Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 3, September 2024, pp. 520-533
ISSN: 2338-3070, DOI: 10.26555/jiteki.v%vi%i.29479

The Effectiveness of Data Imputations on Myocardial Infarction Complication Classification Using Machine Learning Approach with Hyperparameter Tuning

Muhammad Itqan Mazdadi, Triando Hamonangan Saragih, Irwan Budiman, Andi Farmadi, Ahmad Tajali
Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Indonesia

ARTICLE INFO

Article history:

Received July 17, 2024
Revised August 14, 2024
Published September 04, 2024

Keywords:

Myocardial Infarction;
Machie Learning;
Classification;
Data Imputation;
Bayesian Optimization

ABSTRACT

Complications from Myocardial Infarction (MI) represent a critical medical emergency caused by the blockage of blood flow to the heart muscle, primarily due to a blood clot in a coronary artery narrowed by atherosclerotic plaque. Diagnosing MI involves physical examination, electrocardiogram (ECG) evaluation, blood sample analysis for specific heart enzyme levels, and imaging techniques such as coronary angiography. Proactively predicting acute myocardial complications can mitigate adverse outcomes, and this study focuses on early prediction using classification methods. Machine learning algorithms such as Support Vector Machine (SVM), Random Forest, and XGBoost were employed to classify patient medical records accurately. Techniques like K-Nearest Neighbors (KNN) imputation, Iterative imputation, and Miss Forest were used to handle incomplete datasets, preserving vital information. Hyperparameter optimization, crucial for model performance, was performed using Bayesian Optimization, which minimizes the objective function by modeling past evaluations. The contribution to this study is to see how much influence data imputation has on classification using machine learning methods on missing data and to see how much influence the optimization method has when performing hyperparameter tuning. Results demonstrated that the Iterative Imputation method yielded excellent performance with SVM and XGBoost algorithms. SVM achieved 100% accuracy, precision, sensitivity, F1 score, and AUC. XGBoost reached 99.4% accuracy, 100% precision, 79.6% sensitivity, an F1 score of 88.7%, and an AUC of 0.898. KNN Imputation with SVM showed results similar to Iterative Imputation with SVM, while Random Forest exhibited poor classification outcomes due to data imbalance, causing overfitting.

Corresponding Author:

Triando Hamonangan Saragih, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Indonesia
Email: triando.saragih@ulm.ac.id

1. INTRODUCTION

Complications from Myocardial Infarction (MI) constitute a critical medical emergency precipitated by the obstruction of blood flow to the heart muscle [1]. This blockage occurs when coronary arteries, responsible for supplying blood to the heart, become suddenly blocked, primarily due to a blood clot within an artery narrowed by the accumulation of atherosclerotic plaque [2]. Consequently, the segment of the heart muscle deprived of adequate blood supply begins to experience cellular death due to the lack of oxygen and essential nutrients [3]. Classic symptoms of MI complications include intense chest discomfort, difficulty in breathing, nausea, and vomiting [4]. Diagnosis involves a comprehensive evaluation, including physical examination, electrocardiogram (ECG) assessment, blood sample analysis for specific heart enzymes, and often imaging techniques such as coronary angiography [5]. Prompt therapeutic interventions are crucial to mitigate irreversible cardiac injury and improve patient prognosis. These interventions may include thrombolytic agents

to dissolve the clot, coronary procedures like angioplasty or coronary artery bypass graft surgery, and long-term therapeutic regimens to prevent recurrent MI incidents. Proactive anticipation of acute myocardial complications can significantly improve outcomes. Early prediction can be achieved using classification techniques, which involve organizing data into distinct classes based on identifiable characteristics. Classification is a vital tool across various domains, structuring information coherently and enabling thorough analysis. This can be executed manually or through automated means using computational algorithms, particularly for large and complex datasets [6].

One method of averting acute complications related to myocardial health is through the proactive anticipation of such occurrences. Anticipating these complications can be achieved by engaging in early predictive measures. Early prediction, as a fundamental approach, entails the application of classification techniques [7], [8]. Classification, a pivotal procedure, involves the systematic categorization of objects or data into distinct classes or groups based on identifiable characteristics or specific attributes. Across diverse domains, the practice of classification serves as a vital tool for the purpose of structuring information in a coherent manner, facilitating comprehension and enabling thorough analysis [9]. The process of classification can be executed through manual intervention by human operators or through automated means utilizing computational algorithms, particularly in scenarios involving voluminous datasets characterized by intricate complexities. Algorithms that are frequently utilized in the process of making predictions fall under the realm of Machine Learning, a subset of artificial intelligence that focuses on developing systems and algorithms that can learn and improve from experience without being explicitly programmed [10]. These algorithms are designed to analyze data, recognize patterns, and make intelligent decisions or predictions based on the information provided, thus enabling machines to perform tasks or make decisions that would typically require human intervention or intelligence. Machine Learning is a common set of algorithms frequently employed in the process of classification [11].

Machine Learning (ML), a subset of artificial intelligence, focuses on developing algorithms that enable computers to learn from data, recognize patterns, and make intelligent decisions. Unlike traditional programming methods, machine learning involves the training of computers through datasets, enabling them to identify patterns and autonomously reach decisions. The field of machine learning is in a state of continuous development, playing a vital role in addressing intricate challenges and enhancing productivity across different industries. Its significance is steadily increasing as it proves to be instrumental in tackling complex problems and streamlining operations in various sectors of the economy. Support Vector Machine [12], [13], Random Forest [7], [14], and Extreme Gradient Boosting (Xgboost) [15], [16] are frequently employed Machine Learning techniques for making classifications.

Support Vector Machine (SVM), Random Forest, and XGBoost are machine learning algorithms utilized to ensure the accuracy of predictive outcomes derived from patient medical records for classification purposes. SVM functions as a supervised learning model that scrutinizes data for classification and regression analysis, recognized for its efficacy in high-dimensional spaces and robust performance in achieving clear margin separation [17], [18]. Random Forest, on the other hand, operates as an ensemble learning technique that generates multiple decision trees during training and outputs the mode of the classes for classification tasks, providing high accuracy and resistance against overfitting [14], [19]. XGBoost, also known as Extreme Gradient Boosting, stands as an optimized distributed gradient boosting library formulated for efficiency, flexibility, and portability, renowned for its superior speed and effectiveness in classification and regression tasks [20]. By employing the Gradient Boosted Decision Tree (GBDT) algorithm framework, XGBoost improves the handling of missing values and provides enhanced regularization techniques to avoid overfitting, making it very effective for extensive data analysis [21], [22].

In conjunction with selecting appropriate machine learning models, managing missing data and class imbalances pose significant challenges in building robust predictive models. Methods for data imputation like K-Nearest Neighbors (KNN) imputation [23], Iterative imputation [24], [25], and Miss Forest [26] are crucial in managing incomplete datasets to prevent the loss of important information from missing values. KNN impute operates by identifying the k-nearest neighbors to a missing value and replacing it based on the mean or mode of these neighbors [27]. Iterative impute, also referred to as Multiple Imputation by Chained Equations (MICE), conducts multiple rounds of imputations, taking into account the uncertainty of missing data by generating various imputed datasets [28]. Miss Forest, a non-parametric imputation approach, employs random forest algorithms to predict and substitute missing values based on observed data [29].

Despite the progress made in ML and data imputation methodologies, effectively managing missing data continues to pose challenges. Prior research has often concentrated on individual imputation techniques or machine learning models without incorporating advanced strategies for hyperparameter optimization [18].

Additionally, the comparative efficacy of these integrated approaches in forecasting complications related to myocardial infarction has not been thoroughly explored. Optimizing the hyperparameters of machine learning models is essential for enhancing their performance [30]. Bayesian Optimization has emerged as a widely utilized method for tuning machine learning hyperparameters, constructing a surrogate model based on past evaluation outcomes of the target to determine the value that minimizes the objective function [31]. Particularly advantageous for problems with costly (high duration), non-differentiable, or complex function evaluations, Bayesian optimization proves to be highly effective [32], [33].

This study seeks to compare the effectiveness of various machine learning algorithms, including SVM, Random Forest, and XGBoost, in predicting myocardial infarction complications while incorporating data imputation techniques (KNN impute, Iterative impute, and Miss Forest). Additionally, it aims to evaluate the impact of hyperparameter optimization using Bayesian Optimization on predictive accuracy. The integration of these advanced techniques is expected to enhance early detection and management of myocardial infarction complications, thus improving patient outcomes and addressing the gaps in existing literature on predictive approaches. The contribution to this study is to see how much influence data imputation has on classification using machine learning methods on missing data and to see how much influence the optimization method has when performing hyperparameter tuning.

2. METHODS

This research process requires evaluating the efficacy of three machine learning algorithms: Support Vector Machine (SVM), Random Forest (RF), and XGBoost, each utilizing three distinct data imputation methods, specifically K-Nearest Neighbors (KNN) imputation, Iterative imputation (MICE), and Miss Forest imputation. All the models undergo assessment with hyperparameter adjustment through Bayesian Optimization. This study is split into five successive stages, involving data collection using a dataset on MI complications, data partitioning using k-fold cross-validation, model training, and evaluation of assessment results. The research progression undertaken in this investigation is illustrated in Fig. 1.

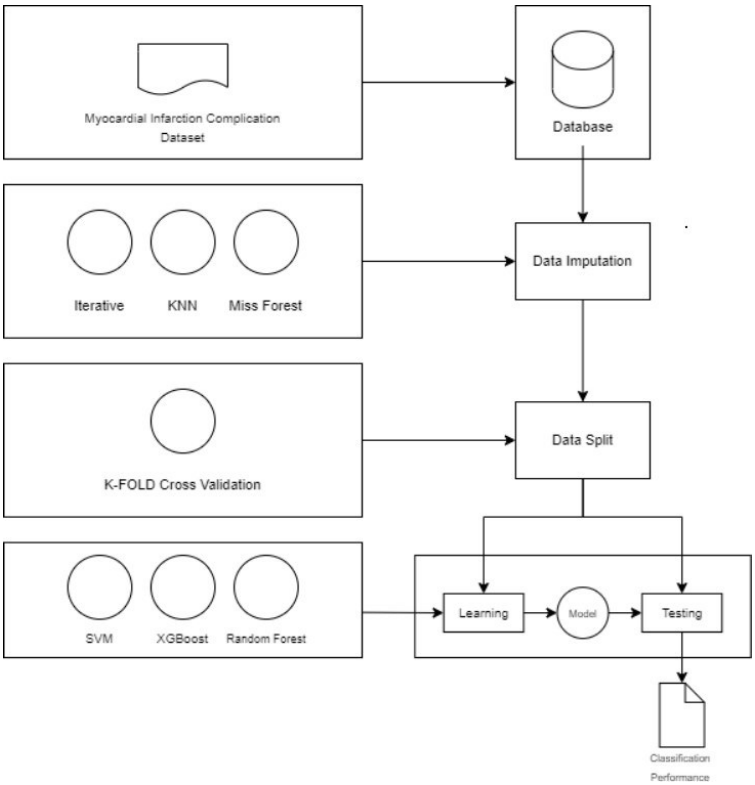

Fig. 1. Research Flowchart

2.1. Data Collection

The dataset analyzed in this research comprises 1700 instances containing 111 attributes related to the medical histories of patients. These attributes cover demographic details, medical background, results of diagnostic tests, and clinical observations during different phases of hospital stay, including admission, first

day, second day, and third day. Additionally, the dataset contains annotations for various potential complications of myocardial infarction (MI) like atrial fibrillation, supraventricular tachycardia, ventricular tachycardia, pulmonary edema, and others. The information is obtained from https://archive.ics.uci.edu/dataset/579/myocardial+infarction+complications.

2.2. Iterative Imputation

Iterative Imputation is a methodology utilized for managing missing data within datasets. It involves the gradual replacement of missing values through the application of predictive models. This process comprises multiple stages that are executed iteratively until the missing values are resolved and stabilized. The progression of the iterative imputation approach is structured as follows [34]:

1. Start: Load the dataset containing missing values.
2. Initialization: Substitute the missing values with preliminary estimations such as mean, median, or mode.
3. Iterations:
   - For each specific feature i with missing values:
     - Segment the data into target features (feature i) and predictor features (remaining features).
     - Develop a predictive model (e.g., regression, decision tree, etc.) to forecast the value of feature i.
     - Utilize the model for predicting and replacing the missing values in the features i.
   - Assess convergence:
     - Cease if the imputed values exhibit minimal alteration (convergence).
     - Otherwise, repeat this stage.
4. End: Generate the dataset with the replaced missing values.

2.3. MissForest Imputation

MissForest imputation is an approach that employs the Random Forest algorithm as a non-parametric technique for addressing missing values within a dataset. This method leverages the capabilities of Random Forest in managing intricate and interconnected data to offer precise estimations for the missing values. Random Forest, functioning as an ensemble learning algorithm, merges forecasts from numerous decision trees to enhance precision and mitigate overfitting. MissForest harnesses the potential of Random Forest to anticipate missing values by considering the available dataset values [29].

Similar to iterative imputation, MissForest operates in an iterative manner. During each cycle, the Random Forest model is trained using other features in the dataset to predict the missing values. Through the utilization of MissForest imputation, more accurate estimations can be used to replace missing values in the dataset, thereby enabling a more dependable subsequent analysis and modeling process. The progression of the MissForest imputation approach is structured as follows [26]:

1. Firstly, the identification of missing values involves determining their location and quantity within the dataset.
2. Subsequently, missing values are filled with initial estimates (such as mean, median, or mode) to initiate the iterative process.
3. The iterative process entails the segmentation of data into target features (specific feature with missing values) and predictor features (other features), followed by training a Random Forest model to forecast the value of the target feature based on the other features. The model is then employed to predict and fill the missing values in the target feature. The convergence is evaluated by assessing the magnitude of change in the imputed values; if minimal (indicating convergence), the process is halted, otherwise, it is repeated.
4. Ultimately, the final imputed outcomes from the iteration are utilized to substitute the missing values in the dataset.

2.4. KNN Imputation

K-Nearest Neighbors (KNN) imputation is a technique employed to address the absence of data values within a dataset by leveraging the principles of KNN. This approach involves replacing the missing values with the average (or mode for categorical variables) of the closest neighbors in the feature space. KNN, a non-parametric algorithm commonly utilized for both classification and regression tasks, is utilized in imputation to identify a set of k neighboring data points that lack missing values in order to infer and substitute the missing values. In this process, KNN employs a distance metric (such as Euclidean, Manhattan, or Minkowski) to locate

the nearest neighbors of data points with missing values. Subsequently, the missing values are imputed with the average (for numerical data) or mode (for categorical data) of these identified nearest neighbors.

Through the application of KNN imputation, the gaps in the dataset can be filled in a manner that leverages the localized similarities between data points, leading to more dependable estimates that align with the prevailing data patterns [23]. The equation of KNN imputation can be seen in (1).

$$d_{i,j} = \frac{\sum_{k=1}^{p} w_k \delta_{i,j,k}}{\sum_{k=1}^{p} w_k} \quad (1)$$

This research employs KNN imputation utilizing distance weighting parameters, which have the capability to manage binary, categorical, ordered, continuous, and semi-continuous distance variables. The calculation of the distance between two values involves a weighted mean of the contribution of each variable, with the weights intended to reflect the significance of the respective variable.

2.5. Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a machine learning technique employed to categorize a given set of training data along with associated labels. The optimal decision boundary is characterized by having the greatest distance and margin between the two data classes. SVM identifies the most suitable hyperplane for data segregation [18], [35].

Based on Fig. 2, to effectively divide the data into two distinct linear classes, SVM seeks out the ideal hyperplane by enhancing the separation or margin between the hyperplane and the nearest data points from each class [13], [36].

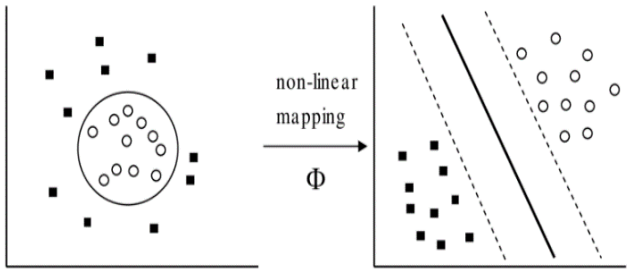

Fig. 2. SVM Model Generation [13]

In this particular investigation, SVM utilized a "kernel" setting of Polynomial with a regulatory parameter denoted as "C" set to 1. The subsequent equation outlines the SVM classification as well as the parameters relevant to the polynomial function. The equation of SVM kernel can be seen in (2).

$$K(x_i, x_j) = (x_i, x_j + c)^d \quad (2)$$

Here, the regulatory parameter is designated as c, while d signifies the polynomial degree, and K(xi, xj) corresponds to the kernel function.

2.6. Random Forest

The Random Forest algorithm is based on the concept of decision-making driven by a sequence of decisions structured in a decision tree format. Several decision trees are developed within the Random Forest structure, with each tree producing its own predictive outcomes. Eventually, the predictive class that receives the highest number of votes is selected as the ultimate prediction. A deeper comprehension of the Random Forest's framework can be attained by analyzing its structure. The architecture of Random Forest can be seen in Fig. 3 [11], [37].

Two techniques, namely bagging and random subspace, can be utilized for the construction of a Random Forest model. The subsequent section will elaborate on the steps involved in developing a Random Forest model in the field [39], [40]:

1. Utilizing the bootstrapping method to perform random resampling is a strategy that involves employing a sample size identical to that of the training data.
2. The random subspace technique entails selecting K attributes from a set of M attributes, where K is a value less than M, typically corresponding to the square root of M.
3. The development of a decision tree involves using bootstrap samples and previously selected attributes.

4. To attain the desired outcome, it is essential to repeat steps 1 to 3 multiple times in order to shape the tree accordingly. The quantity of trees within the Random Forest model is determined by assessing the out-of-bag error rate (OOB).

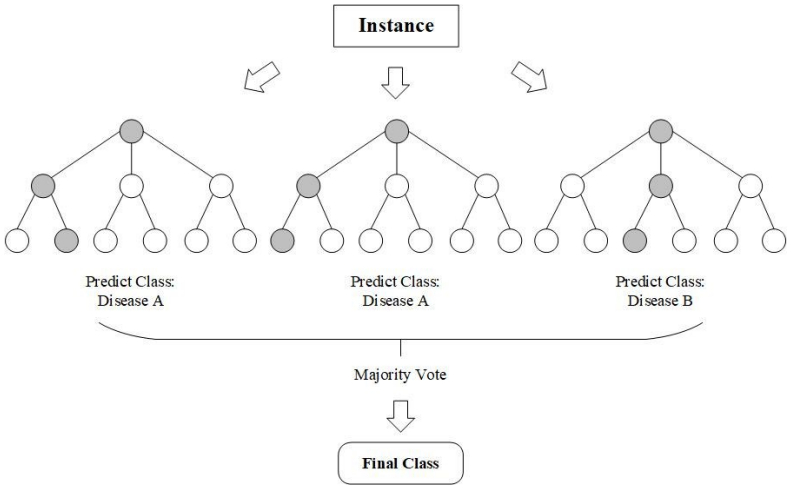

Fig. 3. Random Forest Model Architecture [38]

2.7. Extreme Gradient Boosting (Xgboost)

The XGBoost principle entails the development of an ensemble-based algorithm that amalgamates ensemble learning and decision trees [41]. When employing the XGBoost method, the concept of ensemble learning plays a crucial role in influencing the training process for the subsequent generation of trees. This influence is manifested in the addition of the residual outcome from the previous training process as a new threshold for the creation of a new tree. Such a process serves to diminish the likelihood of overfitting that may arise from the generation of new trees. Upon reaching the maximum number of iterations, the final output value is designated as the ultimate result. The architecture of XGBoost is visually depicted in Fig. 4, showcasing its underlying structure and components [42], [43].

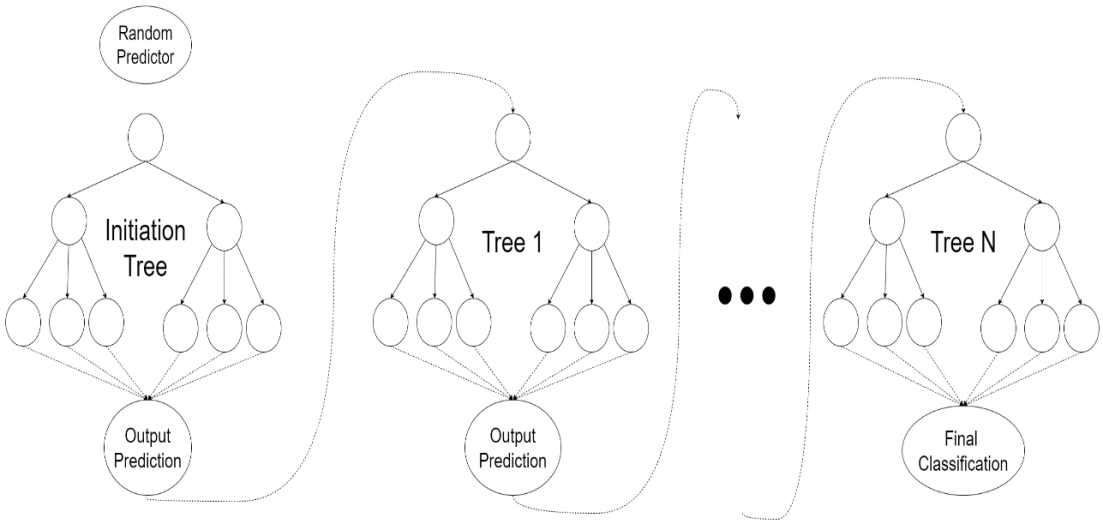

Fig. 4. Extreme Gradient Boosting Model Architecture [44]

A XGBoost model can be created through the process of forming trees and executing an ensemble learning method. The steps involved in developing a XGBoost model include the following [38], [44]:

1. The initialization phase begins by making a prediction for the 0-th tree, which is set to be equal to 0. This initial prediction sets the foundation for the subsequent steps in the model development process.

2. Next, the Splitting Mode is determined by the algorithm, which involves the calculation and traversal of all leaf node gain values until the maximum gain score relative to the root node is obtained. This step is crucial for identifying the optimal splitting points within the tree structure.
3. Following the determination of the Splitting Mode, the current binary leaf node set is established by continuing the calculation process until the gain score becomes negative or another stopping condition is met. This iterative process helps in refining the structure of the tree for better predictive accuracy.
4. Subsequently, the predicted value of the entire leaf node is calculated based on the information gathered from the previous steps. This predicted value serves as the basis for making decisions on how to further optimize the model for better performance.
5. A new tree is then established using the latest prediction result as the threshold, with the condition that the value is greater than the threshold. This process is repeated iteratively until the maximum number of trees specified for the model is reached, ensuring a comprehensive ensemble of trees is created.
6. Finally, the ultimate result of the XGBoost model is determined by calculating the output values of the latest node in the ensemble. This final step brings together the individual predictions of each tree to generate a collective output that represents the overall predictive power of the model.

2.8. Bayesian Optimization

Bayesian Optimization is a method for optimizing objective functions that are unknown and costly to evaluate, based on a probabilistic model. This technique is particularly valuable for tackling optimization challenges where direct assessment of the objective function is time-consuming or expensive, such as hyperparameter tuning in machine learning.

Bayesian Optimization involves several key steps [45]:

1. Prior Model: A probabilistic prior model, typically a Gaussian Process (GP), is established to represent the objective function. The GP is favored for its adaptability in capturing intricate functions and its ability to offer predictive uncertainty.
2. Observation Data: Begin with a small set of initial observation data, including appropriate inputs and outputs. The objective function is assessed at randomly chosen starting points or based on prior knowledge.
3. Construct Surrogate Model: Develop a surrogate model using the available observational data. This model aims to mimic the true objective function and provides a probabilistic approximation of the output.
4. Acquisition Function: Define an acquisition function that utilizes the surrogate model to identify the next point for evaluation. The acquisition function is crafted to balance exploration (exploring less-known regions) and exploitation (exploring areas expected to yield optimal outcomes).
5. Acquisition Function Optimisation: optimizing the acquisition function to determine the next input point for evaluation.
6. Evaluation and Update: Assessing the objective function at the new input point, updating the observation dataset with the new data.
7. Iteration: Iterating through the process from model construction to evaluation and update until a predefined stopping criterion is met, such as a maximum number of iterations or convergence.

2.9. Performance Metrics

In machine learning, the assessment of the combined model's classification performance is typically achieved by employing confusion matrices. These matrices offer a more effective means of displaying outcomes in classification problems, offering insights into both actual and predicted classification results.

Terms such as False Negative (FN), False Positive (FP), True Negative (TN), and True Positive (TP) are commonly utilized within the context of confusion matrices. True Positive (TP)is the test predicts "positive," and the result is actually positive. True Negative (TN) is the test predicts "negative," and the result is actually negative. False Positive (FP) is the test predicts "positive," but the result is actually negative. False Negative (FN) is the test predicts "negative," but the result is actually positive [46]. The terms are defined in Table 1.

Table 1. Confusion Matrix [47]

Actual Class	Predicted Class	
	True	**False**
True	True Positive (TP)	False Negative (FN)
False	False Positive (FP)	True Negative (TN)

The evaluation matrix under consideration incorporates these confusion matrix parameters to assess each parameter's performance [48].

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F1 = \frac{2 * precision * Recall}{precision + Recal} \tag{6}$$

Utilizing a mathematical formula that combines the curves, the Area Under the Curve (AUC) may be construed as the likelihood that the classification model will accurately differentiate between positive and negative instances. The method of categorization suggests that if chosen randomly, positive instances will yield higher rankings than negative ones. Consequently, an increased AUC signifies an enhanced capability of the classification model in effectively distinguishing between positive and negative categories. The primary objective in crafting an efficient classification model is to maximize the AUC value[49].

The AUC metric spans from 0 to 1, where a higher AUC denotes superior model performance. AUC can be modeled mathematically in (7).

$$AUC = \frac{\left(\frac{TP}{TP + FN}\right) x \left(\frac{TN}{TN + FP}\right)}{2} \tag{7}$$

Moreover, the AUC value's interpretation reflects the model's competence in distinguishing between positive and negative categories. Furthermore, AUC serves as a valuable instrument for model selection and comparison, enabling practitioners to assess the relative efficacy of different classifiers. The classification quality assessment based on the AUC value is illustrated in Table 2 [50].

Table 2. Categories of results from classification based on AUC values[50]

AUC Values	Category
0.90 – 1.00	Excellent
0.80 – 0.90	Good
0.70 – 0.80	Fair
0.60 – 0.70	Poor
0.50 – 0.60	Failure

3. RESULTS AND DISCUSSION

The results section provides a detailed analysis of the performance of the SVM, Random Forest, and XGBoost classification algorithms, each coupled with different data imputation methods (KNN, Iterative, and MissForest) and hyperparameter optimization via Bayesian Optimization. The metrics used for evaluation include accuracy, precision, sensitivity, F1-score, and AUC of the ROC curve. The analysis is conducted using k-fold cross-validation with k-values of 2 and 3. The evaluation aims to compare the performance of machine learning algorithms and gauge the impact of data imputation. In this study, k-fold cross validation is employed for splitting the data due to imbalanced data classes [51].

3.1. Testing Results with K-Fold value 2

This part presents the empirical results derived from the machine learning classification model utilizing a k-fold value of 2. Based on Table 1, evaluation of the machine learning classification model using a k-fold value of 2 indicates a high level of accuracy. The model's accuracy rate of 97.1% demonstrates its capability in effectively categorizing the data. Nonetheless, the outcomes of additional performance metrics reveal a subpar level of performance. Within the SVM method utilizing iterative imputation, the AUC result reached its peak at 0.589. Subsequently, a further test will be carried out employing a k-fold value of 3. A comparison of performance metrics for all strategies utilized is presented in Fig. 5.

Table 3. Classification Result using K-Fold value 2

Model	Imputation Method	Peformance Metrics				
		AUC	F1	Accuracy	Sensitivity	Precision
SVM	Iterative	0.589	0.25	0.965	0.188	0.375
	MissForest	0.500	0.00	0.969	0.000	nan
	KNN	0.530	0.111	0.969	0.062	0.500
Random Forest	Iterative	0.500	0.00	0.969	0.000	nan
	MissForest	0.500	0.00	0.969	0.000	nan
	KNN	0.500	0.00	0.969	0.000	nan
Xgboost	Iterative	0.500	0.00	0.969	0.000	nan
	MissForest	0.531	0.118	0.971	0.062	1,000
	KNN	0.531	0.118	0.971	0.062	1,000

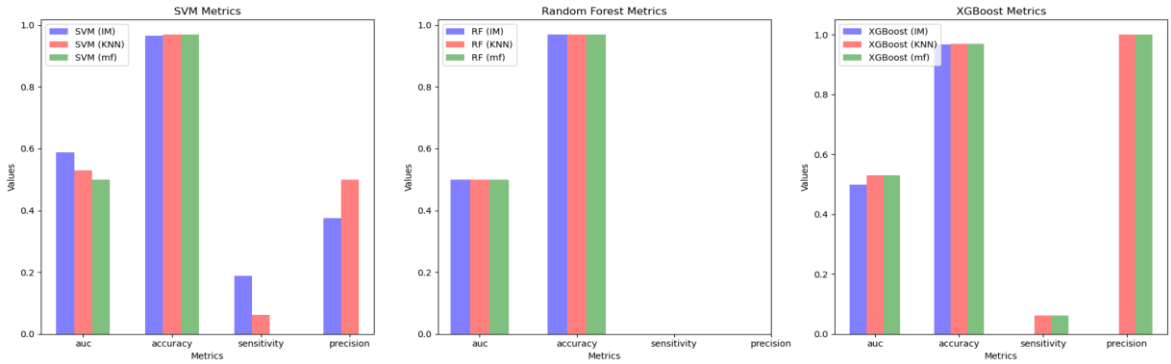

Fig. 5. Comparison of Machine Learning Methods using K-Fold Value 2

3.2. Testing Results with K-Fold value 3

This part presents the empirical results derived from the machine learning classification model utilizing a k-fold value of 3. Based on Table 4, the results significantly improved with a k-fold value of 3, particularly for the SVM and XGBoost models. This improvement highlights the importance of choosing an appropriate value for k in cross-validation to obtain a more reliable performance assessment. Within the SVM approach, all performance metrics demonstrated optimal outcomes when employing Iterative and KNN imputation techniques. The Xgboost method also exhibited favorable results, achieving a maximum accuracy of 99.4% and an AUC of 0.898, placing it within the good range. The Random Forest algorithm consistently performed poorly, with an AUC of 0.5 across different imputation methods and k-values. This suggests that Random Forest may not be suitable for this particular task, or it might require further tuning or preprocessing adjustments. Iterative and KNN imputation methods yielded superior results compared to MissForest, especially when paired with the SVM and XGBoost algorithms. This indicates that these imputation methods may be better suited for this specific dataset. A comparison of performance metrics for all strategies utilized is presented in Fig. 6.

Table 4. Classification Result using K-Fold value 3

Model	Imputation Method	Peformance Metrics				
		AUC	F1	Accuracy	Sensitivity	Precision
SVM	Iterative	1.000	1.000	1.000	1.000	1.000
	MissForest	0.712	0.568	0.979	0.426	0.852
	KNN	1.000	1.000	1.000	1.000	1.000
Random Forest	Iterative	0.500	0.000	0.968	0.000	nan
	MissForest	0.500	0.000	0.968	0.000	nan
	KNN	0.500	0.000	0.968	0.000	nan
Xgboost	Iterative	0.898	0.887	0.994	0.796	1.000
	MissForest	0.750	0.667	0.984	0.500	1.000
	KNN	0.722	0.615	0.982	0.444	1.000

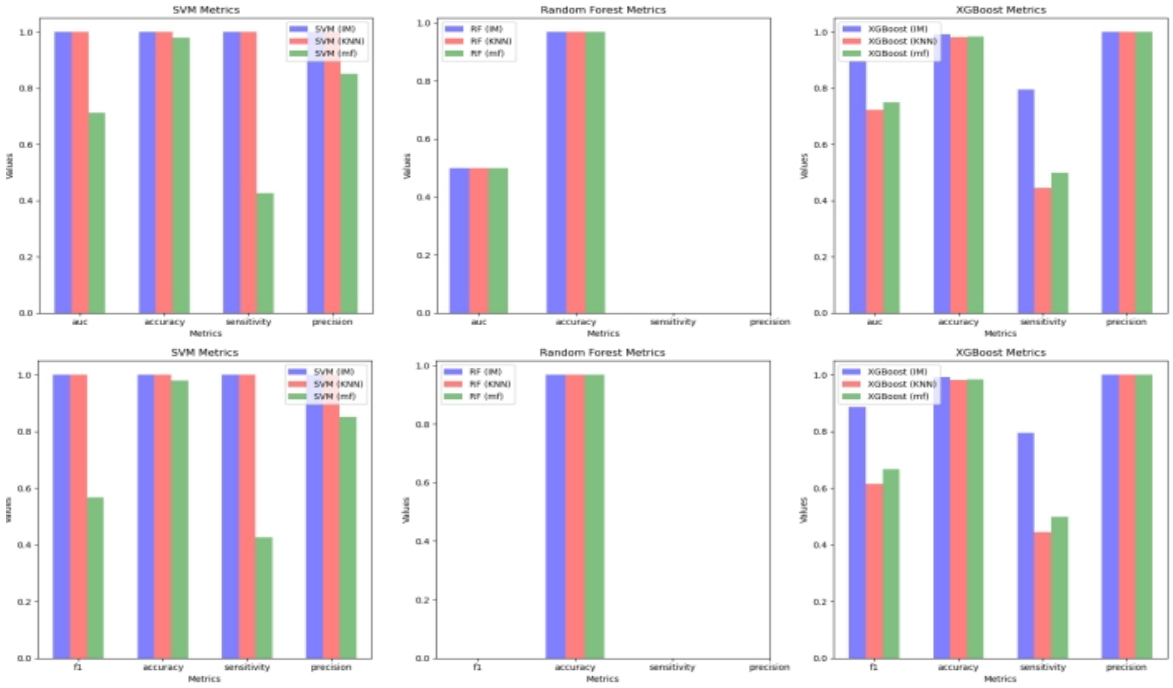

Fig. 6. Comparison of Machine Learning Methods using K-Fold Value 3

3.3. Discussion

The assessment findings indicate that the utilization of imputation techniques proved to be effective in yielding satisfactory outcomes for the SVM and Xgboost algorithms. In the SVM algorithm, exemplary results were achieved in terms of accuracy, precision, sensitivity, and F1 scores of 100%, along with an AUC of 1.00, when employing the iterative and knn imputation techniques with a k-fold of 3. These outcomes demonstrated an enhancement compared to a k-fold of 2. Conversely, in the Xgboost algorithm, optimal outcomes were observed with the iterative imputation technique, showcasing an accuracy of 99.4%, precision of 100%, sensitivity of 79.6%, F1 score of 88.7%, and an AUC of 0.898. The results suggest that Xgboost can yield favorable outcomes when utilizing a k-fold value of 3, overcoming overfitting issues associated with imbalanced data. Nevertheless, the outcomes for Random Forest were found to be unsatisfactory, as indicated by an AUC value of 0.5, signifying its failure in addressing overfitting concerns within the dataset.

Upon comparing the various methodologies applied, it is evident that the Iterative Imputation Method stands out as the most effective approach for handling missing data concerns. Conversely, Random Forest exhibited subpar results due to its AUC value of 0.5, despite achieving high accuracy levels. These results imply that the prevalence of the majority class significantly influences the high accuracy rates through correct classification. The perfect scores (100%) observed in the SVM with iterative and KNN imputation for k=3 might indicate overfitting. It would be beneficial to investigate this further by using additional evaluation metrics or validation techniques. Moving forward, additional research is warranted to explore the implementation of data balancing techniques, intended to equalize the representation of minority class data with that of the majority class. Implement techniques such as SMOTE (Synthetic Minority Over-sampling Technique) [51] or ADASYN (Adaptive Synthetic Sampling) [52], [53] to balance the dataset before training the models. This can help improve the model's performance on minority classes and provide a more accurate evaluation of its efficacy. While Bayesian Optimization was used for hyperparameter tuning, further exploration with other optimization techniques such as Grid Search [54] or Random Searchz [55] might uncover better hyperparameter configurations.

4. CONCLUSION

According to the findings presented earlier, the Iterative Imputation technique demonstrates superior performance in SVM and Xgboost algorithms for classification tasks. SVM achieves perfect accuracy, precision, sensitivity, F1 test score of 100%, and AUC of 1.00. XGBoost accomplishes 99.4% accuracy, 100% precision, 79.6% sensitivity, F1 score of 88.7%, and AUC of 0.898. Similarly, KNN Imputation in SVM yields

identical outcomes to Iterative Imputation in SVM. However, poor classification results are observed with Random Forest due to data class imbalance leading to overfitting.

In forthcoming studies, it is imperative to incorporate class balancing techniques like SMOTE and ADASYN in order to enhance the efficacy of the Random Forest algorithm and to support imputation approaches such as MissForest and KNN Imputation. The utilization of class balancing methods is anticipated to address the issue of overfitting during the classification process. While Bayesian Optimization was used for hyperparameter tuning, further exploration with other optimization techniques such as Grid Search or Random Search might uncover better hyperparameter configurations.

Acknowledgments

This work was supported in part by LPPM of Lambung Mangkurat University under Contract Number 1090.56/UN8.2/PG/2024. We want to say thank you to Rector of Lambung Mangkurat University and Head of LPPM of Lambung Mangkurat University.

REFERENCES

[1] A. A. Damluji *et al.*, "Mechanical Complications of Acute Myocardial Infarction: A Scientific Statement From the American Heart Association," *Circulation*, vol. 144, no. 2, 2021, https://doi.org/10.1161/CIR.0000000000000985.

[2] N. R. Stephens, C. S. Restrepo, S. S. Saboo, and A. J. Baxi, "Overview of complications of acute and chronic myocardial infarctions: revisiting pathogenesis and cross-sectional imaging," *Postgrad. Med. J.*, vol. 95, no. 1126, pp. 439–450, 2019, https://doi.org/10.1136/postgradmedj-2018-136279.

[3] J. Yang *et al.*, "Current status of emergency medical service use in ST-segment elevation myocardial infarction in China: Findings from China Acute Myocardial Infarction (CAMI) Registry," *Int. J. Cardiol.*, vol. 406, p. 132040, 2024, https://doi.org/10.1016/j.ijcard.2024.132040.

[4] N. Dewaswala and R. D. Chait, "A Complication of Acute Myocardial Infarction," *HCA Healthc. J. Med.*, vol. 1, no. 3, 2020, https://doi.org/10.36518/2689-0216.1054.

[5] J. Zalewski, K. Nowak, P. Furczynska, and M. Zalewska, "Complicating Acute Myocardial Infarction. Current Status and Unresolved Targets for Subsequent Research," *J. Clin. Med.*, vol. 10, no. 24, p. 5904, 2021, https://doi.org/10.3390/jcm10245904.

[6] E. Moras *et al.*, "Complications in Acute Myocardial Infarction: Navigating Challenges in Diagnosis and Management," *Hearts*, vol. 5, no. 1, pp. 122–141, 2024, https://doi.org/10.3390/hearts5010009.

[7] C. Browne *et al.*, "Multivariate random forest prediction of poverty and malnutrition prevalence," *PLoS One*, vol. 16, no. 9, pp. 1–23, 2021, https://doi.org/10.1371/journal.pone.0255519.

[8] M. Rostami, K. Berahmand, and S. Forouzandeh, "A novel community detection based genetic algorithm for feature selection," *J. Big Data*, vol. 8, no. 1, 2021, https://doi.org/10.1186/s40537-020-00398-3.

[9] Y. Suzuki, A. Suzuki, S. Nakamura, T. Ishikawa, and A. Kinoshita, "Machine learning model estimating number of COVID-19 infection cases over coming 24 days in every province of South Korea (XGBoost and MultiOutputRegressor)," *medRxiv*, p. 2020.05.10.20097527, 2020, https://doi.org/10.1101/2020.05.10.20097527.

[10] J. R. Saura, B. R. Herraez, 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, https://doi.org/10.1109/ACCESS.2019.2905301.

[11] D. Chaerul, E. Saputra, Y. Maulana, T. A. Win, R. Phann, and W. Caesarendra, "Implementation of Machine Learning and Deep Learning Models Based on Structural MRI for Identification of Autism Spectrum Disorder," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 2, pp. 307–318, 2023, https://doi.org/10.26555/jiteki.v9i2.26094.

[12] H. Hairani, K. E. Saputro, and S. Fadli, "K-means-SMOTE for handling class imbalance in the classification of diabetes with C4.5, SVM, and naive Bayes," *J. Teknol. dan Sist. Komput.*, vol. 8, no. 2, pp. 89–93, 2020, https://doi.org/10.14710/jtsiskom.8.2.2020.89-93.

[13] D. Fitria, T. H. Saragih, D. Kartini, and F. Indriani, "Classification of Appendicitis in Children Using SVM with KNN Imputation and SMOTE Approach to Improve Prediction Quality," *J. Electron. Electromed. Eng. Med. Informatics*, vol. 6, no. 3, pp. 302–311, 2024, https://doi.org/10.35882/jeeemi.v6i3.470.

[14] T. H. Saragih, V. N. Wijayaningrum, and M. Haekal, "Jatropha Curcas Disease Identification using Random Forest," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 1, p. 9, 2021, https://doi.org/10.26555/jiteki.v7i1.20141.

[15] N. Zhai, P. Yao, and X. Zhou, "Multivariate time series forecast in industrial process based on XGBoost and GRU," vol. 2020, no. X, pp. 1397–1400, 2020, https://doi.org/10.1109/ITAIC49862.2020.9338878.

[16] C. Chen *et al.*, "Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier," *Comput. Biol. Med.*, vol. 123, p. 103899, 2020, https://doi.org/10.1016/j.compbiomed.2020.103899.

[17] J. Zhou, P. Yang, P. Peng, M. Khandelwal, and Y. Qiu, "Performance Evaluation of Rockburst Prediction Based on PSO-SVM, HHO-SVM, and MFO-SVM Hybrid Models," *Mining, Metall. Explor.*, vol. 40, no. 2, pp. 617–635, 2023, https://doi.org/10.1007/s42461-022-00713-x.

[18] U. Haris, V. Kabeer, and K. Afsal, "Breast cancer segmentation using hybrid HHO-CS SVM optimization techniques," *Multimed. Tools Appl.*, 2024, https://doi.org/10.1007/s11042-023-18025-7.

[19] B. Thomas and C. J., "Random forest application on cognitive level classification of E-learning content," *Int. J.*

Electr. Comput. Eng., vol. 10, no. 4, p. 4372, 2020, https://doi.org/10.11591/ijece.v10i4.pp4372-4380.
[20] J. Wang, Q. Cheng, and Y. Dong, "An XGBoost-based multivariate deep learning framework for stock index futures price forecasting," *Kybernetes*, vol. 52, no. 10, pp. 4158-4177. 2022, https://doi.org/10.1108/K-12-2021-1289.
[21] M. D. Guillen, J. Aparicio, and M. Esteve, "Gradient tree boosting and the estimation of production frontiers," *Expert Syst. Appl.*, vol. 214, p. 119134, 2023, https://doi.org/10.1016/j.eswa.2022.119134.
[22] L. Nespoli and V. Medici, "Multivariate Boosted Trees and Applications to Forecasting and Control," no. 2017, 2020, [Online]. Available: http://arxiv.org/abs/2003.03835.
[23] A. Fadlil, "K Nearest Neighbor Imputation Performance on Missing Value Data Graduate User Satisfaction," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 4, pp. 570–576, 2022, https://doi.org/10.29207/resti.v6i4.4173.
[24] D. Jarrett, B. Cebere, T. Liu, A. Curth, and M. van der Schaar, "HyperImpute: Generalized Iterative Imputation with Automatic Model Selection," *Proc. Mach. Learn. Res.*, vol. 162, pp. 9916–9937, 2022, https://proceedings.mlr.press/v162/jarrett22a.html.
[25] N. Fazakis, G. Kostopoulos, S. Kotsiantis, and I. Mporas, "Iterative Robust Semi-Supervised Missing Data Imputation," *IEEE Access*, vol. 8, pp. 90555–90569, 2020, https://doi.org/10.1109/ACCESS.2020.2994033.
[26] I. Nirmala, H. Wijayanto, and K. A. Notodiputro, "Prediction of Undergraduate Student's Study Completion Status Using MissForest Imputation in Random Forest and XGBoost Models," *ComTech Comput. Math. Eng. Appl.*, vol. 13, no. 1, pp. 53–62, 2022, https://doi.org/10.21512/comtech.v13i1.7388.
[27] A. R. Ismail, N. Z. Abidin, and M. K. Maen, "Systematic Review on Missing Data Imputation Techniques with Machine Learning Algorithms for Healthcare," *J. Robot. Control*, vol. 3, no. 2, pp. 143–152, 2022, https://doi.org/10.18196/jrc.v3i2.13133.
[28] V. Nassiri, G. Molenberghs, G. Verbeke, and J. Barbosa-Breda, "Iterative Multiple Imputation: A Framework to Determine the Number of Imputed Datasets," *Am. Stat.*, vol. 74, no. 2, pp. 125–136, 2020, https://doi.org/10.1080/00031305.2018.1543615.
[29] S. Hong and H. S. Lynn, "Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction," *BMC Med. Res. Methodol.*, vol. 20, no. 1, p. 199, 2020, https://doi.org/10.1186/s12874-020-01080-1.
[30] P. Eleftheriadis, S. Leva, and E. Ogliari, "Bayesian Hyperparameter Optimization of stacked Bidirectional Long Short-Term Memory neural network for the State of Charge estimation," *Sustain. Energy, Grids Networks*, vol. 36, p. 101160, 2023, https://doi.org/10.1016/j.segan.2023.101160.
[31] L. Tani and C. Veelken, "Comparison of Bayesian and particle swarm algorithms for hyperparameter optimisation in machine learning applications in high energy physics," *Comput. Phys. Commun.*, vol. 294, p. 108955, 2024, https://doi.org/10.1016/j.cpc.2023.108955.
[32] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, and S. H. Deng, "Hyperparameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019, https://doi.org/10.11989/JEST.1674-862X.80904120.
[33] R. Herrera Casanova and A. Conde, "Enhancement of LSTM models based on data pre-processing and optimization of Bayesian hyperparameters for day-ahead photovoltaic generation prediction," *Comput. Electr. Eng.*, vol. 116, p. 109162, 2024, https://doi.org/10.1016/j.compeleceng.2024.109162.
[34] H. I. Oberman, S. van Buuren, and G. Vink, "Missing the Point: Non-Convergence in Iterative Imputation Algorithms," *International Conference on Machine Learning*, pp. 9916-9937, 2021, https://doi.org/10.5334/dsj-2017-037.
[35] H. Saputra, D. Stiawan, and H. Satria, "Malware Detection in Portable Document Format (PDF) Files with Byte Frequency Distribution (BFD) and Support Vector Machine (SVM)," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 4, pp. 1144–1153, 2023, https://doi.org/10.26555/jiteki.v9i4.27527.
[36] W. J. Bidul, S. Surono, and T. B. Kurniawan, "Comparative Evaluation of Feature Selection Methods for Heart Disease Classification with Support Vector Machine," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 10, no. 2, pp. 265–278, 2024, https://doi.org/10.26555/jiteki.v10i2.28647.
[37] X. Zhou, P. Lu, Z. Zheng, D. Tolliver, and A. Keramati, "Accident Prediction Accuracy Assessment for Highway-Rail Grade Crossings Using Random Forest Algorithm Compared with Decision Tree," *Reliab. Eng. Syst. Saf.*, vol. 200, p. 106931, 2020, https://doi.org/10.1016/j.ress.2020.106931.
[38] T. H. Saragih and M. I. Mazdadi, "Comparison of Air Quality Prediction using Random Forest and Gradient Boosting Tree," in *2023 Eighth International Conference on Informatics and Computing (ICIC)*, pp. 1–5, 2023, https://doi.org/10.1109/ICIC60109.2023.10382104.
[39] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata J. Promot. Commun. Stat. Stata*, vol. 20, no. 1, pp. 3–29, 2020, https://doi.org/10.1177/1536867X20909688.
[40] A. Khairunnisa, K. A. Notodiputro, and B. Sartono, "A Comparative Study of Random Forest and Double Random Forest Models from View Points of Their Interpretability," *Sci. J. Informatics*, vol. 11, no. 1, pp. 207–218, 2024, https://doi.org/10.15294/sji.v11i1.48721.
[41] M. Dwinanda, N. Satyahadewi, and W. Andani, "Classification Of Student Graduation Status Using Xgboost Algorithm," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 17, no. 3 SE-Articles, 2023, https://doi.org/10.30598/barekengvol17iss3pp1785-1794.
[42] D. Arifah, T. H. Saragih, D. Kartini, M. Muliadi, and M. I. Mazdadi, "Application of SMOTE to Handle Imbalance

Class in Deposit Classification Using the Extreme Gradient Boosting Algorithm," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 2, pp. 396–410, 2023, https://doi.org/10.26555/jiteki.v9i2.26155.
[43] D. Pebrianti, H. Kurniawan, L. Bayuaji, and R. Rusdah, "XgBoost Hyper-Parameter Tuning Using Particle Swarm Optimization for Stock Price Forecasting," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 9, no. 4, pp. 1179–1195, 2023, https://doi.org/10.26555/jiteki.v9i4.27712.
[44] T. H. Saragih, R. Ramadhani, M. I. Mazdadi, and M. Haekal, "Energy Efficiency in Buildings Using Multivariate Extreme Gradient Boosting," in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, pp. 1–5, 2022, https://doi.org/10.1109/ICIC56845.2022.10006902.
[45] V. Nguyen, "Bayesian Optimization for Accelerating Hyper-Parameter Tuning," in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 302–305, 2019, https://doi.org/10.1109/AIKE.2019.00060.
[46] G. Phillips *et al.*, "Setting nutrient boundaries to protect aquatic communities: The importance of comparing observed and predicted classifications using measures derived from a confusion matrix," *Sci. Total Environ.*, vol. 912, p. 168872, 2024, https://doi.org/10.1016/j.scitotenv.2023.168872.
[47] Y. Wang, Y. Jia, Y. Tian, and J. Xiao, "Deep reinforcement learning with the confusion-matrix-based dynamic reward function for customer credit scoring," *Expert Syst. Appl.*, vol. 200, p. 117013, 2022, https://doi.org/10.1016/j.eswa.2022.117013.
[48] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognit.*, vol. 91, pp. 216–231, 2019, https://doi.org/10.1016/j.patcog.2019.02.023.
[49] J. Y. Verbakel *et al.*, "ROC curves for clinical prediction models part 1. ROC plots showed no added value above the AUC when evaluating the performance of clinical prediction models," *J. Clin. Epidemiol.*, vol. 126, pp. 207–216, 2020, https://doi.org/10.1016/j.jclinepi.2020.01.028.
[50] J. Xu, "Comparing multi-class classifier performance by multi-class ROC analysis: A nonparametric approach," *Neurocomputing*, vol. 583, p. 127520, 2024, https://doi.org/10.1016/j.neucom.2024.127520.
[51] T. R. Mahesh *et al.*, "AdaBoost Ensemble Methods Using K-Fold Cross Validation for Survivability with the Early Detection of Heart Disease," *Comput. Intell. Neurosci.*, 2022, https://doi.org/10.1155/2022/9005278.
[52] Y. He, J. Zhou, C. Cao, S. Wang, and H. Fu, "Detection of electricity theft based on Minimal Gated Memory network combined adaptive synthesis sampling and decision tree," *Sustain. Energy, Grids Networks*, vol. 39, p. 101415, 2024, https://doi.org/10.1016/j.segan.2024.101415.
[53] T. A. Assegie, A. O. Salau, K. Sampath, R. Govindarajan, S. Murugan, and B. Lakshmi, "Evaluation of Adaptive Synthetic Resampling Technique for Imbalanced Breast Cancer Identification," *Procedia Comput. Sci.*, vol. 235, pp. 1000–1007, 2024, https://doi.org/10.1016/j.procs.2024.04.095.
[54] S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, "The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction," *Clean. Eng. Technol.*, vol. 15, p. 100664, 2023, https://doi.org/10.1016/j.clet.2023.100664.
[55] N. A. Pérez-Padilla *et al.*, "Optimizing trigger timing in minimal ovarian stimulation for In Vitro fertilization using machine learning models with random search hyperparameter tuning," *Comput. Biol. Med.*, vol. 179, p. 108856, 2024, https://doi.org/10.1016/j.compbiomed.2024.108856.

BIOGRAPHY OF AUTHORS

Muhammad Itqan Mazdadi, a lecturer in the Department of Computer Science, Lambung Mangkurat University. His research interest is centered on Data Science and Computer Networking. Before becoming a lecturer, he completed his undergraduate program in the Computer Science Department at Lambung Mangkurat University In 2013. He then completed his master's degree from Department of Informatics at Islamic Indonesia University, Yogyakarta. Currently, he serves as the Secretary of the Computer Science Department at Lambung Mangkurat University. Email: mazdadi@ulm.ac.id. Orcid ID: 0000-0002-8710-4616.

Triando Hamonangan Saragih, currently holding the position of a lecturer within the Department of Computer Science at Lambung Mangkurat University, is heavily immersed in the realm of academia, with a profound focus on the multifaceted domain of Data Science. His academic pursuits commenced with the successful completion of his bachelor's degree in Informatics at the esteemed Brawijaya University, located in the vibrant city of Malang, back in the year 2016. Building upon this foundational achievement, he proceeded to further enhance his scholarly credentials by enrolling in a master's program in Computer Science at Brawijaya University, Malang, culminating in the conferral of his advanced degree in 2018. The research field he is involved in is Data Science. Email: triando.saragih@ulm.ac.id. Orcid ID: 0000-0003-4346-3323.

Irwan Budiman, He is a lecturer at Lambung Mangkurat University and currently serves as the Coordinator in the Department of Computer Science, Faculty of Mathematics and Natural Sciences. He earned his Bachelor's Degree in Informatics Engineering from Islam Indonesia University, Yogyakarta. Subsequently, he completed his Master's studies in information systems at Diponegoro University, Semarang. His research interests include data mining, human-computer interaction, applied business intelligence, and e-government. Email: irwan.budiman@ulm.ac.id. Orcid ID: 0000-0002-0514-7429.

Andi Farmadi, a senior lecturer in the Computer Science program at Lambung MangkuratUniversity. He has been teaching since 2008 and currently serves as the Head of the Data Science Lab since 2018. He completed his undergraduate studies at Hasanuddin University and his graduate studies at Bandung Institute of Technology. His research area, up to the present, focuses on Data Science. One of his research projects, along with other researchers, published in the International Conference of Computer and Informatics Engineering (IC2IE), is titled "Hyperparameter tuning using GridsearchCV on the comparison of the activation function of the ELM method to the classification of pneumonia in toddlers," and this research was published in 2021. Email: andifarmadi@ulm.ac.id. Orcid ID: 0009-0009-0926-8082.

Ahmad Tajali, a student at Lambung Mangkurat University who began his education in 2021 in the Department of Computer Science. His current research field is Data Science. Email: ahmadtajali61@gmail.com.