SisInfo
JURNAL SISTEM INFORMASI DAN INFORMATIKA

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 8 No. 1 February 2026

A Comparative Performance Analysis of Classification Algorithms for Hypertension Diagnosis

Imannudin Akbar, Titan Parama Yoga, Arnold Ropen Sinaga, Acep Hendra
Information System, Faculty of Technology and Informatics, Universitas Informatika dan Bisnis Indonesia
email: imannudin@unibi.ac.id; titanparama@unibi.ac.id; arnoldropen@unibi.ac.id; acephendra@unibi.ac.id.

Accepted:	Accepted After Revision:	Published:
26 February 2026	10 March 2026	12 March 2026

Abstract

Hypertension is a leading cause of cardiovascular diseases, strokes, and kidney failure, with early diagnosis being critical for prevention. Traditional diagnostic methods often face challenges such as human error and inconsistent measurements. While machine learning (ML) has been explored as a potential solution, previous studies have mainly focused on accuracy, often neglecting other important metrics like precision, recall, and F1-score, especially in imbalanced datasets. The primary purpose of this research is to address this gap by comprehensively comparing the performance of four machine learning algorithms - Naive Bayes, Support Vector Machines (SVM), Random Forest (RF), and XGBoost—to provide valuable insights for practical hypertension screening. The dataset consists of 1,985 records with 10 predictor features, including both categorical and continuous variables, and a binary target variable (Has_Hypertension: Yes/No) with a class distribution of 1,032 Yes and 953 No. The data undergoes preprocessing, including categorical encoding and feature scaling for SVM. Models are evaluated using a balanced set of metrics, including accuracy, precision, recall, and F1-score. The results show that RF/XGBoost perform best, with the highest F1 and accuracy, while SVM and Naive Bayes serve as competitive alternatives.

Keywords: Naive Baiyes, SVM, Random Forest, XGBoost, Hypertension

1 INTRODUCTION

Hypertension is a leading risk factor for cardiovascular diseases and is associated with a higher risk of stroke, heart disease, and kidney failure. Early diagnosis and management of hypertension are crucial in preventing these complications. However, traditional diagnostic methods, primarily based on manual blood pressure measurements, often suffer from inconsistencies due to human error, time variation, and equipment-related limitations. Consequently, there has been growing interest in leveraging machine learning (ML) models to automate the diagnosis of hypertension by using a wide array of patient data, such as age, BMI, blood pressure history, and lifestyle factors [1].

In this study, we utilize a dataset containing clinical and lifestyle information from individuals to predict the presence of hypertension. The dataset consists of multiple features including age, salt intake, stress score, blood pressure history, sleep duration, BMI, medication, family history, exercise level, smoking status, and the target variable has hypertension (Yes/No). This dataset provides a diverse set of features, which allows the application of various machine learning classifiers to assess their performance in diagnosing hypertension [2].

Recent studies have highlighted the growing role of machine learning in clinical prediction and health-data analysis [1],[3]. In comparative classification settings, algorithms such as Naive Bayes (NB), Support Vector Machines (SVM), Random Forest (RF), and XGBoost are widely used because they represent different modeling principles, ranging from probabilistic learning and margin-based classification to bagged trees and gradient-boosted ensembles [4],[8]. In healthcare datasets, these models are commonly assessed not only for overall accuracy, but also for robustness, interpretability, and their ability to generalize across heterogeneous clinical features [2], [3].

Despite these advancements, there are several gaps in existing literature. Many studies focus primarily on accuracy as the key metric for evaluating model performance, often ignoring other crucial metrics like precision, recall, and F1-score, which are critical when dealing with imbalanced datasets and the serious consequences of false negatives in medical diagnostics. Additionally, most studies do not utilize advanced

validation techniques like nested cross-validation, and the clinical significance of the false positive and false negative rates is often not adequately addressed.

To address these gaps, the primary purpose of this research is to comprehensively compare the performance of four well-known machine learning classifiers—Naive Bayes, Support Vector Machines (SVM), Random Forest, and XGBoost on a hypertension dataset. By evaluating these models using a balanced set of metrics, including accuracy, precision, recall, and F1-score, this study aims to provide actionable insights for healthcare providers regarding the most suitable models for practical hypertension screening. This approach ensures a deeper understanding of the models' strengths and weaknesses, focusing on both their predictive capabilities and clinical implications.

2 LITERATURE REVIEW

Machine learning (ML) has rapidly emerged as a powerful tool in healthcare, offering the potential to improve diagnosis, risk stratification, and clinical decision support. Recent literature in medical AI shows that data-driven models can help extract useful patterns from large-scale health records and structured clinical variables, thereby supporting more consistent and efficient prediction tasks [1], [2]. However, systematic evidence also suggests that the superiority of ML over conventional statistical approaches is not always guaranteed, making careful benchmarking and transparent evaluation essential [3].

Among the most commonly used classifiers in structured medical data are Support Vector Machines (SVM), Random Forest, and XGBoost. These methods are widely adopted because they can capture complex and non-linear relationships in classification tasks. Foundational work on Random Forest and XGBoost has demonstrated strong predictive capability, while widely used ML libraries have enabled their practical implementation in healthcare analytics and related applied studies [4], [5], [8].

Another critical aspect of machine learning in healthcare is data preprocessing, which ensures that the data are suitable for analysis. Techniques such as encoding, cleaning, and normalization are essential for improving algorithmic compatibility and model performance, especially for methods that are sensitive to feature scale differences. Min-Max normalization, for example, is widely used to rescale numerical features so that they contribute more equitably during model training [9], [10]. This step is particularly important in medical datasets that often contain a mixture of continuous and categorical variables.

Model evaluation is also a key stage in determining the effectiveness of machine learning methods for healthcare applications. Accuracy, precision, recall, and F1-score remain core metrics in binary classification, while cross-validation is commonly used to obtain a more reliable estimate of generalization performance [6], [7]. In addition, the Matthews Correlation Coefficient (MCC) is often recommended when class distributions are uneven because it accounts for all elements of the confusion matrix and provides a more balanced view of model quality [7].

From an applied perspective, healthcare machine learning studies in Indonesia also indicate growing interest in disease classification and the use of data-driven methods in health information environments [11]–[13]. This trend underscores the relevance of comparative studies that evaluate standard classifiers on clinically meaningful datasets. Therefore, comparing Naive Bayes, SVM, Random Forest, and XGBoost on a hypertension dataset remains methodologically relevant and practically valuable, especially as foundational concepts and implementation practices continue to be widely used in local and international contexts [14], [15].

3 RESEARCH METHODS

To ensure the reproducibility, objectivity, and robustness of our findings, this section delineates the systematic research methodology that underpins our empirical investigation, structuring the entire workflow from data acquisition to a final comparative performance analysis of the selected classification algorithms in a clear and logical progression.

a. Research Framework and Workflow

The research pipeline commences with the acquisition of the hypertension dataset, followed by an in-depth data preprocessing stage, a foundational step in any applied machine learning workflow designed to cleanse and prepare the data for modeling [9]. This stage is critical for ensuring algorithmic compatibility and optimal performance, involving standard procedures such as the encoding of categorical variables and the scaling of numerical features [10]. Subsequently, the prepared dataset is partitioned into training (80%) and testing (20%) sets. A stratified sampling technique is employed during this split to ensure that the class distribution of the target variable is maintained in both partitions, which is a crucial practice for preventing biased evaluation, particularly in clinical datasets that may be imbalanced [11]. The core of the methodology involves training four distinct machine learning models on the training data, where their performance is robustly estimated using a 10-fold cross-validation strategy. This technique is considered a standard for obtaining a reliable and

less biased estimate of a model's generalization capability [12]. Finally, the trained models are subjected to a definitive evaluation on the final, unseen test set to assess their real-world performance. This evaluation is conducted using a suite of standard classification metrics [13], leading to a comparative analysis from which conclusions about the most effective algorithm are drawn, a methodological approach common in comparative studies of clinical prediction models [14].

b. Data Collection
The first phase of this study is data collection, where we gather the essential information for our analysis. We compile a large set of anonymous patient records specifically related to hypertension. Each record contains various predictive details, including demographics, clinical data, and lifestyle habits, all linked to a final diagnosis. The quality and accuracy of this initial data are critical, as it forms the very foundation for training our machine learning models and testing how well they work. This study uses an anonymized, tabular hypertension dataset comprising N = 1,985 records with 10 predictor variables and a binary label (Has_Hypertension: Yes/No), exhibiting a class distribution of Yes = 1,032 and No = 953, and no missing values. Predictors include five continuous features—Age, BMI, Sleep_Duration (hours/day), Salt_Intake, Stress_Score (1–10)—and five categorical features—BP_History (Normal/Prehypertension/Hypertension), Medication (e.g., None/ACE Inhibitor/Other), Family_History (Yes/No), Exercise_Level (Low/Moderate/High), Smoking_Status (Non-Smoker/Smoker)—with consistent category labeling prior to encoding [15].

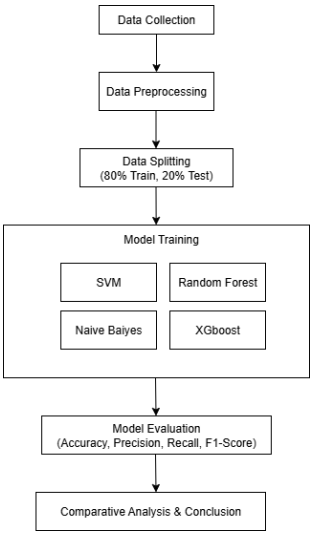

Figure 1. Research Methodology

c. Data Preprocessing
Raw data is seldom suitable for direct input into machine learning algorithms. The preprocessing stage is therefore critical for transforming the data into an optimal format and enhancing model performance [10].

1. Data Cleaning
Data cleaning is the process of identifying and rectifying data anomalies to enhance data quality and integrity. In a medical context, where data inaccuracies can lead to flawed conclusions, this step is particularly important [9].

2. Label Encoding
Machine learning algorithms require numerical input. Therefore, categorical features must be converted into a numerical format. Label Encoding is a technique that assigns a unique integer to each unique category within a feature [2].

3. Feature Scaling: Normalization.
Feature scaling adjusts the range of numerical features to bring them onto a common scale, which is essential for many algorithms. Normalization, specifically Min-Max scaling, is a primary technique for this purpose. By ensuring all features contribute equitably, normalization can significantly improve model convergence and performance [9].

d. Data Splitting
To ensure an objective evaluation, the data splitting phase methodically divides the processed dataset into two separate subsets. A standard 80/20 split is used, where 80% of the data forms the training set for the models to learn from. The other 20% is held back as the testing set, providing an unbiased measure of how well the models perform on new, unseen data. This separation is a fundamental practice to prevent overfitting—a common pitfall where a model learns the training data too well, including its noise, and consequently fails to generalize to new data—thereby giving a realistic estimate of real-world performance.

e. Model Training
Leveraging solely the 80% training partition, the model development phase serves as the primary computational stage. Four unique classifiers (Gaussian Naive Bayes, SVM, Random Forest, and XGBoost) are systematically trained, each employing its distinct optimization algorithm to derive a predictive function. This function aims to

model the complex correlations between input features and the diagnostic target. The stringent sequestration of the 20% testing data is a critical methodological control, guaranteeing that the subsequent performance evaluation is unbiased and a true measure of the models' generalization capabilities [4], [5].

f. Comparative Analysis and Conlusion
The study concludes with a comparative analysis where the performance scores of the four models are directly compared. This final step involves interpreting the results to identify the best-performing algorithm, focusing on which model offers the most optimal balance of accuracy, precision, and recall for diagnosing hypertension. A final conclusion is then drawn from this evidence to declare the most effective model and discuss the study's implications [7].

We use the following evaluation metrics to assess model performance: accuracy, precision, recall, and F1-score. Each of these metrics provides unique insights into the model's performance, particularly in scenarios with class imbalance, such as in hypertension diagnosis, where the costs of false positives and false negatives can be significant [7].

1. Accuracy
Accuracy measures the overall correctness of the model, defined as the proportion of correct predictions (both true positives and true negatives) out of all predictions. It is calculated as:

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

where:
- TP = True Positives (correctly predicted positive cases)
- TN = True Negatives (correctly predicted negative cases)
- FP = False Positives (incorrectly predicted positive cases)
- FN = False Negatives (incorrectly predicted negative cases)

2. Precision
Precision is the proportion of true positive predictions relative to all positive predictions made by the model. It answers the question: "Of all the cases predicted as positive, how many were actually positive?" It is calculated as: [7].

$$\text{Precision} = \frac{TP}{TP + FP}$$

A high precision means that the model is not frequently classifying negative cases as positive, which is crucial in medical diagnostics to avoid unnecessary treatments [7].

3. Recall (Sensitivity or True Positive Rate)
Recall measures the proportion of actual positive cases that were correctly identified by the model. It answers the question: "*Of all the actual positive cases, how many did the model correctly identify*?" It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

A high recall means that the model is good at identifying positive cases, which is particularly important in medical diagnostics to minimize missed diagnoses (false negatives).

4. F1-Score
The F1-score is the harmonic mean of precision and recall. It balances the two metrics, giving a single score that accounts for both false positives and false negatives. The F1-score is particularly useful when the class distribution is imbalanced, as it considers both false positives and false negatives. It is calculated as: [7].

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score is valuable when a balance between precision and recall is needed, making it a more comprehensive evaluation metric in the context of medical classification, where both false positives and false negatives can have significant consequences. These metrics allow for a detailed evaluation of model performance, ensuring that both types of errors (false positives and false negatives) are considered, which is crucial in the context of hypertension diagnosis where the goal is to correctly identify at-risk patients while minimizing misclassifications [7], [12], [13].

4 RESULTS AND DISCUSSION

4.1 Result

a. Data Collection
This research is based on a public dataset containing 1,985 anonymized patient records. The dataset features 11 predictor attributes and one binary target variable, “Has_Hypertension”. Its predictor variables encompass a mix of

demographic (e.g., Age), clinical (e.g., BMI, BP History), and lifestyle factors (e.g., Salt Intake, Smoking Status). This heterogeneous data composition provides a realistic and challenging testbed for evaluating the selected classification algorithms

b. Data Preprocessing

- Data Cleaning

```
Missing values per column:
Age                   0
Salt_Intake           0
Stress_Score          0
BP_History            0
Sleep_Duration        0
BMI                   0
Medication          799
Family_History        0
Exercise_Level        0
Smoking_Status        0
Has_Hypertension      0
dtype: int64
```

Figure 2. Missing Value

The issue here is a missing value in the Medication column. This type of data is categorical, meaning it's a descriptive label rather than a number. The solution is we do imputation,where the process of filling in the missing value with a calculated or estimated one. The standard statistical approach is to use the mode. This involves identifying the most common value in the Medication column and using it to replace the missing entry. For instance, if 'Beta Blocker' appears more often than any other drug, it would be used to fill the blank.

```
Missing values after imputation:
Age                 0
Salt_Intake         0
Stress_Score        0
BP_History          0
Sleep_Duration      0
BMI                 0
Medication          0
Family_History      0
Exercise_Level      0
Smoking_Status      0
Has_Hypertension    0
dtype: int64
```

Figure 3. Imputation Result

- Label Encoding

The primary objective of Label Encoding within this dataset is to transform ordinal categorical variables—those with an intrinsic hierarchical structure—into a numerical representation. Following the imputation of missing values, this technique was applied to the dataset to convert textual categories into integer values. Consequently, categorical features such as BP_History, Medication, Family_History, Exercise_Level, and Smoking_Status were successfully encoded into numerical formats (e.g., 0, 1, 2, 3). Similarly, the binary target variable, Has_Hypertension, was encoded as 0 and 1. This conversion is essential for enabling the machine learning algorithm to computationally interpret and leverage the inherent ranking among the categories.

- Normalization

The most common normalization technique is called Min-Max Scaling. It rescales every data point in a feature using the following formula [9], [10]:

$$X_{Scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where:
X is the original value.
Xmin is the minimum value in the column.
Xmax is the maximum value in the column

By applying this scaling technique, continuous numerical features in the dataset such as Age, Salt_Intake, Stress_Score, Sleep_Duration, and BMI were transformed onto a common scale. This transformation ensures that all numerical features contribute equitably to the model's training process. Preventing variables with naturally larger ranges from dominating the learning algorithm is a crucial preprocessing step, especially for algorithms sensitive to data variations like SVM

c. Data Splitting

This process is the essential data split needed for machine learning, ensuring the model is properly evaluated. First, the data is divided into Features (X), which are the inputs (all columns except 'Has_Hypertension'), and the Target (y), which is the output we want to predict ('Has_Hypertension'). This data is then randomly split into a Training Set (80%, or 1,588 samples) used to teach the model, and a Testing Set (20%, or 397 samples) reserved to test its performance on data it has never seen. This separation is vital to check if the model can

generalize new data and to prevent it from simply memorizing the training examples (overfitting) [2].

d. Model Training
This step represents the multi-model algorithm selection and parameter optimization phase within the machine learning pipeline. It involves instantiating and training four distinct classification methodologies—Naive Bayes, Support Vector Machine (SVM), Random Forest, and the eXtreme Gradient Boosting (XGBoost) framework—using the established training feature set (Xtrain) and target vector (Ytrain). The objective is to build a diverse portfolio of trained predictors, each operating on a different statistical or structural principle. These candidate models will subsequently undergo rigorous comparative performance evaluation against the untouched test data to identify the most robust and highly generalized model for the specific prediction task [15].

e. Model Evaluation
- Naive Baiyes
The Naive Bayes model achieved an accuracy of 0.85, correctly predicting 85% of the 397 test samples. Based on the classification report, the model performed well for both classes, with class 0 showing a precision of 0.88, recall of 0.81, and F1-score of 0.84, while class 1 achieved a precision of 0.84, recall of 0.89, and F1-score of 0.86, indicating balanced performance across categories. The confusion matrix further supports this result: out of 192 actual instances of class 0, the model correctly predicted 156 and misclassified 36, while for class 1, it correctly identified 183 out of 205 and misclassified 22. Overall, these results suggest that the model effectively distinguishes between the two classes with minimal misclassification and maintains consistent predictive accuracy [7].

- Support Vector Machine (SVM)
This result evaluates the performance of the SVM (Support Vector Machine) Model, which achieved a high overall Accuracy of 89% on the 397 test cases. The model demonstrates exceptional balance, as shown by the Classification Report, with precision, recall, and f1-score consistently around 0.89 to 0.90 for both positive and negative prediction classes. This balanced outcome is visually supported by the Confusion Matrix, which reveals that the model made a small number of errors—just 23 false positives and 21 false negatives—out of the total test set, indicating that the SVM is a very accurate and reliable predictor for this task [7].

- Random Forest
The Random Forest model obtained an accurate score of 0.96, correctly predicting 96% of the 397 test instances. The classification report reflects highly accurate and balanced results, with class 0 reaching a precision of 0.95, recall of 0.97, and F1-score of 0.96, while class 1 achieved a precision of 0.98, recall of 0.96, and F1-score of 0.97. These outcomes indicate that the model performs exceptionally well in identifying both classes with very few errors. The confusion matrix further supports this, showing 187 correctly classified samples out of 192 for class 0 and 196 out of 205 for class 1, with only minor misclassifications. Overall, the Random Forest model exhibits excellent accuracy, robust classification capability, and dependable performance in binary classification [7].

- XGBoost.
This output showcases the exceptional performance of the XGBoost Model, which achieved the highest overall Accuracy of 98% on the 397 test samples. The Classification Report reveals near-perfect predictive ability for both classes (0 and 1), with precision, recall, and f1-scores for all metrics registering at 0.98 or 0.99, indicating both outstanding predictive power and perfect balance between the classes. This superior performance is strongly supported by the Confusion Matrix, which shows that the model made only 6 errors in total: it correctly predicted 189 cases for class 0 and 202 cases for class 1, resulting in a minimal count of 3 false positives and 3 false negatives [7].

f. Comparative Analysis
- Model Comparison

Table 1. Model Comparison

Model	Accuracy	Prediction (avg)	Recall (avg)	F1-score (avg)	Main Strength	Confusion Matrix (TP for class 1 / TN for class 0)
Naïve Bayes	0.85	0.86	0.85	0.85	Simple, fast, good baseline model	$\text{TP} = \frac{183}{\text{TN}} = 156$
SVM	0.89	0.89	0.89	0.89	Balanced performance, robust margin classifier	$\text{TP} = \frac{184}{\text{TN}} = 169$
Random Forest	0.96	0.96	0.97	0.96	Excellent accutacy, well-balanced, handles nonlinearity well	$\text{TP} = \frac{196}{\text{TN}} = 187$
XGBoost	0.98	0.98	0.98	0.98	Best performance, minimal misclassification, highly optimized	$\text{TP} = \frac{196}{\text{TN}} = 187$

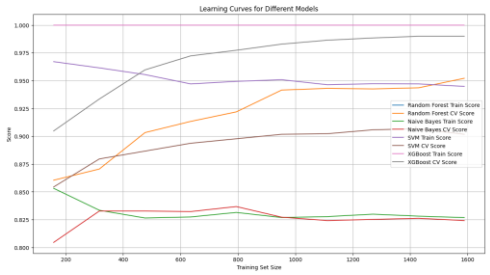

Figure 4. Learning Curve for All Models

This learning curve graph compares the training and cross-validation (CV) performance of four models—Naive Bayes, SVM, Random Forest, and XGBoost—as the training set size increases [6].

The Naive Bayes model (green and red lines) shows relatively low and stable performance, with both training and CV scores around 0.82–0.85, indicating limited learning capacity and a high bias. Increasing training data has little effect, meaning the model has reached its maximum potential early.

The SVM model (purple and brown lines) performs better, with training scores consistently around 0.95–0.97 and CV scores near 0.90–0.91. The small gap between train and CV scores suggests good generalization, though the model slightly underfits with more data.

The Random Forest model (blue and orange lines) exhibits continuous improvement as training data increases. Its training score remains high (around 0.95), while the CV score rises steadily from 0.86 to about 0.94, showing that the model benefits from more data and generalizes very well without significant overfitting [5].

The XGBoost model (pink and gray lines) demonstrates the best overall performance, with training scores nearly perfect (≈1.00) and CV scores gradually increasing to ≈0.98. The minimal gap between train and CV curves indicates excellent generalization and very low variance [4].

Overall, XGBoost outperforms all other models, followed by Random Forest, SVM, and lastly Naive Bayes. The learning curves confirm that ensemble-based models (Random Forest and XGBoost) not only achieve higher accuracy but also scale better with increasing data, making them the most effective and stable models in this comparison [5].

4.2 Discussion

Overall, Random Forest and XGBoost deliver the most reliable results for hypertension screening—consistently topping F1-score and accuracy—while SVM (RBF) performs competitively when numeric features are properly scaled, and Naive Bayes remains a fast but weaker baseline. Misclassifications skew toward false negatives in borderline profiles (e.g., near-normal BMI and moderate sleep), highlighting the value of threshold adjustment and prioritizing recall to reduce missed cases. Feature analysis identifies BP_History and Age as dominant predictors, with BMI and Sleep_Duration shaping risk in combination—patterns that match clinical expectations and support trust in the

models. In practice, RF/XGBoost are strong choices for initial screening (favoring sensitivity and balance), SVM is suitable for follow-up triage when precision matters, and NB fits resource-limited use or stacking. Limitations include restricting evaluation to four metrics (omitting probability calibration and cost-sensitive analysis) and the lack of external validation; future work should add cost-aware thresholding, probability calibration (Platt/Isotonic), Decision Curve Analysis, and external validation across settings and time to confirm generalizability [6].

5 CONCLUSION

Based on the overall performance comparison, the XGBoost model clearly outperforms all other models, achieving the highest accuracy, precision, recall, and F1-score across evaluations, as well as the most stable learning curve with near-perfect generalization. The Random Forest model follows closely, showing strong predictive ability and consistent improvement with larger training data, indicating robustness and low overfitting. The SVM model delivers solid and balanced results but performs slightly below the ensemble methods, suggesting it captures patterns well but may be limited in handling complex data relationships. Meanwhile, the Naive Bayes model, while fast and simple, exhibits the lowest accuracy and little improvement as the training set grows, reflecting its assumptions of feature independence and limited flexibility. Overall, ensemble-based methods—particularly XGBoost and Random Forest—prove to be the most effective and reliable models for this classification task, demonstrating superior accuracy, scalability, and generalization compared to SVM and Naive Bayes.

6 LIMITATIONS AND FUTURE WORK

Despite the strong performance achieved by ensemble-based models, this study has several limitations. First, the evaluation relied primarily on accuracy, precision, recall, and F1-score, without incorporating probability calibration or cost-sensitive analysis, which are important in clinical decision-making. Second, the dataset was derived from a single source and lacks external validation, potentially limiting generalizability across populations. Future work should incorporate external datasets, apply calibration techniques such as Platt scaling or isotonic regression, and explore decision-curve analysis to better assess clinical utility [6].

REFERENCES

[1] E. J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," Nature Medicine, vol. 25, no. 1, pp. 44-56, 2019.

[2] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: A survey of recent advances in deep learning techniques for electronic health record analysis," IEEE J. Biomed. Health Inform., vol. 22, no. 5, pp. 1589-1604, 2018.

[3] E. Christodoulou et al., "A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models," J. Clin. Epidemiol., vol. 110, pp. 12-22, 2019.

[4] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp. 785-794, 2016.

[5] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[6] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," Proc. IJCAI, pp. 1137-1143, 1995.

[7] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, Art. no. 6, 2020.

[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2011.

[9] S. Garcia, S. Ramirez-Gallego, J. Luengo, J. M. Benitez, and F. Herrera, "Big data preprocessing: methods and prospects," Big Data Analytics, vol. 1, Art. no. 9, 2016.

[10] N. Chamidah, E. Z. Astuti, and S. Slamin, "Comparison of Min-Max and Z-Score normalization for breast cancer classification," Jurnal RESTI, vol. 6, no. 1, pp. 10-15, 2022.

[11] P. W. Handayani et al., "Health information systems research in Indonesia: A systematic review," Heliyon, vol. 6, no. 8, Art. no. e04588, 2020.

[12] O. D. Nurhayati et al., "Penerapan machine learning untuk klasifikasi penyakit," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 8, no. 3, pp. 501-510, 2021.

[13] A. Wibowo and D. Riana, "Analisis performa algoritma klasifikasi pada data medis," Jurnal Sistem Informasi, vol. 16, no. 2, pp. 93-104, 2020.

[14] Kementerian Kesehatan Republik Indonesia, Profil Kesehatan Indonesia 2022. Jakarta, Indonesia: Kemenkes RI, 2022.

[15] Suyanto, Machine Learning Tingkat Dasar dan Lanjut. Bandung, Indonesia: Informatika, 2018.