', 'LEU', 'AST', 'APTT', 'EOSABS', 'NEUABS', 'LIMABS', 'Vomiting', 'Nausea', 'bleeding', 'Rumpel leede', 'SPLENOMEGALI', 'ASITES', 'EFUSIPLEURA', 'ENCEPHALOPATI', 'ENCEPHALITIS', 'Hepatic failure']
PSO	19	['LOS', 'body mass', 'dfever', 'HB1', 'LEU', 'AST', 'CR', 'PPT', 'BASABS', 'NEUABS', 'MONABS', 'gender', 'fever', 'Nausea', 'epigastrium', 'Rumpel leede', 'Hepatomegali', 'MYOCARDITIS', 'contact history']
Chi^2	20	['age', 'TROM1', 'TROMSEL', 'LEU', 'UR', 'PPT', 'NEUABS', 'LIMABS', 'MONABS', 'gender', 'fever', 'Vomiting', 'Nausea', 'epigastrium', 'cephalgi', 'bleeding', 'Rumpel leede', 'SPLENOMEGALI', 'MYOCARDITIS', 'contact history']

Notes: LOS: length of stay in hospital. HB1: hemoglobin on the 1st day. HCT: hematocrit. Trom: Thrombocyte. AST: aspartate aminotransferase. ALT: alanine aminotransferase. EOSABS: eosinophil absolute. BASABS: basophil absolute. NEUABS: neutrophil absolute. LIMABS: lymphoid absolute. MONABS: monocyte absolute. PT: partial thrombin. APTT: activated partial thrombin time.

XGBoost is an open-source machine-learning library and can be accessed by anyone. For reproducibility, this library is implemented to select the important feature and to build a classifier model. It was developed by Tianqi Chen (2016) and is maintained as an open-source project on GitHub [18].

Results and discussion

To know the performance of the method, we used the proportion of datasets as 401 (80%) of data training and 100 (20%) of data testing. In the feature selection stage, various reduced numbers of features were generated until the best performance was achieved with the XGBoost classifier as shown in Figure 3. The number of the selected features was in the range 9 to 24 and the details of the selected clinical data were shown in Table 1. They were evaluated for precision, recall, and f1-score as included in Table 2. The best performance evaluation was achieved using a statistical approach, ANOVA feature selection.

This study focused on detecting Dengue shock syndrome, a severe complication of dengue fever, and employed a computational approach

Table 2. Comparison performance evaluation using various feature selection.

FS Method	Mean Precision	Mean Recall	Mean F1-Score	Mean Accuracy
ANOVA	0.8746	0.9203	0.8966	0.8304
None	0.8556	0.9177	0.8853	0.8103
SA	0.8562	0.9177	0.8853	0.8104
PCA	0.8578	0.9078	0.8814	0.8045
GA	0.8599	0.9028	0.8808	0.8045
PSO	0.8432	0.9102	0.8752	0.7924
Chi^2	0.8321	0.8929	0.8612	0.7706

and utilized machine learning methods for the detection process. It emphasized that the incorporation of determinative predictors into the prediction model was a crucial component of this study to enhance the efficiency of Dengue shock illness detection. For this purpose, the study developed a prediction model using computational approaches, by injecting the feature selection method to XGBoost algorithm. Certain variables or predictors that were thought to be essential for producing precise predictions were included in this model. Improvements in performance measurements like the Area Under the Receiver Operating Characteristic (ROC) Curve (AUC), which gauged how well the model could distinguish between positive and negative situations was shown in Figure 4. The proposed method also demonstrated the decrease in the loss function that was anticipated as the results of this study (Figure 5). The study also attempted to develop a more stable and trustworthy prediction model for the identification of Dengue shock syndrome, which was essential for prompt diagnosis and treatment of cases of this serious illness.

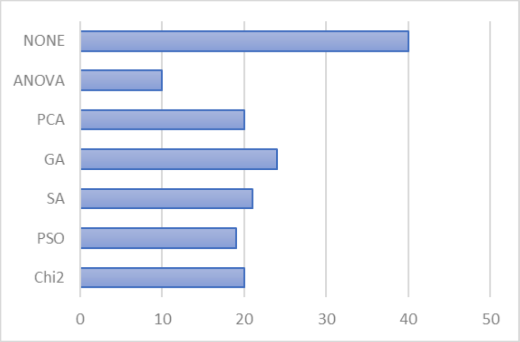

Figure 3. The number of clinical data selected.

Using the XGBoost classifier, the learning process aimed to identify the ideal subset of characteristics that yielded the greatest performance. The process of feature selection entailed deleting certain features from the original set, and it seemed that various feature subsets were tried repeatedly until the XGBoost classifier performed at its peak, which implied that to improve the XGBoost model's predicting skills, the study or analysis was concentrated on identifying the most illuminating feature set.

A loss function is a crucial component in machine learning and optimization algorithms, calculating the error between predicted and actual values in a model. Its primary purpose is to guide the model towards minimizing this error during the training process. There are various types of loss functions, each suites to different types of machine learning tasks. The choice of loss function depends on the specific problem and data nature. The loss function's value can vary widely depending on the problem and the specific loss function used. During training, the model's parameters are adjusted to minimize the loss function, and the choice of the appropriate loss function significantly impacts a model's robustness, generalization, and suitability for a particular task variance [19, 20]. The right feature selection helps identify patterns, reduce overfitting, and improve model performance. The appropriate use of feature selection techniques leads to more effective and understandable predictive models [9, 10]. Then, the AUC is a widely used metric in machine learning and data analysis to assess the performance of classification models. It is based

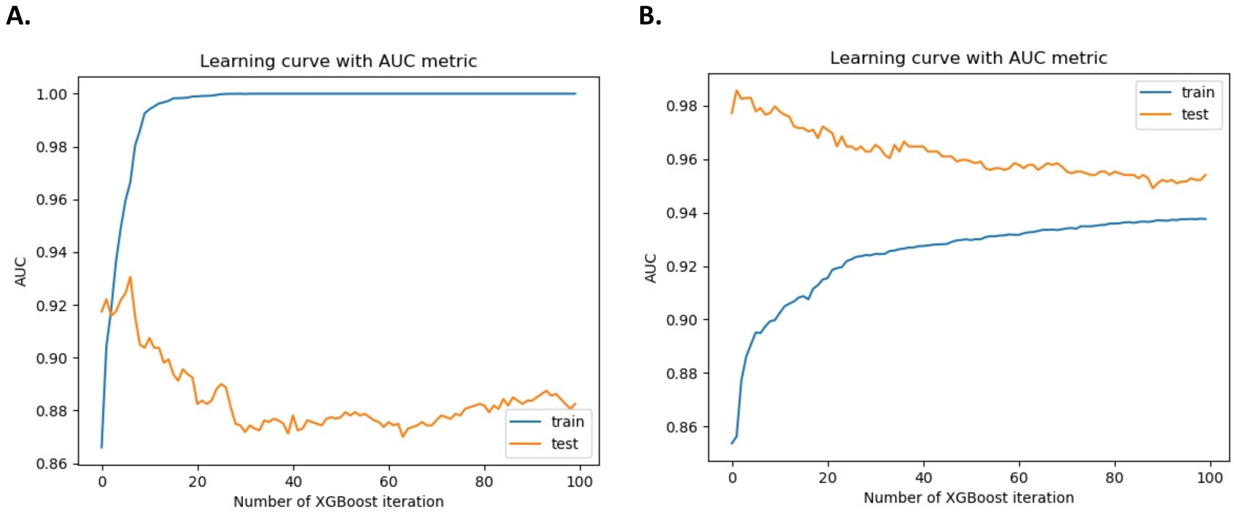

Figure 4. The AUC using XGBoost method. **A.** without feature selection. **B.** using ANOVA feature selection.

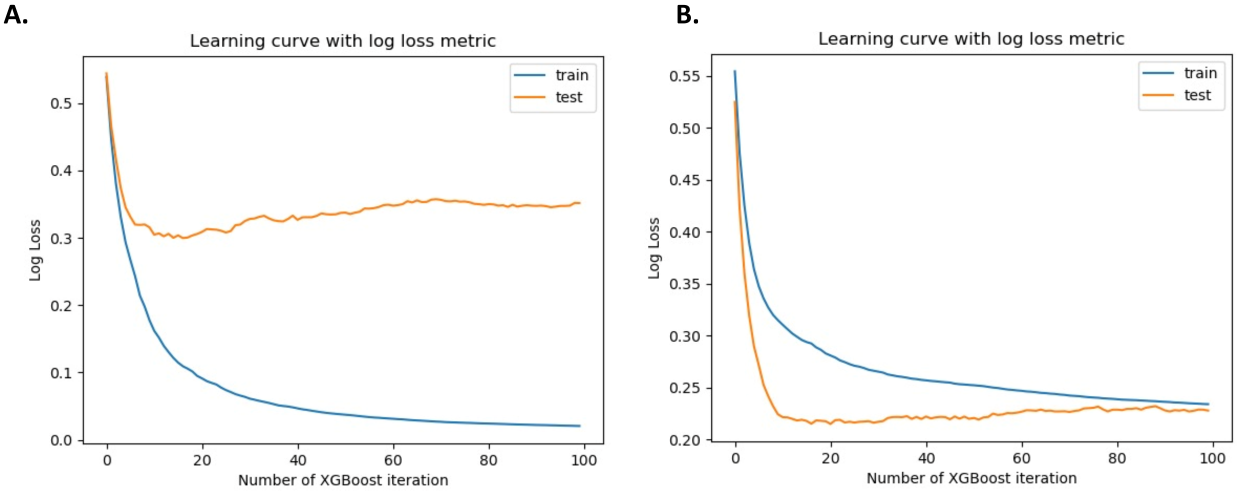

Figure 5. Loss function using XGBoost algorithm. **A.** without feature selection. **B.** using ANOVA feature selection.

on the ROC curve, which plots the True Positive Rate against the False Positive Rate for different discrimination thresholds. AUC is calculated as the numerical value under the ROC curve with a higher AUC indicating better model performance. AUC is interpretable as 1 for perfect classifiers, 0.5 for random classifiers, and > 0.5 for better models. AUC is useful for model comparison, as it is less sensitive to class imbalance and threshold-independent, evaluating a model's performance across all possible thresholds. It is particularly useful when evaluating a model's overall discriminatory ability without being tied to a specific decision threshold [21].

Conclusion

A combination of ANOVA feature selection and the XGBoost classifier has shown promise in enhancing the early detection of dengue shock syndrome and achieving the highest performance for result evaluation. This research highlights the potential for data-driven approaches to significantly improve clinical decision-making and patient outcomes in the context of critical medical conditions.

Acknowledgment

We thanks RISTEK/DIKTI for conducting this research. The Indonesian government provided financial support for this study under contract number 119/E5/PG.020.00.PL/2023, which was issued on June 19, 2023, through the Ministry of Research and Technology/National Agency for Research and Innovation (RISTEK/DIKTI).

References

1. Yuan K, Chen Y, Zhong M, Lin Y, Liu L. 2022. Risk and predictive factors for severe dengue infection: a systematic review and meta-analysis. PLOS ONE. 17(4):e0267186.
2. Sarma D, Hossain S, Mittra T, Bhuiya MdAM, Saha I, Chakma R. 2020. Dengue prediction using machine learning algorithms. IEEE 8th R10 Humanitarian Technology Conference (R10-HTC). pp:1-6.
3. Husin NA, Salim N, Ahmad AR. 2008. Modeling of dengue outbreak prediction in Malaysia: a comparison of neural network and nonlinear regression model. International Symposium on Information Technology. (3):1-4.
4. Mohd Sharef N, Husin NA, Kasmiran KA, Ninggal MI. 2019. Temporal trends analysis for dengue outbreak and network threats severity prediction accuracy improvement. Journal of Digital Information Management. 17(3):122
5. Nynalasetti KKR, Varma G, Rao M. 2014. Classification rules using decision tree for dengue disease. IJRCCT. (3):340-343.
6. Marimuthu T, Balamurugan V. 2015. A novel bio-computational model for mining the dengue gene sequences. Accessed April 12, 2023. https://www.semanticscholar.org/paper/a-novel-bio-computational-model-for-mining-the-gene-Marimuthu-Balamurugan/aa6c9fd853eeb246e49355f00050f55e4f501cfc
7. Sudsom N, Thammapalo S, Pengsakul T, Techato K. 2016. A spatial clustering approach to identify risk areas of dengue infection after insecticide spraying. Jurnal Teknologi. 78:73-77.
8. Hoyos W, Aguilar J, Toro M. 2021. Dengue models based on machine learning techniques: A systematic literature review. Artificial Intelligence in Medicine. 119:102157.
9. Dey SK, Uddin KMM, Babu HMdH, Rahman MdM, Howlader A, Uddin KMA. 2022. Chi^2-MI: A hybrid feature selection-based machine learning approach in diagnosis of chronic kidney disease. Intelligent Systems with Applications. 16:200144.
10. Javid I, Zager Alsaedi AK, Ghazali R, Mohmad Hassim YM, Zulqarnain M. 2022. Optimally organized GRU-deep learning model with Chi2 feature selection for heart disease prediction. J Intell Fuzzy Syst. 42(4):4083-4094.
11. Moorthy U, Gandhi UD. 2021. A novel optimal feature selection technique for medical data classification using ANOVA based whale optimization. J Ambient Intell Human Comput. 12(3):3527-3538.
12. Rostami M, Forouzandeh S, Berahmand K, Soltani M. 2020. Integration of multi-objective PSO-based feature selection and node centrality for medical datasets. Genomics. 112(6):4370-4384.
13. Thenmozhi T, Helen R. 2021. Feature selection using extreme gradient boosting Bayesian optimization to upgrade the classification performance of motor imagery signals for BCI. J Neurosci Methods. 366:109425.
14. Tripathi A, Rani P. 2023. PCA-based feature selection and hybrid classification model for speech emotion recognition. In: Hassanien AE, Castillo O, Anand S, Jaiswal A. (eds) International Conference on Innovative Computing and Communications. ICICC 2023. Lecture Notes in Networks and Systems. Springer, Singapore. 703:347-353.
15. Muflikhah L, Hidayat N, Hariyanto DJ. 2019. Prediction of hypertention drug therapy response using K-NN imputation and SVM algorithm. Indonesian Journal of Electrical Engineering and Computer Science. 15(1):460-467.
16. Rostami M, Berahmand K, Forouzandeh S. 2021. A novel community detection based genetic algorithm for feature selection. Journal of Big Data. 8(1):2.
17. Taguchi YH. 2016. Principal component analysis based unsupervised feature extraction applied to publicly available gene expression profiles provides new insights into the mechanisms of action of histone deacetylase inhibitors. Neuroepigenetics. 8:1-18.
18. Chen T, Guestrin C. 2016. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. 2016:785-794.
19. James GM. 2003. Variance and bias for general loss functions. Machine Learning. 51(2):115-135.
20. Nasiri H, Alavi SA. 2022. A novel framework based on deep learning and ANOVA feature selection method for diagnosis of covid-19 cases from chest x-ray images. Comput Intell Neurosci. 2022:4694567.
21. Tian M, Yu J, Kim J. 2023. Estimation of the area under a curve via several B-spline-based regression methods and applications. J Biopharm Stat. 30(4):704-720.