Journal of Applied Science, Engineering, Technology, and Education *ISSN: 2685-0591 (online)*
Vol. 7 No. 1 (2025)
https://doi.org/10.35877/454RI.asci4191

Implementation of Machine Learning Algorithm with Extreme Gradient Boosting (XGBoost) Method in Hypertension Level Classification

Zulkifli Rais*, Muhammad Fahmuddin S, Saida, & Agung Tri Utomo

Department of Statistics, Faculty of Mathematics and Natural Science ,Universitas Negeri Makassar, Makassar, Indonesia

Abstract

The increasing number of hypertension patients and the threat of serious complications make hypertension one of the leading causes of death worldwide. Early prevention is currently considered one of the best solutions. Early prevention through early detection can be achieved by utilizing machine learning technology. XGBoost is a machine learning algorithm based on gradient boosting machines. XGBoost applies regularization techniques to reduce overfitting and has faster execution speed as well as better performance. The objective of this research is to classify hypertension levels using the XGBoost method and leveraging hyperparameter tuning for optimization. In this study, the hyperparameter optimization technique used is gridsearchCV. The evaluation results of the XGBoost classification method using the best combination of parameters show good performance, where the XGBoost model achieves an accuracy of 93.3%, Precision of 97%, Recall of 92%, F1-Score of 93%, and AUC value of 0.935. This implies that the classification of hypertension levels in patients at Pelamonia Makassar Hospital can be well or accurately classified using the XGBoost method.

Keywords: Hypertension, Machine Learning, XGBoost

Received: 9 January 2025 Revised: 29 March 2025 Accepted: 15 April 2025

1. Introduction

Hypertension or high blood pressure is a significant health issue and a global public concern. It is the most common chronic disease and a crucial risk factor for cardiovascular and cerebrovascular diseases (Chang et al., 2019). Hypertension is caused by abnormal increases in blood pressure, both systolic and diastolic. It is diagnosed when systolic blood pressure is equal to or greater than 140 mmHg and diastolic blood pressure is greater than or equal to 90 mmHg (Lindblom et al., 2024).

According to the World Health Organization (WHO), one billion people worldwide suffer from hypertension. This number is expected to continue increasing each year (Sianipar, 2025). By 2025, an estimated 29% of adults worldwide are projected to experience high blood pressure. Additionally, hypertension can lead to serious complications such as stroke, coronary heart disease, etc. (Luo et al., 2018). This disease poses a serious threat, and the risk of complications is not easily detected, as around 50% of young hypertensive patients do not show symptoms (Chang et al., 2019), contributing to a high mortality rate.

Hypertension is the leading cause of death globally each year. According to the WHO, hypertension kills around 8 million people annually, including 1.5 million in Southeast Asia. The high mortality rate due to hypertension requires attention and proper management. One possible approach is early detection for prevention purposes.

Medical evidence indicates that one way to reduce the impact of hypertension is early detection(Martinez-Ríos et al., 2021). Therefore, a method is needed that can quickly and accurately detect hypertension based on the patient's condition. A commonly used method is classification, which involves grouping data or subjects according to predefined

* Corresponding author.
E-mail address: zulkifli.rais89@unm.ac.id

standards. The abundance of clinical data available in electronic medical records facilitates the use and development of various methods, including machine learning techniques (Koshimizu et al., 2020).

Several previous studies have utilized machine learning for classifying types of hypertension. A study by (Mroz et al., 2024), employed a machine learning model for classifying hypertension types based on personal features, achieving an accuracy rate of 73.3%. Another study by (Sianipar, 2025) used the LVQ machine learning model for classifying hypertension types following the ESH guidelines, achieving an accuracy of 94.6%. Additionally, research conducted by (López-Martínez et al., 2018) used machine learning for classifying the hypertension population with various risk factors, obtaining an AUC value of 73%.

Machine learning technology operates by utilizing computers to learn from data and produce data classifications. One commonly used method in machine learning is ensemble learning. Research by Zhang has shown that ensemble learning can combine multiple classifier models to achieve better performance, ensuring generalization ability and classification result stability (Y. Zhang et al., 2019; Ahmar et al., 2024). A commonly used ensemble method is gradient boosting, which employs a gradient descent boosting approach.

Gradient boosting, introduced by (J. H. Friedman, 2001a), is further developed into Extreme Gradient Boosting (XGBoost) by (Chen & Guestrin, 2016). XGBoost is an extension of three classic Gradient Boosting Machine (GBM) algorithms and is only used for labeled data in its training process (Maalik et al., 2019). XGBoost offers several advantages compared to other algorithms. It is a regression and classification algorithm with ensemble methods. Its most notable advantage is its speed, being ten times faster than gradient boosting (Chen & Guestrin, 2016). Due to these advantages, XGBoost is considered one of the best choices for data classification.

Previous studies have found that XGBoost classification is superior to random forest methods (Maalik et al., 2019), as well as when compared to support vector machines (SVM) (Liew et al., 2021). In terms of accuracy, computational time, and resource usage, XGBoost is considered one of the best choices (Goyal et al., 2021).

Based on this background, the studied aims to contribute specifically to the use of the XGBoost machine learning model for hypertension level classification and utilize hyperparameter tuning for optimization, aiming to achieve better accuracy results in the classification model.

2. Literature Review

2.1. Classification

The goal of classification is to accurately predict categories for unknown data. Classification algorithms can be applied to categorical data. If the target data is numeric, regression algorithms are used for prediction models (Sumathi et al., 2016). Classification is a method of sorting or categorizing objects based on specific features, much like how humans try to distinguish one object from another based on four classification components:

(a). Class: It is a dependent variable consisting of categorical data representing the label of an object.

(b). Predictor: It is an independent variable representing the characteristics of an object to be classified.

(c). Training Data: It is a set of data consisting of the two previous components, class, and predictor. This data is used to train the model in the process of classifying objects, resulting in the appropriate class based on the prediction results from the model.

(d). Testing Data: It is new data used for classification by the model obtained from the training data. This is done to evaluate the accuracy of the classification results (Gorunescu, 2011).

2.2. Ensamble Learning

Ensemble learning involves multiple sets of training used to solve the same problem, combining the results from individual classifications and then merging them using ensemble techniques to create a single grouping, enhancing the performance of a single classification (Tsai et al., 2021). The ensemble approach is implemented by combining several machine learning models to achieve improved performance. The goal of ensemble learning is to overcome the weaknesses of one classifier by leveraging the strengths of another. The process of ensemble learning can be illustrated as follows:

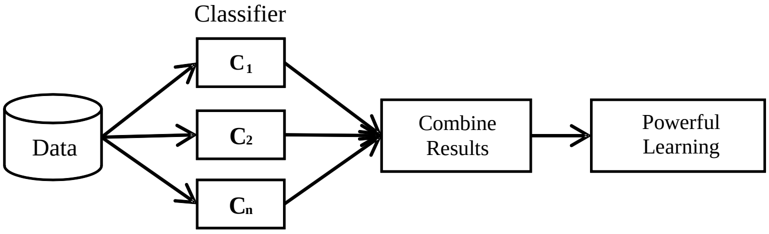

Fig. 1. Ensamble Process Illustration

One popular ensemble learning algorithm is boosting. Boosting introduced by Robert E. Schapire in 1998 is one method of ensemble learning that can improve the performance of some weak classification results in order to be a robust classification process. The boosting technique can be seen as an averaging model method that was originally designed for classification methods but can also be applied to regression methods (Syarif et al., 2012). Boosting brings together models that are of the same type and utilizes voting for classification or calculating the numerical estimated mean for the output of a single individual model. Unlike bagging which builds individual models separately, boosting builds new models from the performance results of previously built models (Subasi & Yaman, 2020).

2.3. Extreme Gradient Boosting (XGBoost)

The XGBoost method was first introduced by (J. H. Friedman, 2001a) in his research, Friedman connects boosting and optimization in building a gradient boosting machine (GBM). Building a new model to predict the error/residual of the previous model is used in the boosting method. Adding a new model is done until there is no more error correction that can be done, and gradient descent is used to minimize the error when creating a new model. The first tree in XGBoost is weak in classification, with the initialization probability determined by the researcher, and then the weights will be updated on each tree built so as to produce a collection of strong classification trees (Vemulapalli et al., 2024). The computational process of the XGBoot algorithm can be seen in the following figure (Mo et al., 2019):

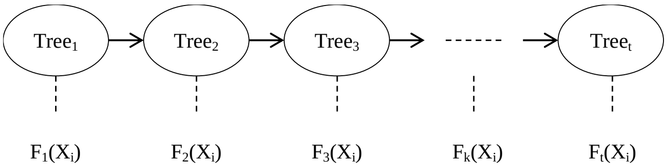

Fig. 2. XGBoost Algorithm Computing Process

The formula at step t is summed $\hat{y}_i^{(t)}$ by:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k\ (x_i) \tag{1}$$

2.3.1. Objective Function

It is necessary to define objective functions to measure how well the model fits the data training (Hanif, 2020). An important characteristic of objective function, consists of two parts, namely training loss and regularization term as in the following equation (L. Zhang & Zhan, 2017):

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{2}$$

Where L is the training loss function, Ω is the regularization term function, and θ is the parameter in the form of a related model. The regularization term controls the complexity of the model, to avoid overfitting. Training loss measures how predictive the model is in relation to data training. The training loss function in general can be written as in the following equation (Liu & Wang, 2024):

$$L(\theta) \ = \sum_{i=1}^{n} l(y_i, \hat{y}_i) \tag{3}$$

where y_i is the actual value assumed to be true, $\hat{y}_i$ is the predicted result of the related model, and is the number of iterations of the input value for the associated model.

2.3.2. Additive Training

It is difficult to learn all the trees at once instead using an additive strategy, which is to improve what has been learned by adding a new tree. Because the ensemble tree model is a function as a parameter and cannot be optimized using traditional optimization methods. So it is replaced with models trained in an additive way, by using $\hat{y}_i^{(t)}$ on the i-th prediction and t-th iteration. And in minimizing the loss function, f_t is added so that the following equation is obtained (Chen & Guestrin, 2016):

$$obj(t) = \sum_{i=1}^{t} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \ \Omega(f_t) + constant \tag{4}$$

where $\hat{y}_i^{(t)}$ is the predicted value, y_i is the actual value, , $l\left(y_i, \hat{y}_i^{(t-1)}\right)$ is the lost function, and $\Omega(f_t)$ is the regularization term. The final target of the loss function is transformed into the above equation, then trained according to the following target loss function (J. Friedman et al., 2000):

$$obj(t) = \sum_{i=1}^{t} \left[g_i f_i(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{5}$$

where:

$$g_i = \partial_{y_i^{(t-1)}} \, y_i^{(t-1)} \, l\left(y_i, y_i^{(t-1)}\right) \tag{6}$$

$$h_i = \partial^2_{y_i^{(t-1)}} \, y_i^{(t-1)} \, l\left(y_i, y_i^{(t-1)}\right) \tag{7}$$

g_i and h_i represent the first and second order gradient statistics of the loss function. An important advantage of this definition is that the value of the objective function depends only on . g_i and h_i. This is how XGBoost supports the loss function.

2.3.3. Model Complexity

The model in XGBoost consists of a set of trees and the regularization term (Ω) function is used to measure the complexity of each tree based on the number of nodes on the tree, depth, and leaf score. The regularization term can be calculated using the equation below which is used to reduce the complexity of the model and can increase its usefulness in other datasets (Chen & Guestrin, 2016).

$$\Omega(f_i) = \gamma T + \frac{1}{2}\lambda - \sum_{j=1}^{t} \omega_j^2 \tag{8}$$

where T = Number of leaves, ω = Leaf weight, γ and λ = Coefficients, with default values set to λ = 1 γ and = 0

2.3.4. The Structure Score

After reformulating the model tree, we can write objective value with t tree as the following equation (Chen & Guestrin, 2016):

$$obj(t) \approx \sum_{i=1}^{n}\left[g_w \omega_q(x_i) + \frac{1}{2} h_i \omega_q^2(x_i)\right] + \gamma T + \frac{1}{2} - \sum_{j=1}^{t} \omega_j^2 \tag{9}$$

$$= \sum_{j=1}^{t}\left[\left(\sum_{i \epsilon I_j} g_i\right) + \omega_j + \frac{1}{2}\left(\sum_{i \epsilon I_j} h_i + \lambda\right)\right] \gamma T \tag{10}$$

where g_i and h_i are the first and second order gradients of the loss function.

2.3.5. Hyperparameter Tuning

There are two types of hyperparameter optimization methods namely manual search and automatic search method. Manual search requires users to have a more professional knowledge background and practical experience.(Wu et al., 2019) To address the problem of manual approach, some common approaches to address this problem are with grid search, random search and other alternatives (Putatunda & Rama, 2018). The parameters used for hyperparameters can be seen in the following table:

Table 1. XGBoost Method Parameters

Parameters	Captions
n_estimator	Number of trees
learning_rate	Size shrinkage used to prevent overfitting
gamma	Minimum loss needed to partition nodes on a tree
colsample_bytree	The ratio of column subsamples when building each tree
max_depth	The maximum depth of the tree
min_child_weight	The minimum amount of weight (hessian) required on the child node.

source : (Shouri, 2024)

2.4. Performance Measurement of Classification Algorithm

Confusion matrix is a tabulation of calculations based on the evaluation of the performance of a classification model based on the number of correctly and incorrectly predicted research objects. This method compares the results of the classification of the system with the actual class (Koshimizu et al., 2020).

Table 2. Confusion Matrix

	Predictive Positive	Predictive Negative
Actual Positive	TP	FP
Actual Negative	FN	TN

where TP is the amount of data with a positive actual value and a positive predicted value, TN is the amount of data with a positive actual value and a negative predicted value, FP is the amount of data with a negative actual value and a positive predicted value, FN is the amount of data with a negative actual value and a negative predicted value.

The Confusion matrix table produces several components including sensitivity, specificity, recall, and accuracy (Lenny et al., 2025). Accuracy is the percentage of the number of data correctly predicted from the total data (Katoch et al., 2022). Precision is the proportion of correct positive-labeled predictions to positive predictions Recall and Specificity is the effectiveness of classifiers in identifying positive and negative labels (Luque et al., 2019). F1-Score is the average of precision and recall (Xiong et al., 2022). The area under the curve (AUC) can be interpreted as a probability, and is always in the range 0-1, a higher AUC indicates a better classification method. The formula in finding the classification performance value can be seen in the equation:

$$Acc = \frac{(TP + TN)}{(TP + TN + FP + FN)}\, x\, 100\% \tag{11}$$

$$Prec = \frac{(TP)}{(TP + FP)}\, x\, 100\% \tag{12}$$

$$Rec = \frac{(TP)}{(TP + FN)} x\ 100\% \tag{13}$$

$$Spec = \frac{(TN)}{(TN + FP)} x\ 100\% \tag{14}$$

$$F1 - Score = 2\,\mathrm{X}\,\frac{rec\ \mathrm{x}\ prec}{rec + prec} \tag{15}$$

$$AUC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{16}$$

The degree of accuracy of the classification results using AUC values is shown in the Table 3.

Table 3. Accuracy of Classification

Value AUC	Category
0.90 ≤ AUC ≤ 1.00	Classification Excellent
0.80 ≤ AUC < 0.90	Classification Good
0.70 ≤ AUC < 0.80	Classification Pretty Good
0.60 ≤ AUC < 0.70	Classification Less Good
0.50 ≤ AUC < 0.60	Bad

Source : (Gorunescu, 2011b)

2.5. Hypertension

Hypertension is a clinical syndrome with increased vascular pressure (Nour & Polat, 2020). People with hypertension generally will not feel the same symptoms as other diseases in general because hypertension is difficult to detect. It can often be detected by tests related to high blood pressure, such as stroke or diabetes (Sianipar, 2025). Hypertension can increase the risk of heart disease, nerves, kidneys, blood vessels, and getting higher blood pressure (Lindblom et al., 2024). Blood pressure is divided into 4 types, the 4 types are defined as systolic or diastolic blood pressure as can be seen in Table 4.

Table 4. Classification of Hypertension Levels

Classification	Systolic (mmHg)		Diastolic (mmHg)
Normal	80 – 119	Dan	60 – 79
Pre-Hypertension	120 – 129	Dan	< 80
Stage 1 Hypertension	130 – 139	Atau	80 – 89
Stage 2 Hypertension	≥ 140	Atau	90 >

Source: American Heart Association Task Force (Jamerson et al., 2017)

3. Research Methods

In this study, a machine learning algorithm was used, namely the Exreme Gradient Boosting (XGBoost) method. This algorithm was used to classify the 75 hypertension data of Pelamonia Makassar Hospital consisting of hypertension level status, gender, age, triglyceride levels, total cholesterol levels and patients' pulse. The stages of data analysis shown on Figure 3.

4. Results and Discussion

4.1. Preprocessing

The preprocessing stage is used to solve some problems that can interfere during data processing. This is due to a lot of data whose format is inconsistent. The first preprocessing stage performed in this study was to perform outlier data detection. In this study, outliers will be addressed with IQR and quantile based flooring/capping approaches. The IQR value determines the length of the box on the boxplot, the higher or the width of the IQR field indicates the more diffuse the data has. Data larger than the IQR value limit is considered an outlier. Because the dataset is small, the outlier data will not be deleted, but will be replaced with the maximun/minmum value of the IQR of the non outlier data. The next step is labeling the data. Labeling the data in this study was done by transforming the data on the class label by converting the data type of the label column to numeric so that it can be processed at the time of classification. This labeling is used to distinguish categories based on data variation. Finally, divide the data into two parts, namely data training and data testing. Data training is used to train algorithms for the formation of a model, and data testing is used to measure the degree of accuracy and performance obtained from data training. The total dataset is divided by a proportion of 80% for data training and 20% for data testing.

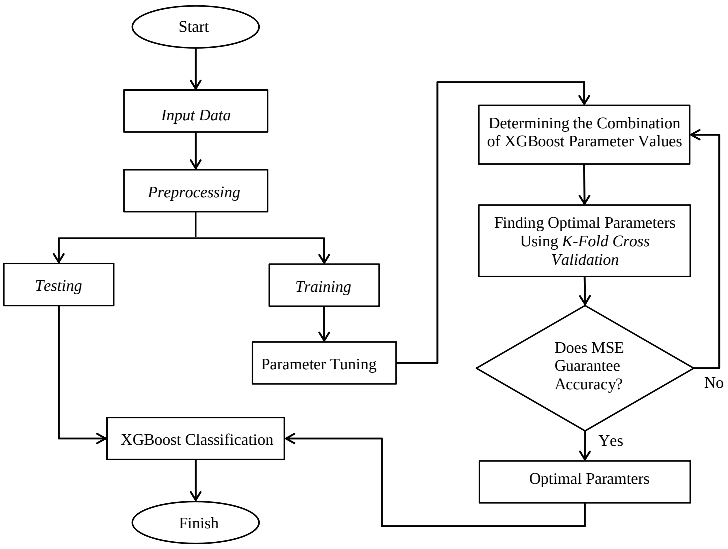
Start
Input Data
Preprocessing
Testing
Training
Parameter Tuning
XGBoost Classification
Finish
Determining the Combination of XGBoost Parameter Values
Finding Optimal Parameters Using K-Fold Cross Validation
Does MSE Guarantee Accuracy?
No
Yes
Optimal Paramters

Figure 3. Flowchart Stages of Data Analysis

4.2. XGBoost Classification

The algorithm used for classification is XGBoost. During the classification process, researchers did so by using the python programming language with the help of sklearn, matplotlib, seaborn and xgboost libraries to perform the classification with XGBoost. After separating the X and Y label sections, then, separate data training and data testing. Dataset testing is not used in modeling because it gets cross-validation error estimates from data training, whereas set data testing is used as validation for final models that match all training data.

In this study, the process of hyperparameter tuning was performed, i.e. optimization of parameters useful to improve the performance of the model in classifying. The process of hyperparameter tuning used before training the data can improve algorithm performance, particularly for classification techniques (Pavan et al., 2021). It can be said that hyperparameter tuning is the recommended thing to be one of the stages before performing the classification.

The process of hyperparameter tuning performed on the parameters in this study used the GridSearchCV method. The optimal hyperparameter configuration of gridsearch was selected based on the highest cross validation accuracy value of the hyperparameter candidate (Liu & Wang, 2024). GridSearchCV is categorized as a conscientious method, since in determining the best parameters an exploration of each parameter is carried out by setting the type of predicted value in advance. The results of the best parameter values are as follows:

Table 5. XGBoost Parameter Value Combination

Parameter	Grid Search Values	Best Parameter
n_estimator	[50, 100; 200]	50
learning_rate	[0.1; 0.2; 0.3]	0.1
Gamma	[0; 0.1; 0.3]	0.3
colsample_bytree	[0; 0.5; 1]	1
max_depth	[6; 8; 10; 12]	6
min_child_weight	[1; 2; 3]	1

Used CV = 5 to evaluate the performance of the model by five loops in the grid search process of each parameter to obtain the best parameter value. The best parameter values are used in the determination of the classification model. Data training is used to estimate parameters for a given model from a data set. After obtaining the best parameter value then adjust the series on the XGBoost model and the parameters that are not tuned are set by default. Finally perform the classification model testing using data testing. The tree and features of XGBoost impotence can be seen in the figure 3.

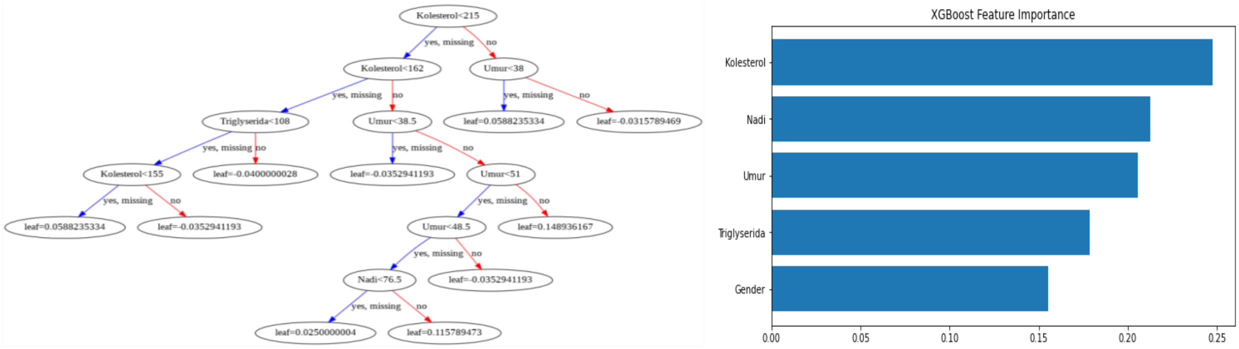

Figure 4. XGBoost tree and Plot Feature Importance

Table 6. Confusion Matrix Data Testing

Data Testing	Stage 1 Hypertension	Stage 2 Hypertension	Normal	Pre-Hypertension
Hipertensi Stadium 1	3	0	0	0
Hipertensi Stadium 2	0	2	0	1
Normal	0	0	2	0
Pre-Hypertension	0	0	0	7

Evaluation of the classification results in this study using the confucion matrix. Based on the values of the confucion matriix, the XGBoost classification also resulted in very high values judging by the precision and accuracy values of 97% and 93% respectively indicating that type 1 and type 2 prediction errors are at the minimum level in multiclass predictions. This can be considered an excellent prediction taking into account the complexity of multiclass features and categories in the data. Since this study is a multiclass classification it is very important to determine the threshold points of classifiers across multiple classes or categories. Therefore, the use of AUC values was used in this study. The results of the study show that the XGBoost algorithm has the capacity to learn between features and to identify with very high precision in carrying out the classification of the level of hypertension. It is proved at a very high AUC value of 0.935 for the multiclass classification.

Table 7. Evaluation of Classification Results

Category	Precision	Recall	Specificity	F1-Score	Accuracy	AUC
Stage 1 Hypertension	1	1	1	1	0.933	0.935
Stage 2 Hypertension	1	0.67	0.92	0.80		
Normal	1	1	1	1		
Pre-Hypertension	0.88	1	0.88	0.93		
Mean	0.97	0.92	0.95	0.93		

5. Conclusion

The application of the XGBoost method by performing hyperparameter tuning using GridSearchCV obtained the best combination of parameters, namely n_estimator = 50, learning_rate = 0.1, gamma = 0.3, colsample_bytree = 1, max_depth = 6, min_child_weight = 1, and for classification performance obtained good results. The XGBoost method produces an accuracy of 93.3%, Precision of 97%, Recall of 92%, F1-Score of 93, and an AUC value of 0.94 so that it can be interpreted that cases of hypertension classification in patients at Pelamonia Hospital Makassar can be classified well or correctly using the XGBoost method.

References

Ahmar, A. S., Meliyana, S. M., Botto-Tobar, M., & Hidayat, R. (2024). The Comparison of Single and Double Exponential Smoothing Models in Predicting Passenger Car Registrations in Canada. *Daengku: Journal of Humanities and Social Sciences Innovation*, *4*(2), 367-371. https://doi.org/10.35877/454RI.daengku2639

Chang, W., Liu, Y., Xiao, Y., Yuan, X., Xu, X., & Zhang, S. (2019). *A Machine-Learning-Based Prediction Method for Hypertension Outcomes Based on Medical Data*.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.2939785

Friedman, J. H. (2001a). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Friedman, J., Tibshirani, R., & Hastie, T. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2), 337–407. https://doi.org/10.1214/aos/1016120463

Gorunescu, F. (2011a). *Data Mining Concepts, Models and Techniques*. Springer International Publishing.

Gorunescu, F. (2011b). *Data Mining Concepts, Models and Techniques*. Springer International Publishing.

Goyal, J., Khandnor, P., & Aseri, T. C. (2021). A Comparative Analysis of Machine Learning classifiers for Dysphonia-based classification of Parkinson's Disease. *International Journal of Data Science and Analytics*, *11*(1), 69–83. https://doi.org/10.1007/s41060-020-00234-0

Hanif, I. (2020). Implementing Extreme Gradient Boosting (XGBoost) Classifier to Improve Customer Churn Prediction. *Proceedings of EAI International Conference*. https://doi.org/10.4108/eai.2-8-2019.2290338

Jamerson, K. A., Jones, D. W., MacLaughlin, E. J., Muntner, P., Ovbiagele, B., Smith, S. C., Spencer, C. C., Stafford, R. S., Taler, S. J., Thomas, R. J., Williams, K. A., Williamson, J. D., & Wright, J. T. (2017). 2017 Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults. In *American College of Cardiology*. https://doi.org/10.1016/j.jacc.2017.07.745

Katoch, S., Singh, V., & Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. *Array*, *14*, 100141. https://doi.org/10.1016/j.array.2022.100141

Koshimizu, H., Kojima, R., & Okuno, Y. (2020). Future possibilities for artificial intelligence in the practical management of hypertension. *Hypertension Research*, *43*(12), 1327–1337. https://doi.org/10.1038/s41440-020-0498-x

Lenny, L., Violyn, V., Ridwan, A., & Yennimar, Y. (2025). Comparison of C4. 5 & Random Forest Based on AdaBoost For Determining Loan Eligibility Customer Funds. *Sinkron: Jurnal Dan Penelitian Teknik Informatika*, *9*(1), 455–461.

Liew, X. Y., Hameed, N., & Clos, J. (2021). An investigation of XGBoost-based algorithm for breast cancer classification. *Machine Learning with Applications*, *6*, 100154. https://doi.org/https://doi.org/10.1016/j.mlwa.2021.100154

Lindblom, S., Ivarsson, C., Wändell, P., Bergqvist, M., Norrman, A., Eriksson, J., Lund, L., Hagströmer, M., Hasselström, J., Sandlund, C., & others. (2024). Lifestyle counseling in patients with hypertension in primary health care and its association with antihypertensive pharmacotherapy. *The Journal of Clinical Hypertension*, *26*(7), 816–824.

Liu, X., & Wang, S. (2024). Credit Card Default Prediction Based on XGBoost. *Proceedings of the 2024 International Conference on Machine Intelligence and Digital Applications*, 374–382.

López-Martínez, F., Schwarcz, A., Núñez-Valdez, E. R., & García-Díaz, V. (2018). Machine learning classification analysis for a hypertensive population as a function of several risk factors. *Expert Systems with Applications*, *110*, 206–215. https://doi.org/10.1016/j.eswa.2018.06.006

Luo, Y., Li, Y., Lu, Y., Lin, S., & Liu, X. (2018). The prediction of hypertension based on convolution neural network. *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2122–2127. https://doi.org/10.1109/CompComm.2018.8780834

Luque, A., Carrasco, A., Martín, A., & de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, *91*, 216–231. https://doi.org/10.1016/j.patcog.2019.02.023

Maalik, Si., Ananta, K. W., & Wahjuni, S. (2019). Comparation Analysis of Ensemble Technique With Boosting (Xgboost) and Bagging (Randomforest) for Classify Splice Junction Dna Sequence Category. *Jurnal Penelitian Pos Dan Informatika*, *9*(1), 27–36. https://doi.org/10.17933/jppi.2019.090

Martinez-Ríos, E., Montesinos, L., Alfaro-Ponce, M., & Pecchia, L. (2021). A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data. *Biomedical Signal Processing and Control*, *68*, 102813. https://doi.org/10.1016/j.bspc.2021.102813

Mo, H., Sun, H., Liu, J., & Wei, S. (2019). Developing window behavior models for residential buildings using XGBoost algorithm. *Energy and Buildings*, *205*, 109564. https://doi.org/10.1016/j.enbuild.2019.109564

Mroz, T., Griffin, M., Cartabuke, R., Laffin, L., Russo-Alvarez, G., Thomas, G., Smedira, N., Meese, T., Shost, M., & Habboub, G. (2024). Predicting hypertension control using machine learning. *Plos One*, *19*(3), e0299932.

Nour, M., & Polat, K. (2020). Automatic Classification of Hypertension Types Based on Personal Features by Machine Learning Algorithms. *Mathematical Problems in Engineering*, *2020*, 1–14. https://doi.org/10.1155/2020/2742781

Pavan, R., Nara, M., Gopinath, S., & Patil, N. (2021). Bayesian optimization and gradient boosting to detect phishing websites. *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, 2–6. https://doi.org/10.1109/CISS50987.2021.9400317

Putatunda, S., & Rama, K. (2018). A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost. *ACM International Conference Proceeding Series*, 6–10. https://doi.org/10.1145/3297067.3297080

Shouri, S. (2024). Validation of health insurance customers using XGBoost modeling. *Journal of International Crisis and Risk Communication Research*, *7*(S12), 1016.

Sianipar, E. A. (2025). Optimization of Diabetes Disease Classification Using Learning Vector Quantization Algorithm (LVQ). *Journal of Computer Science and Informatics Engineering*, *4*(2), 72–84.

Subasi Abdulhamit and Yaman, E. (2020). EMG Signal Classification Using Discrete Wavelet Transform and Rotation Forest. In R. and G. P. L. Badnjevic Almir and Škrbić (Ed.), *CMBEBIH 2019* (pp. 29–35). Springer International Publishing.

Sumathi, K., Kannan, S., & Nagarajan, K. (2016). Data Mining: Analysis of student database using Classification Techniques. *International Journal of Computer Applications*, *141*(8), 22–27. https://doi.org/10.5120/ijca2016909703

Syarif, I., Zaluska, E., Prugel-Bennett, A., & Wills, G. (2012). Application of bagging, boosting and stacking to intrusion detection. *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 593–602.

Tsai, C.-F., Sue, K.-L., Hu, Y.-H., & Chiu, A. (2021). Combining feature selection, instance selection, and ensemble classification techniques for improved financial distress prediction. *Journal of Business Research*, *130*, 200–209. https://doi.org/10.1016/j.jbusres.2021.03.018

Vemulapalli, L., Devendra, K. S., Satwik, G. S. S., & Kiran, T. U. (2024). Integrating SMOTE and XGBoost for enhanced Hepatitis Detection. *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, 1–6.

Wu, J., Chen, X. Y., Zhang, H., Xiong, L. D., Lei, H., & Deng, S. H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology*, *17*(1), 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120

Xiong, X., Hu, S., Sun, D., Hao, S., Li, H., & Lin, G. (2022). Detection of false data injection attack in power information physical system based on SVM–GAB algorithm. *Energy Reports*, *8*, 1156–1164. https://doi.org/10.1016/j.egyr.2022.02.290

Zhang, L., & Zhan, C. (2017). Machine Learning in Rock Facies Classification: An Application of XGBoost. *International Geophysical Conference (IGC)*, 1371–1374. https://doi.org/10.1190/igc2017-351

Zhang, Y., Miao, D., Wang, J., & Zhang, Z. (2019). A cost-sensitive three-way combination technique for ensemble learning in sentiment classification. *International Journal of Approximate Reasoning*, *105*, 85–97. https://doi.org/10.1016/j.ijar.2018.10.019