Journal of Soft Computing Exploration
Homepage: htttps://shmpublisher.com/index.php/joscex
p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Enhancing diabetes classification performance using XGBoost integrated with SMOTE and bayesian hyperparameter optimization

Muhammad Nurul Ihyaul Ulum[1], Jumanto Unjung[2]
[1,2] Department of Computer Science, Universitas Negeri Semarang, Indonesia

Article Info

Article history:

Received Mar 9, 2026
Revised Mar 16, 2026
Accepted Mar 18, 2026

Keywords:

Diabetes classification
XGboost
SMOTE
Bayesian optimization
ROC

ABSTRACT

Diabetes mellitus is a long-term metabolic disorder that is becoming more common around the world. Finding people at risk early can help prevent serious health problems and improve patient outcomes. Machine learning is often used to predict diabetes, but imbalanced medical data can make it harder for models to spot positive cases. In this study, we created a diabetes classification model by combining the Extreme Gradient Boosting (XGBoost) algorithm with the Synthetic Minority Over-sampling Technique (SMOTE), and we used Bayesian Optimization to fine-tune the model's settings. We worked with the Pima Indians Diabetes Dataset, which has 768 patient records and eight clinical features. Our steps included preprocessing the data, splitting it into training and testing sets, using SMOTE to balance the training data classes, training the XGBoost model, and performing hyperparameter tuning using Bayesian Optimization with Stratified 5-Fold Cross-Validation to determine the optimal parameter configuration. The final model reached an accuracy of 0.88, a precision of 0.79, a recall of 0.91, an F1-score of 0.84, and a ROC-AUC of 0.955. These results show that our approach can identify diabetes cases more effectively while keeping strong overall performance.

Corresponding Author:

Jumanto Unjung,
Department of Computer Science,
Universitas Negeri Semarang
Sekaran, Kota Semarang, Jawa Tengah, Indonesia
Email: jumanto@mail.unnes.ac.id
https://doi.org/10.52465/joscex.v7i1.3

1. INTRODUCTION

Diabetes mellitus is a long-term metabolic disease that has become a major health problem around the world, with more people affected each year. It happens when the body cannot control blood sugar levels properly because of problems with insulin production, insulin resistance, or both [1]. If diabetes isn't treated correctly, it can cause major health problems like heart disease, renal failure, nerve damage, and permanent eyesight loss. The World Health Organization (WHO) says that diabetes is one of the top causes of death around the world [1]. In Indonesia, diabetes cases are also rising, with about 20.4 million people affected and a prevalence rate of 11.3% [2]. This makes Indonesia one of the countries with the most diabetes patients in Southeast Asia [3]. Detecting diabetes risk early is very important to prevent serious complications and help reduce the strain on healthcare systems.

The advancement of information technology and data analysis has driven the use of machine learning methods in the health sector, especially for disease prediction and classification. Machine learning allows

computer systems to learn patterns from historical data, enabling the creation of predictive models that support faster and data-driven clinical decision-making processes [4], [5]. Various algorithms such as Decision Tree, Logistic Regression, Random Forest, and XGBoost have been widely used in disease classification research due to their ability to handle complex data [6]–[8]. Among these algorithms, XGBoost is known for its high predictive performance and good computational efficiency on tabular data through an ensemble gradient boosting approach [9].

Several previous studies have developed diabetes prediction models using machine learning methodologies. Iparraguirre-Villanueva et al. [8] combined several classification algorithms with the SMOTE technique and reported that the K-Nearest Neighbor (KNN) model provided the best performance with an accuracy of 79.6% and a recall value of 79%. Research by Islam et al. [10] applied the XGBoost algorithm with the ADASYN oversampling technique and obtained an accuracy of 81% and an AUC value of 0.84. Meanwhile, Talukder et al. [11] conducted a comparative study of several machine learning algorithms and found that Random Forest produced an accuracy of 85.5% with a recall of 81.3%. In addition, Shetty et al. [12] developed a clinical decision support system for the classification of gestational diabetes mellitus using a wrapper feature selection approach combined with several machine learning algorithms, achieving accuracy rates between 75% and 82%. Another study by Rahman et al. [13] proposed a diabetes prediction approach based on feature selection and boosting-based classifiers on the Pima Indian Diabetes Dataset (PIDD) and DiaHealth, with an AUC value reaching 90% on the PIDD dataset. These studies indicate that machine learning approaches have great potential to improve the performance of diabetes disease classification.

However, one of the main challenges in developing diabetes classification models is data distribution imbalance (class imbalance) [4], [14], [15]. In many diabetes datasets, the number of patients without diabetes is generally greater than the number of patients diagnosed with diabetes. This condition can cause machine learning models to become biased toward the majority class, thereby reducing the model's ability to detect positive cases [16]. In the context of medical diagnosis, misclassification in the form of false negatives, where patients who actually have diabetes are predicted as not having the disease, can result in delayed diagnosis and increase the risk of more serious complications [13], [17]. Therefore, addressing data imbalance becomes an important aspect in the development of reliable medical classification models.

One common approach used to overcome data imbalance is the oversampling technique. This method adds synthetic data to the minority class so that the data distribution becomes more balanced without eliminating information from the dataset [18], [19]. One of the most widely used techniques is the SMOTE, which generates synthetic samples based on the proximity between data points in the feature space [12], [20]–[22]. In addition, the performance of machine learning models is also influenced by the selection of optimal hyperparameters. Bayesian Optimization is widely used as a hyperparameter optimization technique because it can find the best parameter combinations more efficiently compared to conventional methods such as grid search or random search [23], [24].

Most previous studies have primarily focused on improving overall classification accuracy when developing diabetes prediction models. However, in medical diagnosis tasks, accuracy alone may be insufficient because it can mask poor detection of minority classes. In particular, the ability of models to detect positive diabetes cases, which is reflected by recall, remains an important yet often underemphasized aspect. Moreover, while techniques like SMOTE have been extensively employed to tackle class imbalance and hyperparameter optimization methods have been utilized to enhance model performance, research that concurrently combines SMOTE and Bayesian Optimization with the XGBoost algorithm remains relatively scarce in the context of diabetes classification. Consequently, this study introduces a diabetes classification model that amalgamates SMOTE and Bayesian Optimization within the XGBoost framework to enhance recall and ROC-AUC performance in identifying diabetes patients.

2. METHOD

This study develops a diabetes classification model by integrating SMOTE, XGBoost, and Bayesian Optimization into a single structured process flow. The stages of this research include data collection, data preprocessing, data splitting into training and testing sets, handling class imbalance using SMOTE on the training data only, training the XGBoost model, performing hyperparameter optimization using Bayesian Optimization with Stratified 5-Fold Cross-Validation, and model performance evaluation. The overall research process flow is illustrated in Figure 1.

This framework is designed to provide a systematic overview for building an optimal diabetes classification model. This approach is expected to enhance the model's ability to detect diabetes, especially on

medical datasets with class imbalance. Further explanation of the research stages is presented in the following section.

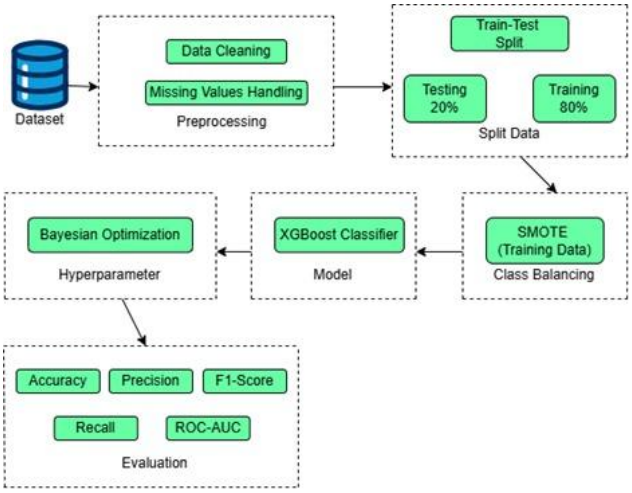

Figure 1. The classification research procedure

Data Collection

The dataset used in this research is sourced from publicly available open-source data on kaggle.com, accessible via the URL: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database, The same dataset has been used in several previous studies [8], [10], [12], [13]. The dataset consists of 768 instances with eight clinical features, namely: number of pregnancies, Glucose, blood pressure, skin thickness, insulin level, BMI, diabetic pedigree function, and age. The dataset also includes one target variable called Outcome. If the value is 1, it means the patient has diabetes, while a value of 0 indicates that the patient does not have diabetes. Of the 768 total instances, 500 belong to class 0 (non-diabetic) and 268 belong to class 1 (diabetic), resulting in an imbalanced class distribution. This situation is common in medical datasets, and it can make classification models favor the majority class, which can make it harder for them to discover disease cases reliably [14].

Data Preprocessing

The data preprocessing stage is an important step in constructing a machine learning model. The quality of the data has a big effect on how well the model works. The data we get often has missing values, inconsistent values, and values that aren't medically legitimate. So, data cleaning needs to be done before analysis [25]. Some of the attributes in the diabetes dataset had a value of 0, which is not medically legitimate. These values are considered missing, and the median value of each feature is used to fill them in. The median method is used because it works better with data that isn't normally distributed. This method can also help reduce the effect of outliers in the dataset [6].

Data Splitting

After the preprocessing step is done, the dataset is split into two parts: one for training and one for testing. The goal of this dataset split is to see how well the model can work with new data that it hasn't seen before [14]. There was an 80% training data ratio and a 20% testing data ratio in this investigation. The train_test_split function in the scikit-learn library divides the data apart at random. The argument random_state = 42 makes sure that the experiment may be repeated every time. The class balancing process with SMOTE is used only on the training dataset after splitting the data. This way, the testing dataset stays the same as the original and avoids data leakage, allowing for a fair and reliable evaluation of the model. After splitting, the training set consists of 614 records (400 class 0 and 214 class 1), and the testing set consists of 154 records. Therefore, the class distribution shown in Figure 3 reflects the training data distribution, rather than the distribution for the entire dataset of 768 instances.

Data Balancing using SMOTE

Data balancing helps solve the problem of uneven class distribution in the dataset. In this study's diabetes dataset, there are more non-diabetic cases than diabetic ones. This imbalance can make the model favor the majority class and miss positive cases. Balancing the data helps the model learn from both classes. To keep important information, this study uses oversampling. Oversampling increases the number of samples in the minority class, usually by creating new synthetic samples instead of just copying existing ones. Here, new synthetic samples for the minority class are generated with a specific algorithm to balance the data distribution [26].

The SMOTE method is used for oversampling. SMOTE is a way to fix data imbalance by making new synthetic samples from the minority class. It works by interpolation, which implies making new samples that are between existing samples in the feature space. SMOTE makes these new samples by interpolating between minority samples that are close to each other in the feature space [27]. Finding the k-nearest neighbors for each minority sample is the initial step. k is the number of data points that are closest to the sample. A parameter value of k = 5 is commonly chosen because it is thought to be enough to show the local data distribution without adding too much noise [27], [28].

After the nearest neighbors are obtained, one of the neighbors is randomly selected to form a new synthetic sample s between the two points according to equation 1.

$$x_{new} = x_i + \delta\ (x_{zi}\ -\ x_i) \tag{1}$$

Where x_i is a sample from the minority class, x_{zi} is its nearest neighbor, and δ is a random number in the range 0–1 that determines the interpolation position. This value is generated using a Pseudorandom Number Generator (PRNG), so the position of the synthetic data lies between and , resulting in new, more realistic data variations and helping the model to better learn minority class patterns [28], [29].

Model Exteme Gradient Boosting (XGBoost)

The classification model used in this study is XGBoost, an *ensemble learning* algorithm based on gradient boosting that is designed to improve prediction accuracy by iteratively combining multiple decision trees [30]. In each iteration, a new model is built to correct the prediction errors of the previous model using a gradient-based optimization approach, resulting in a more accurate and efficient model [20].

The objective function in XGBoost consists of a combination of the loss function and a regularization function, which aims to control model complexity. Mathematically, the objective function is formulated as follows:

$$L = \sum_{i=1}^{n} l(yi, \hat{y}i) + \sum_{k=1}^{k} \Omega(fk) \tag{2}$$

Where $l(yi, \hat{y}i)$ is a loss function that measures the difference between the actual and predicted values, while $\Omega(fk)$ is a regulation function. The regulation function in XGBoost is stated as follows:

$$\Omega(fk) = \gamma T + \frac{1}{2}\lambda \sum_{k=1}^{K} \omega_j^2 \tag{3}$$

Where T represents the leaves of the decision tree, ωj represents the weight of each leaf, while γ and λ are regularization parameters used to control model complexity. This approach helps reduce the risk of overfitting and improves the model's generalization ability to new data [16], [30].

Hyperparameter Optimization using Bayesian Optimization

The Bayesian Optimization approach was used to optimize the hyperparameters of the XGBoost model in order to make it work better. This method probabilistically simulates the connection between hyperparameter combinations and model performance. This makes the search for the best parameters faster than traditional methods like Grid Search or Random Search. The optimization process is generally expressed as:

$$x^* = argmax_x f(x) \tag{4}$$

Where *x* is a combination of hyperparameters and the optimized objective function *f(x)*. In this study, the objective function is focused on Recall and ROC-AUC because these two metrics are more relevant in medical classification, which emphasizes minimizing false negatives and model discrimination ability [23]. To maintain stability and generalization ability, optimization is performed using Stratified 5-Fold Cross Validation on the training data.

The hyperparameter search space is determined based on the general practice of XGBoost optimization on imbalanced data, including model complexity parameters, regularization, and subsampling mechanisms, as shown in Table 1.

Table 1. Xgboost hyperparameter search space

Parameter	Value Range	Description
n_estimators	300 - 600	Number of trees in the boosting model
max_depth	3 - 8	Maximum depth of each tree
learning_rate	0.01 – 0.1 (log-uniform)	Learning rate to control the contribution of each tree
min_child_weight	1 – 10	Minimum sample weight at child nodes
gamma	0 – 3	Minimum loss reduction to perform a split
subsample	0.7 – 1.0	Proportion of samples used in each iteration
colsample_bytree	0.7 – 1.0	Proportion of features used in each tree
reg_alpha	0 – 1	L1 regularization parameter
reg_lambda	1 – 5	L2 regularization parameter

These parameters play a role in controlling the balance between model complexity and generalization ability to obtain an optimal and stable combination of hyperparameters.

Model Evaluation

In this study, the model's predictions are compared against the actual class labels using a confusion matrix. True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the four parts of the confusion matrix. Individuals with diabetes who are appropriately diagnosed as such are referred to as TP, and individuals without diabetes who are correctly identified as non-diabetic are referred to as TN. FN happens when a diabetic patient is overlooked and labelled as non-diabetic, whereas FP happens when a non-diabetic patient is incorrectly labelled as diabetic. Because failure to identify a patient with diabetes may result in treatment being delayed, FN is seen as more significant than FP in medical classification. A number of performance metrics, each of which captures a distinct facet of model performance, are produced from these four variables, including accuracy, precision, recall, F1-score, and ROC-AUC.

Accuracy

Accuracy measures how many predictions the model got right, both positive and negative, out of all the predictions it made. The equation below shows how to calculate the Accuracy metric (5)

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} x\ 100\% \quad (5)$$

Precision

The precision of a model is a measure of how well it predicts favorable outcomes. It is the quantity of favorable forecasts that correspond to the initial label. The equation below shows how to calculate the Precision metric (6)

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

Recall

Recall measures how well a model identifies the true positive cases in a dataset. The equation below shows how to calculate the Recall metric (7)

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

F1-Score

The F1 score combines recall and precision into a single measure. It is especially useful when testing models on imbalanced datasets because it takes both precision and recall into account. This gives a clearer idea of how well a classification system works. The equation below shows how to calculate the F1-score metric (8)

$$F1 - Score = \frac{Precision \times Recall}{Precision+Recall} \quad (8)$$

ROC-AUC

The Receiver Operating Characteristic Area Under the Curve, or ROC-AUC, is another method to measure how well a model classifies data. The ROC curve shows how the True TPR and FPR change as you adjust the classification threshold. The area under this curve (AUC) tells you how well the model can

distinguish between people with diabetes and those without. A higher AUC means the model is better at telling the difference. The equation below shows how to calculate the ROC-AUC metric. (9)

$$AUC = \int_0^1 TPR(FPR)d(FPR) \quad (9)$$

The ROC-AUC metric provides an additional perspective in evaluating model performance, particularly in medical classification tasks where distinguishing between positive and negative cases is critical [13], [31].

3. RESULTS AND DISCUSSIONS

An Exploratory Data Analysis (EDA) was done to get a first look at the features of the dataset used in this investigation. This analysis is meant to find patterns in how data is spread out and how characteristics are related to each other. Both of these things could affect how effectively the classification model works. A heatmap was created using the Pearson correlation coefficient to show how the numerical features in the dataset are related to each other. Figure 2 shows how the features in the dataset are related to each other.

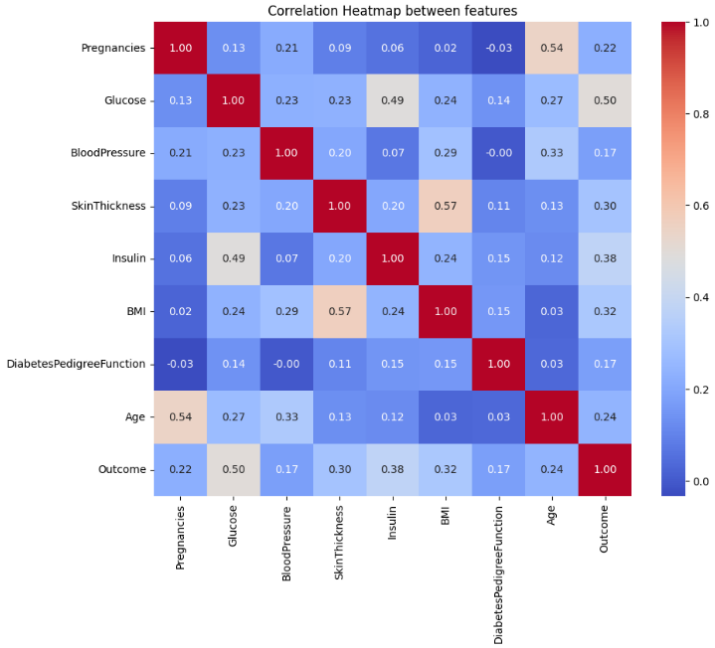

Figure 2. Correlation between features

Figure 2 shows how the features in the diabetes dataset are related to each other. Values for correlation go from -1 to 1. Values that are close to one mean that there is a strong positive association, and values that are close to minus one mean that there is a strong negative relationship. The features are less related to each other the closer the value goes to zero. This graph shows which traits are more closely linked to the target variable.

The research shows that the Glucose characteristic has the strongest relationship with the target variable (Outcome), with a value of about 0.50. This finding indicates a significant correlation between blood glucose levels and the probability of an individual acquiring diabetes. The Insulin and BMI features also show positive correlations with the target variable, which means that metabolic parameters are crucial in raising the risk of diabetes.

In contrast, several other features, such as Blood Pressure and Diabetes Pedigree Function, show relatively weaker correlations with the target variable. Nevertheless, these features are retained in the modeling process because machine learning algorithms are capable of capturing non-linear relationships that may not be evident through simple correlation analysis.

Following the data exploration stage, the next step involves conducting a class distribution analysis to identify potential data imbalance in the dataset. The class distribution of the dataset before and after applying SMOTE is show in Figure 3.

Figure 3. Class distribution before and after applying SMOTE

The difference in how data is spread out between class 0 (non-diabetic) and class 1 (diabetes) is shown in Figure 3. There are clearly more samples in the group of people who don't have diabetes than in the group of people who do. This mismatch in the number of classes can make classification algorithms favor the majority class, which makes it harder for the model to detect people with diabetes. This problem can be fixed with SMOTE. This method makes fresh synthetic samples for the minority class by selecting samples that are close to each other in the feature space and putting them together. This method spreads out the data more evenly without losing any significant information from the dataset.

After balancing the data distribution, the next step is to use the XGBoost method to develop a classification model. We tested three different setups to see how they affected model performance the basic XGBoost model, XGBoost with SMOTE to handle class imbalance, and an optimized XGBoost model that used both SMOTE and Bayesian Optimization. The comparison of the results obtained from these scenarios is presented in Table 2.

Table 2. Model performance comparison

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
XGBoost baseline	0.88	0.82	0.85	0.84	0.947
XGBoost + SMOTE	0.88	0.80	0.87	0.83	0.946
XGBoost + SMOTE + Bayesian optimization	0.88	0.79	0.91	0.84	0.955

Table 2 presents the initial XGBoost model's performance, which exhibited an accuracy of 0.88 and a recall of 0.85. After applying SMOTE to address class imbalance, recall improved slightly from 0.85 to 0.87, while ROC-AUC remained at 0.946. This indicates that SMOTE contributes to better minority class detection, but the improvement is modest without hyperparameter tuning.

The optimized XGBoost model, which integrates SMOTE and Bayesian Optimization, demonstrates improved performance across several evaluation metrics. In particular, the recall value increases from 0.85 in the baseline model to 0.91, indicating that the optimized model is more effective in identifying patients with diabetes.

Similarly, the ROC-AUC value increases from 0.947 in the baseline model to 0.955 in the optimized model, suggesting that the model has a better ability to distinguish between diabetic and non-diabetic cases. Although the precision value slightly decreases from 0.82 (baseline) to 0.79 (optimized), the higher recall indicates that the model becomes more sensitive in detecting diabetes cases. This improvement is particularly important in medical classification tasks, where failing to detect positive cases may delay diagnosis and treatment. Figure 4 illustrates the comparison of model performance across the three modelling scenarios.

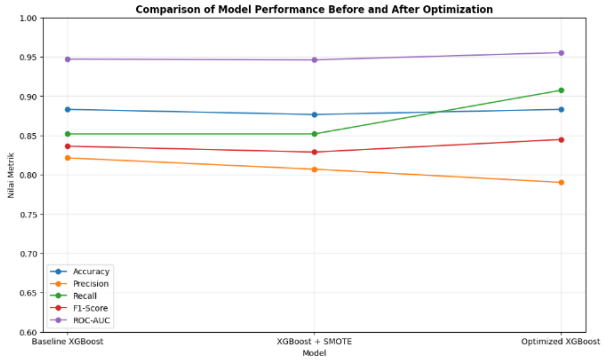

Figure 4. Performance comparison of the classification models across different experimental scenarios

Figure 4 supports the trends observed in Table 2. Recall increases steadily across the three scenarios, while accuracy and F1-score remain relatively stable. Precision decreases slightly, which highlights the trade-off between sensitivity and specificity in imbalanced medical classification tasks. A confusion matrix, as illustrated in Figure 5, is also utilized to evaluate the model's classification performance.

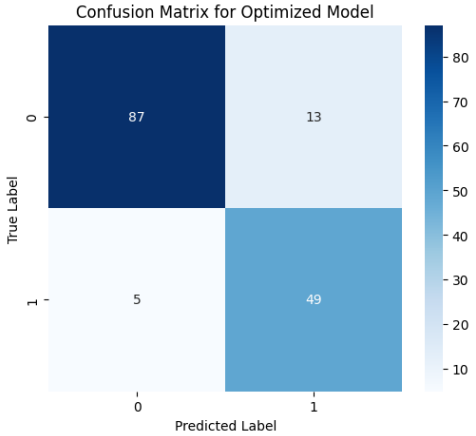

Figure 5. Confusion matrix optimization xgboost model

The confusion matrix showed that the model accurately recognized 87 non-diabetic data points (True Negative) and 49 diabetes data points (True Positive). There were also 13 false positives and 5 false negatives. There aren't many false negatives, which means the model can reliably find people who really have diabetes. This is critical for finding diseases early because missing positive instances might make diagnosis and therapy take longer.

The findings of this study indicate that the combination of data imbalance correction and hyperparameter optimization significantly improves the classification model's performance. By utilizing SMOTE, class distributions are balanced, allowing the model to better recognize patterns within the minority class. Concurrently, Bayesian Optimization assists the model in identifying optimal hyperparameter settings, which subsequently improves its predictive abilities. The observed increase in recall highlights the effectiveness of the suggested approach in detecting diabetes cases, a crucial element in medical classification applications where the ability to identify positive instances is essential for enabling early diagnosis and intervention.

These findings align with prior research concerning diabetes classification through the application of machine learning methodologies. As an illustration, Iparraguirre-Villanueva et al. [8] integrated multiple classification algorithms with the SMOTE technique, observing that the K-Nearest Neighbor (KNN) model attained an accuracy of 79.6%, accompanied by a recall value of 79%. Likewise, Islam et al. [10] employed the XGBoost algorithm in conjunction with the ADASYN oversampling method, achieving an accuracy of 81% and an AUC value of 0.84. Talukder et al. [11] similarly assessed various machine learning algorithms,

discovering that the Random Forest model attained an accuracy of 85.5% and a recall of 81.3%. In contrast to these investigations, the model proposed in this study demonstrated an accuracy of 0.88, a recall value of 0.91, and a ROC-AUC value of 0.955, thereby indicating a more robust capacity for diabetes case detection. These results collectively suggest that the integration of SMOTE, XGBoost, and Bayesian Optimization constitutes an efficacious strategy for constructing machine learning-based systems for disease prediction, especially when addressing medical datasets characterized by imbalanced class distributions.

4. CONCLUSION

This study demonstrates that the integration of SMOTE, the XGBoost algorithm, and Bayesian Optimization for hyperparameter tuning enhances diabetes classification models. SMOTE makes synthetic samples for the minority class to assist in balancing the class distribution. This way, the model can learn more from the data. Hyperparameter optimization helps the model find better settings for its parameters, which makes it better at making predictions. The optimized model has an accuracy of 0.88, a precision of 0.79, a recall of 0.91, an F1-score of 0.84, and a ROC-AUC of 0.955. These results show that the model can accurately discern the difference between people with diabetes and people without diabetes. It is especially good at finding people who have diabetes. In general, combining data balance with hyperparameter optimization can make machine learning models for medical classification much better. This method could also help make systems for finding diabetes early more reliable. Future studies could evaluate the model on larger and more diverse datasets or explore the incorporation of feature selection or more sophisticated ensemble techniques to enhance generalization.

REFERENCES

[1] World Health Organization, "Diabetes – Key Facts," *Online*. https://www.who.int/news-room/fact-sheets/detail/diabetes (diakses Mar 05, 2026).

[2] S. Fan, M. S. D. Wykes, W. E. Lin, R. L. Jones, A. G. Robins, dan P. F. Linden, *Effects of synbiotics surpass probiotics alone in improving type 2 diabetes mellitus: a randomized, double-blind, placebo-controlled trial 4*. European Society for Clinical Nutrition and Metabolism, 2020.

[3] A. Yuli *et al.*, *Profil Keselamatan dan Kesehatan Kerja Nasional Indonesia Tahun 2022*. 2022.

[4] M. Salmi, D. Atif, D. Oliva, A. Abraham, dan S. Ventura, *Handling imbalanced medical datasets : review of a decade of research*, vol. 57, no. 10. Springer Netherlands, 2024.

[5] H. I. Classification, "A Comprehensive Review on Machine Learning in Healthcare Industry: Classification, Restrictions, Opportunities and Challenges," *Sensors*, vol. 23, 2023, doi: https://doi.org/10.3390/s23094178.

[6] P. Sampath, G. Elangovan, K. Ravichandran, V. Shanmuganathan, S. Pasupathi, dan T. Chakrabarti, "Robust diabetic prediction using ensemble machine learning models with synthetic minority over- sampling technique," *Sci. Rep.*, vol. 14, hal. 1–15, 2024, doi: https://doi.org/10.1038/s41598-024-78519-8 1.

[7] K. Abnoosian, R. Farnoosh, dan M. H. Behzadi, "Prediction of diabetes disease using an ensemble of machine learning multi - classifier models," *BMC Bioinformatics*, hal. 1–24, 2023, doi: 10.1186/s12859-023-05465-z.

[8] O. Iparraguirre-villanueva, K. Espinola-linares, R. Ornella, F. Castañeda, dan M. Cabanillas-carbonell, "Application of Machine Learning Models for Early Detection and Accurate Classification of Type 2 Diabetes," *diagnostics Artic.*, vol. 13, 2023, doi: https://doi.org/10.3390/ diagnostics13142383.

[9] P. S. Moon, Y. Chavan, dan P. P. S. Moon, "Machine Learning approach for Diabetes Prediction using Pima Dataset Machine Learning approach for Diabetes Prediction using Pima Dataset," *Int. Con- ference Inf. Manag. Mach. Intell. (ICIMMI 2023)*, no. March, 2023, doi: 10.1145/3647444.3652479.

[10] H. T. Letters, I. Tasin, T. U. Nabil, S. Islam, dan R. Khan, "Diabetes prediction using machine learning and explainable AI," *Healthc. Technol. Lett. Diabetes*, no. November 2022, hal. 1–10, 2023, doi: 10.1049/htl2.12039.

[11] A. Talukder, M. Islam, dan A. Uddin, "Toward reliable diabetes prediction : Innovations in data engineering and machine learning applications," *Digit. Heal.*, 2024, doi: 10.1177/20552076241271867.

[12] N. P. Shetty, J. Shetty, V. Hegde, S. Dattatray, dan M. Kv, "A machine learning-based clinical decision support system for effective stratification of gestational diabetes mellitus and management through Ayurveda," *J. Ayurveda Integr. Med.*, vol. 15, no. 6, hal. 101051, 2024, doi: 10.1016/j.jaim.2024.101051.

[13] F. Rahman, S. Hossain, dan J. Tiang, "Diabetes Prediction Using Feature Selection Algorithms and Boosting-Based Machine Learning Classifiers," *diagnostics*, vol. 115, hal. 1–24, 2025, doi: https://doi.org/10.3390/diagnostics15202622.

[14] X. Feng, Y. Cai, dan R. Xin, "Optimizing diabetes classification with a machine learning - based framework," *BMC Bioinformatics*, hal. 1–20, 2023, doi: 10.1186/s12859-023-05467-x.

[15] A. G. Coimbra, C. G. Oliveira, dan M. P. Libório, "Approaches for handling imbalanced data used in machine learning in the healthcare field : A case study on Chagas disease database prediction," *PLoS One 20*, hal. 1–19, 2025, doi: 10.1371/journal.pone.0320966.

[16] N. Anjum, "Machine learning and artificial intelligence in type 2 diabetes prediction: a comprehensive 33-year bibliometric and literature analysis," *Front. Digit. Heal. 7*, vol. 7, 2025, doi: 10.3389/fdgth.2025.1557467.

[17] A. E. Setiawan, S. Rustad, A. Syukur, dan M. A. Soeleman, "SMOTE-ENN Resampling to Optimize Diabetes Prediction in Imbalanced Data," *IIETA*, vol. 30, no. 5, hal. 1163–1176, 2025, doi: https://doi.org/10.18280/isi.300505 Received:

[18] H. Shao, X. Liu, D. Zong, dan Q. Song, "Optimization of diabetes prediction methods based on combinatorial balancing algorithm," *Nutr. Diabetes*, no. July, hal. 1–13, 2024, doi: 10.1038/s41387-024-00324-z.

[19] Rofik dan Jumanto, "Evaluation of Ridge Classifier and Logistic Regression for Customer Churn Prediction on Imbalanced Telecommunication Data," *Sci. J. Informatics*, vol. 12, no. 2, hal. 311–326, 2025, doi: 10.15294/sji.v12i2.24620.

[20] W. Li, Y. P. Id, dan K. Peng, "Diabetes prediction model based on GA- XGBoost and stacking ensemble algorithm," *PLoS One*, hal. 1–29, 2024, doi: 10.1371/journal.pone.0311222.

[21] M. R. Khurshid, S. Manzoor, T. Sadiq, dan L. H. Id, "Unveiling diabetes onset : Optimized XGBoost with Bayesian optimization for enhanced prediction," *PLoS One*, vol. 1, hal. 1–29, 2025, doi: 10.1371/journal.pone.0310218.

[22] B. Toleva, I. Atanasov, dan I. Ivanov, "An Effective Methodology for Diabetes Prediction in the Case of Class Imbalance,"

bioengineering, vol. 12, no. 35, hal. 1–17, 2025, doi: https://doi.org/10.3390/bioengineering12010035.

[23] D. Yan, X. Li, Y. Wang, dan Z. Cai, "Optimized prediction of diabetes complications using ensemble learning with Bayesian optimization : a cost-ef fi cient laboratory-based approach," *Front. Endocrinol*, no. June, hal. 1–18, 2025, doi: 10.3389/fendo.2025.1593068.

[24] B. Bischl, J. Richter, M. Becker, M. Binder, dan T. Pielok, "Hyperparameter optimization : Foundations , algorithms , best practices , and open challenges," *Wiley*, no. July 2021, hal. 1–43, 2023, doi: 10.1002/widm.1484.

[25] M. Mujahid *et al.*, "Data oversampling and imbalanced datasets : an investigation of performance for machine learning and feature engineering," *J. Big Data*, 2024, doi: 10.1186/s40537-024-00943-4.

[26] M. Kivrak, U. Avci, dan H. Uzun, "The Impact of the SMOTE Method on Machine Learning and Ensemble Learning Performance Results in Addressing Class Imbalance in Data Used for Predicting Total Testosterone Deficiency in Type 2 Diabetes Patients," *diagnostics*, 2024.

[27] Y. Yang, H. A. Khorshidi, dan U. Aickelin, "A review on over-sampling techniques in classi fi cation of," *Front. Digit. Heal. 7*, no. July, 2024, doi: 10.3389/fdgth.2024.1430245.

[28] S. Matharaarachchi, M. Domaratzki, dan S. Muthukumarana, "Machine Learning with Applications Enhancing SMOTE for imbalanced data with abnormal minority instances," *Mach. Learn. with Appl.*, vol. 18, no. December 2023, hal. 100597, 2024, doi: 10.1016/j.mlwa.2024.100597.

[29] Y. Li, Y. Yang, P. Song, L. Duan, dan R. Ren, "An improved SMOTE algorithm for enhanced imbalanced data classification by expanding sample generation space," *Sci. Rep.*, hal. 1–21, 2025.

[30] T. Chen dan C. Guestrin, "XGBoost: A scalable tree boosting system," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, hal. 785–794, 2016, doi: 10.1145/2939672.2939785.

[31] D. Chicco dan G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification," *BioData Min.*, hal. 1–23, 2023, doi: 10.1186/s13040-023-00322-4.

.