Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 3 (2025) 535 - 543 e-ISSN: 2580-0760

Health Risk Classification Using XGBoost with Bayesian Hyperparameter Optimization

Syaiful Anam[1*], Imam N. Purwanto[2], Dwi M. Mahanani[3], Feby I. Yusuf[4], Hady Rasikhun[5]
[1]Department of Mathematics, Brawijaya University, Malang, Indonesia
[2]Mining Engineering Department, Muhammadiyah University of Mataram, Mataram, Indonesia

[1]syaiful@ub.ac.id, [2]wantoinp@ub.ac.id, [3]mdwimifta@ub.ac.id, [4]febyindrianay@ub.ac.id, [5]hady.rasikhun@ummat.ac.id

Abstract

Health risk classification is important. However, health risk classification is challenging to address using conventional analytical techniques. The XGBoost algorithm offers many advantages over the traditional methods for risk classification. Hyperparameter Optimization (HO) of XGBoost is critical for maximizing the performance of the XGBoost algorithm. The manual selection of hyperparameters requires a large amount of time and computational resources. Automatic HO is needed to avoid this problem. Several studies have shown that Bayesian Optimization (BO) works better than Grid Search (GS) or Random Search (RS). Based on these problems, this study proposes health risk classification using XGBoost with Bayesian Hyperparameters Optimization. The goal of this study is to reduce the time required to select the best XGBoost hyperparameters and improve the accuracy and generalization of XGBoost performance in health risk classification. The variables used were patient demographics and medical information, including age, blood pressure, cholesterol, and lifestyle variables. The experimental results show that the proposed approach outperforms other well-known ML techniques and the XGBoost method without HO. The average accuracy, precision, recall and f1-score produced by the proposed method are 0.926, 0.920, 0.928, and 0.923, respectively. However, improvements are needed to obtain a faster and more accurate method in the future.

Keywords: health risk classification; hyperparameters; optimization; XGBoost

How to Cite: S. Anam, I. N. Purwanto, D. M. Mahanani, F. I. Yusuf, and H. Rasikhun, "Health Risk Classification Using XGBoost with Bayesian Hyperparameter Optimization", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 9, no. 3, pp. 535 - 543, Jun. 2025.
Permalink/DOI: https://doi.org/10.29207/resti.v9i3.6307

Received: January 14, 2025
Accepted: May 5, 2025
Available Online: June 13, 2025

1. Introduction

Risks frequently cause losses for people or businesses. Natural disasters and disease outbreaks as well as lifestyle ultimately have an impact on health risks. Taking out insurance may reduce or completely eliminate risk-related losses [1]. The health insurance industry helps people overcome health risks. Health risk classification will group individuals based on their health characteristics. The level of health risk can influence the possibility of disease occurring and the amount of health service costs required. This process is not only important for insurance companies in calculating premiums, but also for health service providers in designing treatment programs that suit patient needs [2]. Health risk classification for insurance companies offers various benefits including: accurate premium pricing [3], reduced adverse selection [4], improved risk management [5], increased market efficiency [6], facilitated product innovation, improved customer segmentation, and improvements in claims management.

It is challenging to address the health risk classification problem using conventional analytical methods and to represent mathematically [7], [8]. Machine Learning (ML) methods offer solutions to complex problems that are difficult to model mathematically. Traditional statistical approaches may not be sufficient to reveal underlying associations and trends, but ML could [9]. The capacity of ML for effectively handling high-dimensional data is one of its primary benefits in classification [10]. Another significant advantage of ML in classification is its adaptability and scalability. ML algorithms can be trained on large data sets and can improve their performance as more data becomes available [11].

Currently, ML methods have been widely applied to various complex problems, such as child nutritional status [12], text classification [13], stunted facial

images classification [14] and many others. ML methods have also been used in risk classification [15]-[19]. There are many ML methods, such as Extreme Gradient Boosting (XGBoost), K-Nearest Neighbor (KNN), Decision Tree (DT), Artificial Neural Networks (ANN), and Support Vector Machine (SVM), etc. Previous research has shown that XGBoost outperforms other ML models, such as KNN, DT, ANN, and SVM [18]-[20]. The benefits of XGBoost include high flexibility, excellent predictability, excellent generalization ability, high scalability, good training of models' efficiency, and remarkable resilience, according to earlier studies [21].

However, the performance of XGBoost depends on the selection of hyperparameters [22], [23]. XGBoost hyperparameter selection can be done with Hyperparameter Optimization (HO). This process is very important to maximize the performance of the XGBoost algorithm and classification accuracy. Manual HO is a widely recognized challenge in the ML field. However, this is impractical, often leading to a time-consuming and inefficient trial-and-error approach [24]. Automated HO has emerged as an important advancement in the ML field, offering several significant advantages over traditional manual tuning methods. One of the main benefits is a major reduction in the time and computational resources required for model optimization. Bayesian Optimization (BO), Grid Search (GS), and Random Search (RS) are a few known autonomous HO techniques. In ML, BO has become a potent method for HO issues. This method offers significant advantages over the RS and GS methods. GS and RS have been used to select the hyperparameters of XGBoost [25], [26]. GS evaluates all possible combinations of hyperparameters exhaustively so it requires a lot of computing time and RS is easier to handle and requires less computing power than looking at every potential combination because it is based on randomly sampling a predetermined selection of hyperparameters. The BO is more efficient and effective than RS and GS in exploring the hyperparameter space. BO builds a probabilistic model of the objective function. This model allows informed decisions regarding the location of the next point of capture, focusing on areas that may provide better results based on previous evaluations [27].

Based on the background explained above, this article proposes the development of a health risk classification method using XGBoost with Bayesian Hyperparameters Optimization (BHO). This article aims to address the weaknesses of previous research that uses GS and RS to select hyperparameters of XGBoost. By integrating XGBoost and BO, this article provides several contributions and benefits. The first contribution is to improve of the accuracy and generalization of XGBoost performance. The second contribution is to provides a good alternative method for classifying health risks, optimizing premium pricing strategies, improving customer service and improving risk management practices. The third contribution is to reduce the time to select the best XGBoost hyperparameters because the proposed method has the ability to select XGBoost hyperparameters automatically. The last contribution is to minimize the risk of bias and human error in selecting XGBoost hyperparameters.

2. Methods

This section explains the research methods used in this research. The type of research used is quantitative research. The stages of this research include: literature collection and literature review, data collection, algorithm development, testing and evaluation.

This study begins by collecting the literature and reviewing the literature or theory. The literature reviewed includes health risk classification, XGBoost, and BO methods. Previous research related to health risk classification methods, the XGBoost and BO methods will be reviewed and discussed to see research developments in this field. The results of this literature review provided an understanding of this research. Furthermore, the outcome of this study is a comprehension of the individual features of each approach, including its advantages and disadvantages.

The dataset used in this research is the Cardio Health Risk Assessment Dataset. This dataset was taken from the web at the address https://www.kaggle.com/datasets/kapoorprakhar/cardio-health-risk-assessment-dataset/data. This website provides the cardio health risk assessment dataset that contains comprehensive patient demographics and medical information, including age, blood pressure, cholesterol, and lifestyle variables.

The next step is to develop a health risk classification method algorithm using the XGBoost method with BHO. The result of the algorithm development stage is a risk classification method algorithm that is ready to be implemented into a programming language. The implementation of the XGBoost-based health risk classification method program with BHO at this stage was carried out using Jupyter Notebook. Data preprocessing methods are also used at this stage to ensure that the data is ready for further processing. The selection of features that will be used for health risk classification is also carried out at this stage to obtain relevant features so that the classification method can work optimally. Implementation of the proposed method uses Python software. The resulting program was evaluated and tested using the dataset obtained in the second stage of this research process. Determining the appropriate parameters for the proposed method is carried out at this stage. The parameters in question are parameters in the BO method so that it produces good performance in the XGBoost method. The success of the program is evaluated by comparing the program's predicted results with real data that has been labeled.

The tools to measure the success of the proposed method are accuracy, recall and precision and f_1 score.

Classification level/accuracy is calculated using Equation 1.

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

The number of cases successfully categorized in the positive class is denoted by TP. The number of cases that were accurately assigned to the negative class is shown by TN. The number of cases in the negative class that were misclassified is denoted by FN. The number of cases in the positive class that were misclassified is denoted by FP. The ratio of accurate forecasts to the total number of events assessed is known as accuracy. Equation 2 is used for calculating recall. The number of successfully categorized positive instances divided by the total number of positive class occurrences is known as recall.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Equation 3 is used for calculating precision. The precision metric quantifies the proportion of precisely predicted instances that are positive to occurrences identified as positive classes.

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

The harmonic average of recall and precision is used for calculating the f_1-score. The results of the testing and evaluation phase are numerical results and graphical visualization of the risk classification and performance analysis methods obtained from testing. The significant increase in method performance for all evaluation measures when compared to XGBoost without HO and a number of other well-known ML techniques demonstrates the effectiveness of the suggested approach.

The outcomes of the experiments conducted in the previous phase served as the foundation for the results analysis. The results of this analysis are the basis for drawing conclusions. The results of this stage are conclusions and suggestions for improvements that can be carried out in further research.

The flowchart of the health risk classification method by using XGBoost with BHO is shown in Figure 1. The proposed method consists of several stages. The first stage is to enter the dataset, BO parameters and determine the range of XGBoost hyperparameters. The next step is to preprocess the data and divide the dataset into two sub datasets, namely training data and test data. The next step is that a risk classification method using XGBoost with BHO is created, evaluated and implemented as shown in Figure 1.

3.1 Enter the Dataset and determine the range of XGBoost hyperparameters

This article uses the heart health risk data set from Kaggle.com at https://www.kaggle.com/datasets/kapoorprakhar/cardio-health-risk-assessment-dataset/data. *Age, sex, type of chest pain, bloop pressure, cholesterol, FBS above 120, ECG findings, maximal heart rate, exercise angina, ST depression, slope of ST, number of fluro vessels*, and *Thallium* are the characteristics that are employed.

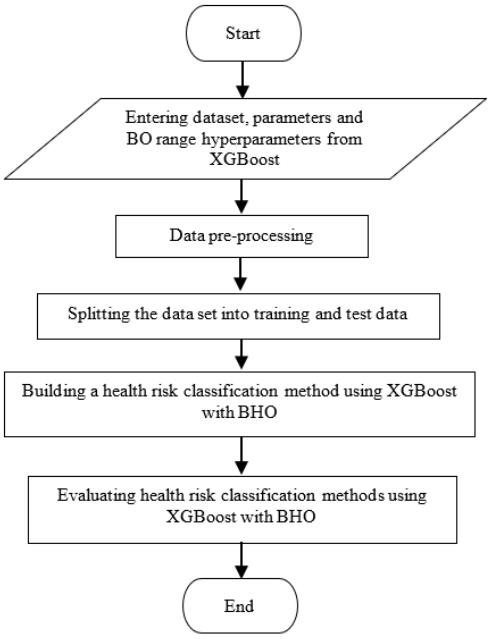

Figure 1. Flowchart of the health risk classification method using XGBBoost with BHO

Table 1. Explanation of each feature

Features / Variable	Description
Age (Ag)	Patient age
Sex (Sx)	Patient's Gender The number 0 is used as a code to represent female patients, and the number 1 is for male.
Chest pain type (CPT)	CPT defines the types of chest pain. It has values of 1-4.
BP	The value of BP defines Blood pressure.
Cholesterol (Chol)	Chol represents the cholesterol level.
FBS over 120 (FBS)	Fasting Blood Sugar > 120 mg/dl (0 = False, 1 = True) refers to a method of categorizing blood sugar levels.
EKG results (EKG)	Electrocardiography (ECG) Test Results (Value 0-2) refers to a classification system used to interpret the results of an ECG test, which records the electrical activity of the heart.
Max HR (MxHR)	Maximum heart rate is reached
Exercise angina (EA)	Exercise-Induced Angina (0 = No, 1 = Yes) refers to the presence or absence of chest pain or discomfort triggered by physical activity or exercise.
ST depression (STD)	ST Depression Caused by Exercise Rather Than Rest refers to a specific finding in an electrocardiogram (ECG) that indicates a possible heart condition
Slope of ST (SST)	Peak exercise ST segment slope (values 1-3)
Number of vessels fluro (NVF)	Number of large blood vessels stained by fluoroscopy (value 0-3)
Thallium (Thal)	Thallium stress test results (values 3, 6, 7)
Heart Disease (HD)	Whether or not there is a risk of heart disease (Yes, None)

The amount of data used was 270 data. The target is the level of heart disease risk. An explanation of each feature can be seen in Table 1. Examples of the dataset can be seen in Table 2.

The range of XGBoost hyperparameters could be seen Table 3. The XGBoost hyperparameters are the learning rate, the maximum depth, the number of estimators, and the subsample. The learning rate in XGBoost determines the step size at each boosting iteration, where smaller values slow down the learning process and often require more trees for good performance, but can lead to better generalization. The maximum depth parameter controls the maximum depth of each individual decision tree, where deeper trees can capture more complex patterns but increase the risk of overfitting. The number of estimators specifies the total number of boosting rounds in the model, and while more estimators can improve performance, they also increase computation time and may cause overfitting without proper regularization or early stopping. The subsample parameter sets the fraction of the training data to be randomly sampled for growing each tree, which introduces randomness and helps prevent overfitting, especially when set to a value less than 1.0.

Table 2. Example of dataset

No	*Ag*	*Sx*	*CPT*	*BP*	*Chol*	*FBS*	*EKG*	*Mx HR*	*EA*	*STD*	*SST*	*NVF*	*Thal*	*HD*
1	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
2	80	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
3	55	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
4	65	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
5	45	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence

Table 3. XGboost hyperparameters range

Hyperparameters	Range
Learning rate	{0.01, 0.011, 0.012, ... ,0.3}
Max depth	{3,4, 5, ...,10}
Number of estimators	{100,101,102,...,1000}
subsample	{0.5,0.501, 0.502,...,1.0}

3.2. Data preprocessing

The results of examining the dataset show that the dataset does not contain empty data so there is no need to handle empty data. The labels for the risk classes are string type so that the HD feature is labeled encoded to transform presence into 1 and absence into 0. Next, the data is subjected to Min-Max normalization techniques because the features in the dataset have different ranges. The Min-Max normalization technique transforms the values in a dataset into a certain range, usually between 0 and 1, without changing the relative distribution of those values. This technique is useful so that all features in the dataset have the same scale. The equation for the Min-Max normalization technique for a value of x can be seen in Equation 4.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4)$$

x' defines the normalized value and x defines the original value. x_{min} and x_{max} represent the minimum value and the maximum value for a feature in the dataset, respectively. The dataset after applying the Min-Max normalization technique could be seen Table 4. It can be shown that all features have scale between 0 and 1.

Subsequently, the study data is separated into two groups: 30% for testing data and 70% for training data. Next, the amount of data in each risk class is calculated to see whether the data class is balanced or unbalanced. Figure 2 shows that the data is unbalanced ie. 55.56% for the non-risk class and 44.44% for the risk class. In order to balance the training data, this study used the Synthetic Minority Over-sampling Technique for Nominal and Continuous (SMOTE-NC).

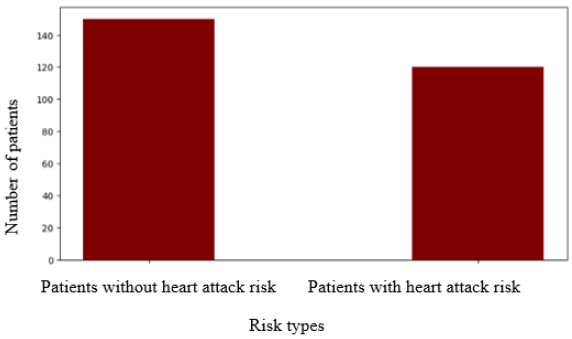

Figure 2. Bar chart of data distribution in each risk class

Designed to handle datasets with mixed features, namely continuous and categorical characteristics, SMOTE-NC is a variation of the SMOTE (Synthetic Minority Over-sampling approach) approach. By using this method, the issue of class imbalance in a dataset, where the minority class has much less samples than the majority class, is resolved. How SMOTE-NC works could be defined into several steps. The first step in the SMOTE-NC method is to identify minority classes. SMOTE-NC first identifies minority classes in the dataset. The second step in the SMOTE-NC method is to choose the nearest neighbor. SMOTE-NC uses the k-nearest neighbors (k-NN) algorithm to find several nearest neighbors in the minority class for each sample in the minority class. The third step in the SMOTE-NC method is create synthetic samples. The way of synthetic samples is calculated by feature type. For continuous features, a synthetic sample is produced by linearly interpolating between the original sample and one of its neighbors. The equation for $x_{\text{synthetic}}$ continuous feature is represented by using Equation 5,

$$\boldsymbol{x}_{\text{synthetic}} = \boldsymbol{x}_{\text{original}} + \lambda(\boldsymbol{x}_{\text{neighbor}} - \boldsymbol{x}_{\text{synthetic}}) \quad (5)$$

where λ is a random number between 0 and 1. For categorical features, a synthetic sample's value is chosen from the most prevalent value (mode) for the categorical characteristic among the original sample and its neighbors. The last step is to set aside synthetic samples. The number of samples in the minority class is then increased by adding the generated synthetic samples to the dataset.

Table 4. Example of dataset after applying Min-Max Normalization

No	Ag	Sx	CPT	BP	Chol	FBS	EKG	Mx HR	EA	STD	SST	NVF	Thal	HD
1	0.80	1.0	1.00	0.34	0.45	0.00	1.00	0.29	0.00	0.39	0.50	1.00	0.00	1
2	1.00	0.0	0.67	0.20	1.00	0.00	1.00	0.68	0.00	0.26	0.50	0.00	1.00	0
3	0.51	1.0	0.33	0.28	0.31	0.00	0.00	0.53	0.00	0.05	0.00	0.00	1.00	1
4	0.71	1.0	1.00	0.32	0.31	0.00	0.00	0.26	1.00	0.03	0.50	0.33	1.00	0
5	0.31	0.0	0.33	0.25	0.33	0.00	1.00	0.38	1.00	0.03	0.00	0.33	0.00	0

The training data after SMOTE-NC is distributed seems to have a balanced distribution, this can be seen in Figure 3.

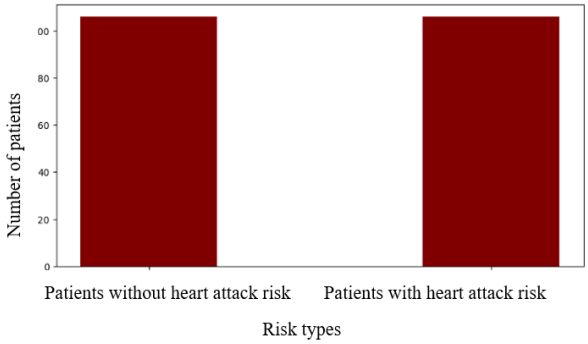

Figure 3. Bar chart of training data distribution after SMOTE-NC

3.3. Building a Risk Classification Model

The next stage after data preprocessing is to build a health risk classification model using XGBoost with BHO. BO is used because it offers a powerful and efficient method for optimizing complex functions. Compared to conventional optimization techniques, the BO approach is intended to determine a function's global optimum with fewer evaluations. BO can find global optima from an objective function with numerous optima, unlike local optimization techniques that might reach convergence to a local minimum [28]. This is especially useful in scenarios where function evaluation is expensive, such as hyperparameter tuning in ML models [29]. BO has an effective balance of exploration and exploitation [30] Because BO effectively searches the parameter space and concentrates on promising locations, it can drastically cut down on the number of function evaluations needed to discover the best solution [31]. Therefore, BO is very effective in using hyperparameter optimization in ML models such as XGBoost. XGBoost has many hyperparameters that can affect model performance, and finding the optimal combination of these hyperparameters can be a complex and time-consuming task. BO in optimizing hyperparameters in XGBoost could be divided in several steps. The first step is to define the objective function. The second step is to select the hyperparameters to be optimized. The last step is to use BO to optimize hyperparameters. The objective function in this context is a model evaluation metric, accuracy is used as the objective function in this research. The pseudocode of the objective function can be seen in Pseudocode 1. The hyperparameters optimized in XGBoost can be seen in Table 4.

Pseudocode 1. Objective function of XGboost hyperparameter optimization using BO.
function objective ($params$):

$Dataset = [(X_{train}, y_{train}), (X_{test}, y_{test})]$

Build XGBClassifier by using params. $params = \{learningRate, MaxDept, NumberEstimator, subsample\}$ and *Dataset.*

Calculate predts by evaluation X_{test} on model.

Calculate the accuracy of data testing.

Return the accuracy of data testing.

An explanation of how the XGBoost Classifier works can be stated in Algorithm 1.

Algorithm 1. XGBoost Algorithm.

Input training data $D = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, objective function ($Obj$), learning rate η, the number of trees or the number of Estimator T, regularization parameters (λ, γ), the maximum depth of trees and subsample ratio

Initialize model with $F_0(\boldsymbol{x}) = 0$ (or a constant prediction)

For $t = 1$ to T (number of trees) do

Calculating the gradient and hessian for the objective function by using Equations 6 and 7, respectively.

$$g_i = \partial Obj(y_i, F(\boldsymbol{x}_i)) \,/\, \partial F(\boldsymbol{x}_i) \quad (6)$$

$$h_i = \partial^2 Obj(y_i, F(\boldsymbol{x}_i)) \,/\, \partial F(\boldsymbol{x}_i)^2 \quad (7)$$

Construct a new tree (T_t) to minimize the objective function.

Start with one node (root).

For each leaf node do

Calculate the gain for all possible feature splits.

Select the split with the highest gain.

Add a split to the tree if obtained > γ (regularization term).

Continue until reaching max_depth or gain < γ for further split.

Prune the tree: remove separations where the set gain is negative.

Calculate the output value for each leaf node using Equation 8,

$$w_j = -\sum g_i / (\sum g_i + \lambda) \quad (8)$$

where w_j is the weight of leaf j.

Update the model by using Equation 9.

$$F_t(\boldsymbol{x}) = F_{t-1}(\boldsymbol{x}) + \eta * T_t(\boldsymbol{x}) \quad (9)$$

Output: Final model $F_T(\boldsymbol{x})$

A new tree is constructed using each data set's gradient and Hessian. By maximizing the gain, which is determined by the model complexity penalty (from the hessian) and the error reduction (from the gradient), the tree is constructed. Once the tree is built, branches with negative gain are removed to avoid overfitting. The model is updated by adding new trees that have been multiplied by the learning rate. Gradient and Hessian are used for faster and more stable optimization. The parameters λ and γ are useful for avoiding overfitting by adding a penalty to the model complexity. Tree Pruning is done by removing branches with negative gain to simplify the model and prevent overfitting. Learning Rate is used to control the size of the contribution of each new tree added to the ensemble model.

3. Results and Discussions

This section will discuss the experimental results and analyze the experimental results. Repeating experiments for the same parameters is very important to be conducted because the solutions produced the model of the risk classification in the health insurance by using XGBoost with BHO can vary from one experiment to another Repetition of experiment helps ensure that the results are consistent and not due to random chance or anomalies in the data. Repetition of experiment also helps in identifying and reducing errors or biases. It ensures that the findings are robust and not influenced by one-time mistakes or external factors. Repeating the experiment under the same conditions ensures that the results can be generalized to other similar scenarios, increasing the study's external validity. By repeating experiments, statistical measures such as the mean, variance, and standard deviation could be calculated. It will help in determining the reliability and significance of the results. This research was repeated for each experiment 25 times.

The distribution of experimental results produced is calculated using the average and standard deviation of accuracy, recall, precision and f_1 score, as well as processing time. The average of accuracy, recall, precision and f_1 score produced by the model of the risk classification in the health insurance using XGBoost with BHO and comparison methods are presented in Tables 5, 6, 7 and 8. Average of accuracy value of training data and testing data produced by the model of the risk classification in the health insurance using XGBoost with BHO is 1.000, 0.926, respectively. From Table 5, it also can be shown that the model of the risk classification in the health insurance using XGBoost with BHO give higher accuracy than all methods, such as ADABOOST, KNN, Linear Logistic, Multilayer Perceptron, Naïve Bayes Classifier Random Forest Classifier, SVM, DT, Histogram Gradient Boosting and XGBoost for training data and testing data. The other finding from Table 5 is that the ADABOOST, Random Forest Classifier, DT, XGBoost and Histogram Gradient Boosting achieve perfect accuracy on training data but decrease significantly on test data. This means that these methods experience overfitting. It is really different with the model of the risk classification in the health insurance using XGBoost with BHO, it is able to reduce the difference in accuracy between training and testing data. It means that the proposed methods significantly reduce the overfitting problem in risk classification in health insurance.

Table 5. Average of accuracy produced by using XGBoost with BHO and other methods for classifying Health insurance risks

Method	Training	Test
ADABOOST	0.972	0.778
KNN	0.873	0.827
Linear Logistic	0.854	0.852
Multilayer Perceptron	0.876	0.836
Naïve Bayes Classifier	0.863	0.864
Random Forest Classifier	1.000	0.800
SVM	0.882	0.827
DT	1.000	0.734
Histogram Gradient Boosting	1.000	0.741
XGBoost	1.000	0.790
XGBoost with BHO	**1.000**	**0.926**

Table 6. Average of precision produced by using XGBoost with BHO and other methods for classifying Health insurance risks

Method	Training	Test
ADABOOST	0.972	0.779
KNN	0.873	0.827
Linear Logistic	0.854	0.853
Multilayer Perceptron	0.877	0.836
Naïve Bayes Classifier	0.864	0.867
Random Forest Classifier	1.000	0.803
SVM	0.882	0.827
DT	1.000	0.734
Histogram Gradient Boosting	1.000	0.742
XGBoost	1.000	0.794
XGBoost with BHO	**1.000**	**0.920**

Table 7. Average of recall produced by using XGBoost with BHO and other methods for classifying Health insurance risks

Method	Training	Test
ADABOOST	0.972	0.772
KNN	0.873	0.824
Linear Logistic	0.854	0.849
Multilayer Perceptron	0.876	0.834
Naïve Bayes Classifier	0.863	0.860
Random Forest Classifier	1.000	0.794
SVM	0.882	0.824
DT	1.000	0.729
Histogram Gradient Boosting	1.000	0.733
XGBoost	1.000	0.783
XGBoost with BHO	**1.000**	**0.928**

From Tables 6, 7 and 8, it can be shown that the model of the risk classification in the health insurance using XGBoost with BHO also shows superiority in precision, recall, f_1 score compared to others methods. It gives an average of precision on test data of 0.920, an average of recall on test data of 0.928, and an average of f_1 score on test data of 0.923. The average of precision, recall and f_1 score produced by proposed method on training data give the perfect values of 1. Similar with accuracy, the average of recall, precision and f_1 score produced by the other methods, such as Random Forest Classifier, DT, XGBoost and Histogram Gradient Boosting are very varying, it shows large drops from training data to test data. These results indicate that the ADABOOST, Random Forest Classifier, DT, XGBoost and Histogram Gradient Boosting experience overfitting problem. From Tables 6, 7 and 8, it also can be shown that the model of the risk classification in the health insurance using XGBoost with BHO is able to reduce the difference in accuracy between training and testing data. This fact also show that the proposed methods significantly reduce the overfitting problem in risk classification in health insurance.

Table 8. Average of f_1-score produced by using XGboost with BHO and other methods for classifying Health insurance risks

Method	Training	Testing
ADABOOST	0.972	0.774
KNN	0.873	0.825
Linear Logistic	0.854	0.850
Multilayer Perceptron	0.876	0.835
Naïve Bayes Classifier	0.863	0.862
Random Forest Classifier	1.000	0.796
SVM	0.882	0.825
DT	1.000	0.730
Histogram Gradient Boosting	1.000	0.735
XGBoost	1.000	0.785
XGBoost with BHO	**1.000**	**0.923**

Table 9. Standard deviation of accuracy produced by using XGboost with BHO and other methods for classifying Health insurance risks

Method	Training	Testing
ADABOOST	0.000	0.000
KNN	0.000	0.000
Linear Logistic	0.000	0.000
Multilayer Perceptron	0.009	0.009
Naïve Bayes Classifier	0.000	0.000
Random Forest Classifier	0.000	0.009
SVM	0.000	0.000
DT	0.000	0.022
Histogram Gradient Boosting	0.000	0.000
XGBoost	0.000	0.000
XGBoost with BHO	0.000	0.000

While Tables 9, 10, 11 and 12, they show the standard deviation of accuracy, precision, recall and f_1 score produced by using XGboost with BHO and other methods for classifying Health insurance risks. These results show that all methods give the small values of standard deviation, both for training and testing data. The small standard deviation implies that the performance metrics (accuracy, precision, recall, f_1 score) are not fluctuating significantly between different runs. This indicates that the models produce stable and reliable results. For health insurance risk classification, consistent performance is crucial as it ensures reliability in risk predictions, which can directly impact decision-making and policy formulation. The small standard deviations also provide confidence that the model will perform reliably in real-world applications, where stable predictions are critical.

Table 10. Standard deviation of precision produced by using XGboost with BHO and other methods for classifying Health insurance risks

Method	Training	Testing
ADABOOST	0.000	0.000
KNN	0.000	0.000
Linear Logistic	0.000	0.000
Multilayer Perceptron	0.009	0.010
Naïve Bayes Classifier	0.000	0.000
Random Forest Classifier	0.000	0.009
SVM	0.000	0.000
DT	0.000	0.022
Histogram Gradient Boosting	0.000	0.000
XGBoost	0.000	0.000
XGBoost with BHO	0.000	0.000

Table 11. Standard deviation of recall produced by using XGboost with BHO and other methods for classifying Health insurance risks

Method	Training	Testing
ADABOOST	0.000	0.000
KNN	0.000	0.000
Linear Logistic	0.000	0.000
Multilayer Perceptron	0.009	0.009
Naïve Bayes Classifier	0.000	0.000
Random Forest Classifier	0.000	0.009
SVM	0.000	0.000
DT	0.000	0.022
Histogram Gradient Boosting	0.000	0.000
XGBoost	0.000	0.000
XGBoost with BHO	0.000	0.000

Table 12. Standard deviation of f_1-score produced by using XGboost with BHO and other methods for classifying Health insurance risks

Method	Training	Testing
ADABOOST	0.000	0.000
KNN	0.000	0.000
Linear Logistic	0.000	0.000
Multilayer Perceptron	0.009	0.009
Naïve Bayes Classifier	0.000	0.000
Random Forest Classifier	0.000	0.009
SVM	0.000	0.000
DT	0.000	0.022
Histogram Gradient Boosting	0.000	0.000
XGBoost	0.000	0.000
XGBoost with BHO	0.000	0.000

Table 13. Computation time produced by using XGboost with BHO and other methods for classifying Health insurance risks

Method	Training	Testing
ADABOOST	0.0688	0.0032
KNN	0.0010	0.00149
Linear Logistic	0.0033	0.00243
Multilayer Perceptron	0.1663	0.03817
Naïve Bayes Classifier	0.0011	0.00068
Random Forest Classifier	0.1240	0.01558
SVM	0.0025	0.00160
DT	0.0012	0.00035
Histogram Gradient Boosting	0.0563	0.00970
XGBoost	0.0257	0.00146
XGBoost with BHO	35.3158	0.71254

Computation times for XGBoost with BHO and comparison methods can be shown in Table 13. Although, XGBoost with BHO for classifying health

risks requires less computing time is higher than all methods, but XGBoost with BHO gives the significant superior performance in key metrics like accuracy, precision, recall, and f_1 score. The best performance in all these metrics suggests that the additional computational cost leads to significantly better classification results, which is often critical in high-stakes fields like health insurance risk assessment. While XGBoost with BHO demands more computational resources, this may be acceptable or even preferable in scenarios where accuracy and reliability are more critical than processing time. If the method's superior performance can potentially reduce costs or risks in the decision-making process (e.g., better risk classification leading to fewer claims or more appropriate premium settings), the higher computation time may be a worthwhile investment.

4. Conclusions

Based on the experimental results and analysis of experimental results, several conclusions were obtained, including that the Risk Classification Model in Health Insurance using XGBoost with BHO was successfully developed. BO is able to optimize the performance of the XGBoost method. Evaluation results show that the performance of the Risk Classification Model in Health Insurance using XGBoost with BHO is better than the ADABOOST, KNN, Linear Logistic, Multilayer Perceptron, Naïve Bayes Classifier, Random Forest Classifier, SVM, Decision Tree, Histogram Gradient Boosting and Standard XGboost for evaluation metrics accuracy, precision, recall, f_1 score, but the computing time is quite large. All average of accuracy, precision, recall and f_1-score produced by proposed method for training data are 1, while the average of accuracy, precision, recall and f_1-score produced by proposed method for testing data are 0.926, 0.920, 0.928 and 0.923, respectively. Although XGBoost with BHO requires more computing time compared to other methods, this trade-off is justified by its superior performance in key metrics like accuracy, precision, recall, and f_1 score. The best performance in all these metrics suggests that the additional computational cost leads to significantly better classification results, which is often critical in high-stakes fields like health insurance risk assessment. The method's superior performance can potentially reduce costs or risks in the decision-making process. The better risk classification leads to fewer claims or more appropriate premium settings. However, the improvement is still needed for obtaining the faster and more accurate method in the future. The decreasing of accuracy of the proposed method may be caused by the complex space searches, so BO gets trapped in local optima. To ovoid this problem, Swarm Intelligence could be considered in future research to solve this problem since Swarm Intelligence has better ability to escape local optima. Some studies also show that Swarm Intelligence is able to find the nearest optimal point with acceptable computational time.

Acknowledgements

The author would like to thank FMIPA Universitas Brawijaya that has supported this research through an Internal Research Grant with contract number 2612.7/UN01.F09/PN/2024.

References

[1] M. Sigala, A. Beer, L. Hodgson, and A. O'Connor, "Big Data for Measuring the Impact of Tourism Economic Development Programmes: A Process and Quality Criteria Framework for Using Big Data," in *Big Data and Innovation in Tourism, Travel, and Hospitality: Managerial Approaches, Techniques, and Applications*, M. Sigala, R. Rahimi, and M. Thelwall, Eds., Singapore: Springer Singapore, 2019, pp. 57–73. doi: 10.1007/978-981-13-6339-9_4.

[2] G. Nguyen *et al.*, "Machine Learning and Deep Learning Frameworks and Libraries for Large-Scale Data Mining: A Survey," 2019. doi: 10.1007/s10462-018-09679-z.

[3] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, p. 60, 2019, doi: 10.1186/s40537-019-0197-0.

[4] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[5] S. A. Thamrin, D. S. Arsyad, H. Kuswanto, A. Lawi, and S. Nasir, "Predicting Obesity in Adults Using Machine Learning Techniques: An Analysis of Indonesian Basic Health Research 2018," 2021. doi: 10.3389/fnut.2021.669155.

[6] A. D. Dwivedi, G. Srivastava, and R. Dhar Shalini and Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors (Basel)*, vol. 19, no. 2, p. 326, Jan. 2019.

[7] F. Al-Turjman, H. Zahmatkesh, and L. Mostarda, "Quantifying Uncertainty in Internet of Medical Things and Big-Data Services Using Intelligence and Deep Learning," *IEEE Access*, vol. 7, pp. 115749–115759, 2019, doi: 10.1109/ACCESS.2019.2931637.

[8] S. Kumar and M. Singh, "Big data analytics for healthcare industry: impact, applications, and tools," *Big Data Mining and Analytics*, vol. 2, no. 1, pp. 48–57, 2019, doi: 10.26599/BDMA.2018.9020031.

[9] L.-M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, "Deployment of IoV for Smart Cities: Applications, Architecture, and Challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2019, doi: 10.1109/ACCESS.2018.2887076.

[10] B. P. L. Lau *et al.*, "A survey of data fusion in smart city applications," *Information Fusion*, vol. 52, pp. 357–374, 2019, doi: https://doi.org/10.1016/j.inffus.2019.05.004.

[11] Y. Wu *et al.*, "Large Scale Incremental Learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 374–382. doi: 10.1109/CVPR.2019.00046.

[12] M. Yanto, F. Hadi, and S. Arlis, "Optimization of machine learning classification analysis of malnutrition cases in children," *J. RESTI (Rekayasa Sist. Dan Teknol. Inf.)*, vol. 7, no. 6, pp. 1378–1386, Dec. 2023.

[13] W. K. Sari, D. P. Rini, R. F. Malik, and Iman Saladin B. Azhar, "Multilabel text classification in news articles using long-Term Memory with Word2Vec," *J. RESTI (Rekayasa Sist. Dan Teknol. Inf.)*, vol. 4, no. 2, pp. 276–285, Apr. 2020.

[14] Y. Yunidar, Y. Yusni, N. Nasaruddin, and F. Arnia, "CNN performance improvement for classifying stunted facial images using early stopping approach," *J. RESTI (Rekayasa Sist. Dan Teknol. Inf.)*, vol. 9, no. 1, pp. 62–68, Jan. 2025.

[15] A. Mosavi, S. Shamshirband, E. Salwana, K. Chau, and J. H. M. Tah, "Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning," *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 482–492, Jan. 2019, doi: 10.1080/19942060.2019.1613448.

[16] V. Palanisamy and R. Thirunavukarasu, "Implications of big data analytics in developing healthcare frameworks – A review," *Journal of King Saud University - Computer and Information Sciences*, vol. 31, no. 4, pp. 415–425, 2019, doi: https://doi.org/10.1016/j.jksuci.2017.12.007.

[17] J. Sadowski, "When data is capital: Datafication, accumulation, and extraction," *Big Data Soc*, vol. 6, no. 1, p. 2053951718820549, Jan. 2019, doi: 10.1177/2053951718820549.

[18] J. R. Saura, B. R. Herráez, and A. Reyes-Menendez, "Comparing a Traditional Approach for Financial Brand Communication Analysis with a Big Data Analytics Technique," *IEEE Access*, vol. 7, pp. 37100–37108, 2019, doi: 10.1109/ACCESS.2019.2905301.

[19] D. Nallaperuma *et al.*, "Online Incremental Machine Learning Platform for Big Data-Driven Smart Traffic Management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019, doi: 10.1109/TITS.2019.2924883.

[20] S. Schulz, M. Becker, M. R. Groseclose, S. Schadt, and C. Hopf, "Advanced MALDI mass spectrometry imaging in pharmaceutical research and drug development," *Curr Opin Biotechnol*, vol. 55, pp. 51–59, 2019, doi: https://doi.org/10.1016/j.copbio.2018.08.003.

[21] P. Zhang, Y. Jia, and Y. Shang, "Research and Application of XGBoost in Imbalanced Data," 2022. doi: 10.1177/15501329221106935.

[22] T.-F. Lee *et al.*, "Development of a Risk Prediction Model for Radiation Dermatitis Following Proton Radiotherapy in Head and Neck Cancer Using Ensemble Machine Learning," 2024. doi: 10.1186/s13014-024-02470-1.

[23] I. Barkiah and Y. Sari, "Overcoming Overfitting Challenges with HOG Feature Extraction and XGBoost-Based Classification for Concrete Crack Monitoring," 2023. doi: 10.24425/ijet.2023.146509.

[24] F. H. Garabaghi, S. Benzer, and R. Benzer, "Sequential GP-UCB Bayesian Optimization for Deep Neural Network Fine-Tuning in Dissolved Oxygen Prediction," Feb. 16, 2024. doi: 10.21203/rs.3.rs-3930680/v1.

[25] J. Wang, W. Rong, Z. Zhang, and D. Mei, "Credit Debt Default Risk Assessment Based on the XGBoost Algorithm: An Empirical Study from China," 2022. doi: 10.1155/2022/8005493.

[26] A. Dimas and S. S. Mukti, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," 2021. doi: 10.22266/ijies2021.1231.19.

[27] L. Gao and Y. Ding, "Disease prediction via Bayesian hyperparameter optimization and ensemble learning," *BMC Res Notes*, vol. 13, no. 1, p. 205, 2020, doi: 10.1186/s13104-020-05050-0.

[28] M. Liang *et al.*, "Improving Genomic Prediction with Machine Learning Incorporating TPE for Hyperparameters Optimization," *Biology (Basel)*, vol. 11, no. 11, 2022, doi: 10.3390/biology11111647.

[29] P. I. Frazier, "A Tutorial on Bayesian Optimization," 2018. [Online]. Available: https://arxiv.org/abs/1807.02811

[30] F. Archetti and A. Candelieri, *Bayesian Optimization and Data Science*. 2019. doi: 10.1007/978-3-030-24494-1.

[31] Y. Chen *et al.*, "Closed-Loop Dynamic Blending Optimization Based on Variational Bayesian and its Application in Industry," *IEEE Access*, vol. 11, pp. 494–505, 2023, doi: 10.1109/ACCESS.2022.3232812.