Indonesian Journal of Electrical Engineering and Computer Science
Vol. 41, No. 1, January 2026, pp. 375~386
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i1.pp375-386 ❒ 375

Development of a machine learning model with optuna and ensemble learning to improve performance on multiple datasets

Akmar Efendi[1], Iskandar Fitri[2], Gunadi Widi Nurcahyo[2]
[1]Department of Informatics Engineering, Faculty of Engineering, Islamic University of Riau, Pekanbaru, Indonesia
[2]Department of Technology Information, Faculty of Computer Science, University of Putra Indonesia YPTK, Padang, Indonesia

Article Info

Article history:

Received Jun 5, 2024
Revised Sep 23, 2025
Accepted Dec 13, 2025

Keywords:

Ensemble
Hyperparameter tuning
Improve
Machine learning
XGBoost

ABSTRACT

Machine learning, a subset of artificial intelligence (AI) is vital for its ability to learn from data and improve system performance. In Indonesia, advancements in ML have significant potential to boost competitiveness and foster sustainable development. However, issues like overfitting and suboptimal parameter settings can hinder model effectiveness. This study aims to improve the classification performance of ML models on various datasets. Advanced techniques like hyperparameter tuning with Optuna and ensemble learning with extreme gradient boosting (XGBoost) are integrated to enhance model performance. The study evaluates the performance of K-nearest neighbors (KNN), support vector machine (SVM), and Gaussian naïve Bayes (GNB) algorithms across three datasets: academic records from the Islamic University of Riau (UIR), diabetes data from Kaggle, and Twitter data related to the 2024 elections. The findings reveal that the GNB algorithm outperforms KNN and SVM across all datasets, achieving the highest accuracy, precision, recall, and F1-score. Hyperparameter tuning with Optuna significantly improves model performance, demonstrating the value of systematic optimization. This study highlights the importance of advanced optimization techniques in developing high-performing ML models. The results suggest that robust algorithms like GNB, combined with hyperparameter tuning and ensemble learning, can significantly enhance classification performance.

Corresponding Author:

Akmar Efendi
Department of Informatics Engineering, Faculty of Engineering, Islamic University of Riau
Pekanbaru, Indonesia
Email: akmarefendi@eng.uir.ac.id

1. INTRODUCTION

Machine learning (ML), a subset of artificial intelligence (AI), enables machines to learn from historical data or past experiences without requiring manual programming for specific tasks [1]. Its ability to facilitate automatic learning allows systems to enhance their accuracy over time [2]. In simpler terms, ML methods mimic human learning processes. Amidst the rapidly evolving digital revolution, AI and ML technologies play pivotal roles in driving innovation and efficiency across various sectors globally. In Indonesia, advancements in AI and ML hold significant potential to boost competitiveness and open new avenues for sustainable development [3].

ML is categorized into two types: supervised learning and unsupervised learning [4]. his research adopts a supervised learning approach, where ML models are trained to identify patterns between input data and output labels. The method involves receiving information as input and generating labeled data as output. The machine analyzes the relationships and dependencies between data, compares actual output with expected output, and implements changes if discrepancies are found, thereby ensuring operational accuracy [5]. Supervised learning employs several algorithms to recognize patterns and detect relationships between input data and output labels [6]. One such algorithm is K-nearest neighbors (KNN), which assumes that similar data points are likely to be found nearby [7]. This algorithm measures the distance between data points, typically using Euclidean distance, and categorizes them based on the most frequent type observed [8].

Numerous studies have employed ML to address various issues in education. For instance, research [9] used ML to predict students' timely graduation. Additionally, ML has been used to forecast new student enrolments [10], assess the impact of online learning activities during the COVID-19 pandemic [11], and determine university majors based on job market trends and experiences [12]. This study also applies ML to students at the Islamic University of Riau (UIR), specifically within the Faculty of Engineering. Currently, the faculty and study programs only determine whether students graduate. However, there is a need to identify whether students graduate on time, late, or withdraw. Manual checks are time-consuming, necessitating the use of ML to address this issue. This research aims to classify engineering students to identify those who graduate on time, late, drop out, or withdraw, providing insights for developing new strategies to improve on-time graduation rates at UIR. Several ML algorithms will be compared, and their accuracy will be enhanced using ensemble methods, hyperparameter tuning, and data balancing techniques.

Research by Anam *et al.* [13] used the support vector machine (SVM) algorithm for online course classification, demonstrating that balanced data improves accuracy, with SVM alone achieving 72% accuracy and SVM with synthetic minority oversampling technique (SMOTE) achieving 95%. Another study by Rani and Gill [14] employed a hybrid model for classification, showing that ensemble or hybrid methods can enhance accuracy, with ensemble techniques achieving 86% accuracy and a proposed hybrid model reaching 91%. Further research by Ramdani *et al.* [15] compared multinomial Naïve Bayes (MNB) and MNB + AdaBoost, finding that AdaBoost improved accuracy by 5%. Another study by Zamsuri *et al.* [16] tested the KNN Algorithm, achieving 79% accuracy. Research by Hadianti [17] improved KNN performance with hyperparameter optimization using optuna, reaching 80.82% accuracy. Studies by Haque *et al.* [18] and Kang *et al.* [19] achieved accuracies of 85.40% and 63.81% with SVM and SVM with Random Search, respectively. Additional studies by Rahmaddeni *et al.* [20] and Vedaraj *et al.* [21] improved SVM performance with extreme gradient boosting (XGBoost), reaching 79% accuracy, and achieved 96% accuracy with the Gaussian Naive Bayes (GNB) algorithm. Finally, Ashari and Untoro [22] optimized GNB with Hyperparameter Optimization using a Genetic Algorithm, achieving 93.2% accuracy, and [23] combined GNB and XGBoost to reach 81.55% accuracy.

These studies indicate that basic algorithms typically achieve low accuracy, highlighting the need for additional methods to improve accuracy. This research will also compare the accuracy of basic ML algorithms with the developed models, using techniques like SMOTE for data balancing [24]–[26]. Besides SMOTE, this study employs XGBoost to enhance accuracy using ensemble methods with boosting techniques [27]. Hyperparameter tuning with Optuna will also be used. Optuna is a library designed to automate parameter tuning processes. It offers efficient search algorithms, enhancing the effectiveness of the search process, and is straightforward to use, making it a valuable tool for improving parameter tuning efficiency and accuracy [28]. The study presents a model that integrates Optuna and XGBoost, referred to as Hyte, to enhance ML performance. This model is tested on three diverse datasets: academic data from the Islamic UIR, diabetes data from Kaggle, and tweets related to the 2024 elections from Twitter (X). The objective is to assess the model's robustness and adaptability across different types of data. By applying Hyte to these varied datasets, the research aims to reveal the strengths and weaknesses of each algorithm in distinct contexts. This comprehensive evaluation underscores the model's versatility and potential for generalization, demonstrating its efficacy in handling diverse data sources. The findings provide valuable insights into improving ML methodologies, emphasizing the importance of testing models on multiple datasets to ensure their effectiveness and reliability in real-world applications.

2. METHOD

2.1. Research framework

This section details the motivation, research framework, research stages, research materials, initial data processing, tools, methods, experiments, model/method testing, and the evaluation and validation of the model/method. The research framework depicted in Figure 1 outlines a structured approach beginning with problem identification, where the specific issues to be addressed are thoroughly examined. This initial step ensures a clear understanding of the research context and the challenges to be tackled. Following this, the research objectives are meticulously defined to provide a focused direction for the study. Subsequently, a comprehensive literature review of previous studies is conducted to ascertain the extent of existing research related to the topic. This review helps identify gaps in the current knowledge and informs the research design. Based on insights gained from the literature, data is collected from the academic records of the 2016 cohort of students from the Faculty of Engineering at the Islamic UIR. The collected data undergoes rigorous preprocessing to ensure it is clean, consistent, and ready for analysis. The study employs several ML algorithms, specifically KNN, SVM, and GNB. The performance of these algorithms is compared to evaluate the effectiveness of the base algorithms against their improved versions. The algorithm demonstrating the highest classification accuracy is selected for further analysis.

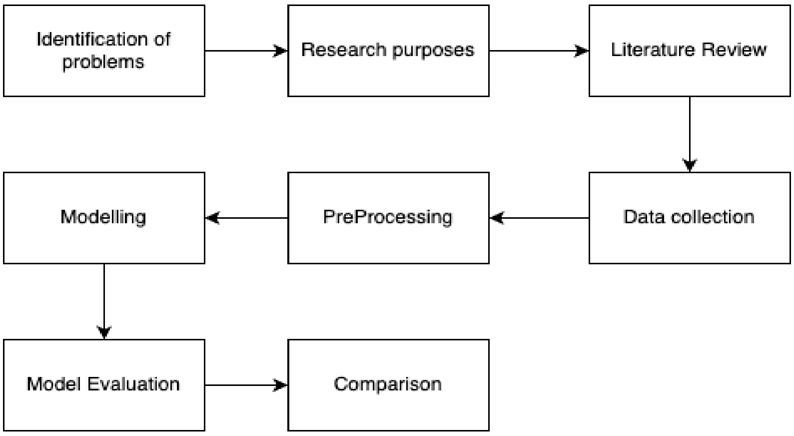

Figure 1. Research framework

2.2. Model development

The model used in this research integrates various techniques for improving ML performance. The development process, as shown in Figure 2, includes stages such as data collection, preprocessing, model selection, training, and evaluation. Each stage is meticulously designed to ensure the robustness and reliability of the model.

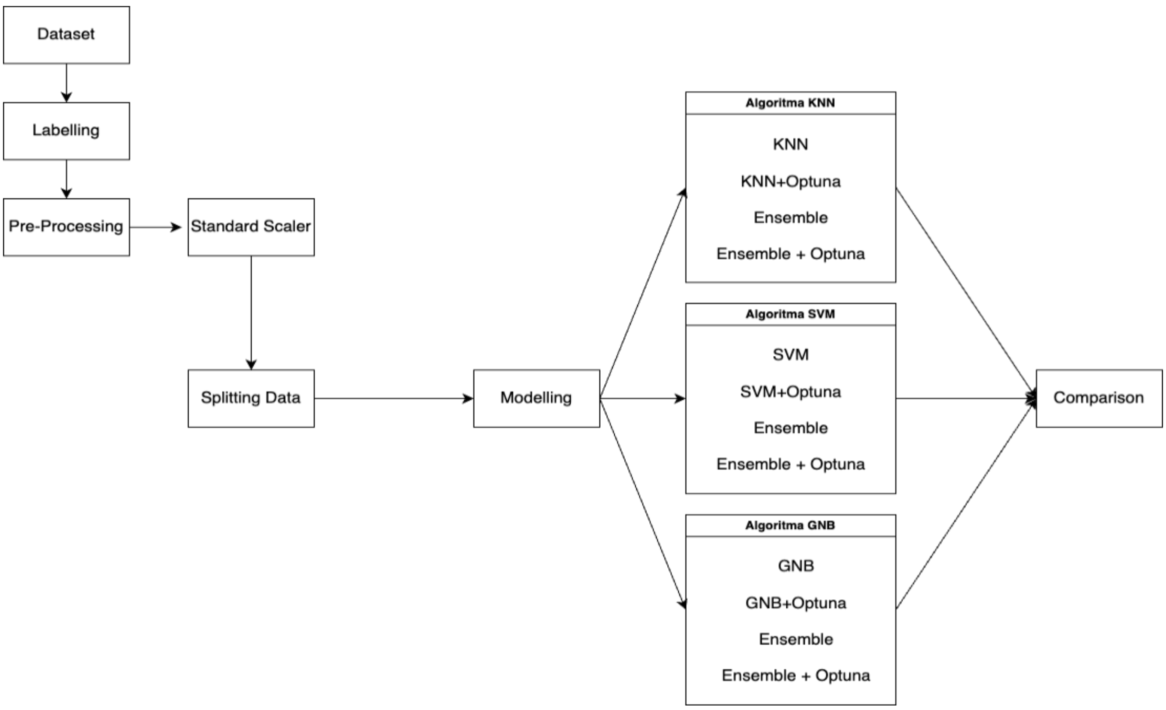

Figure 2. Model development

2.3. Dataset

The dataset in this study consists of three distinct datasets:

- Academic data: the first dataset comprises academic records from the 2016 cohort of the Faculty of Engineering at the Islamic UIR, initially containing 300,205 records. This dataset was filtered to include only students who completed the courses Thesis and Thesis 2, resulting in 1,282 records.
- Diabetes data: the second dataset, sourced from Kaggle, contains diabetes data with a total of 767 records.
- Twitter data: the third dataset, sourced from Twitter, includes 4,000 records related to various tweets using the hashtag #Pemilu2024.

2.4. Data label

The labeling of data in dataset 1 includes four categories: dropout (DO), graduated on time, graduated late, and withdrawn. These categories are determined based on the thesis grades from the academic records of the Faculty of Engineering at the Islamic UIR. The labeling process follows these criteria:

- A student is labelled as DO if they exceed the specified time limit of 14 semesters or if they have committed serious violations.
- A student is labelled as graduated on time if they graduate in the 7th, 8th, or 9th semester.
- A student is labelled as graduated late if they graduate after the 9th semester.
- A student is labelled as withdrawn if they do not take the thesis course or if their KRS indicates they have withdrawn in a particular semester.

For dataset 2, which uses diabetes data from Kaggle, the labels assigned are potential and not potential. For dataset 3, using data from Twitter with the hashtag #Pemilu2024, the labels assigned are positive, negative, and neutral.

2.5. Preprocessing

The labeled data is then standardized using a standard scaler. The standard scaler is employed to ensure that the data has a consistent scale and range. Additionally, the standard scaler has the advantage of providing a more stable influence of outliers compared to other normalization methods. While outliers can significantly impact other normalization methods, the standard scaler relies on the mean and standard deviation, which are less affected by extreme values. After cleaning all the data, standardization will be performed to improve the performance of the algorithm by using 'StandardScaler'. StandardScaler is used to standardize the data by adjusting the distribution of values in each feature to have a mean of 0 and a standard deviation of 1. Figure 3 is the coding to standardize the data.

```
# standarisasi data

from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()
scaler.fit(IPK.drop('Label',axis=1))
```
```
StandardScaler()
```

Figure 3. Coding for data standardization

2.6. Splitting data

This stage is crucial for several reasons that contribute to the development of a robust model and objective evaluation. Firstly, during the splitting phase, the data is divided into two sets: the training set and the testing set. The training set is used to train the model, while the testing set is used to test the trained model. This ensures that the model can generalize well to unseen data. Additionally, using a separate testing set allows for an objective evaluation of the model's performance, providing insight into how well the model performs on previously unseen data. This is particularly relevant for assessing the model's functionality in a production environment or with new data. Once data standardization is complete, the data can be processed by splitting it into training and testing sets. In this process, the data is divided into a 70:30 ratio, with 70% used for training and 30% for testing. Figure 4 shows the coding for splitting data.

The function `train_test_split(x, y, test_size=0.3, random_state=rs)` is used to divide the dataset into training and testing subsets. The variable `x` represents the independent variables (features), and `y` represents the dependent variable (target). The parameter `test_size=0.3` specifies that 30% of the data will be allocated for testing, while `random_state=rs` ensures consistent data splitting, so the results remain the same each time the function is executed. The variables `x_train`, `x_test`, `y_train`, and `y_test` store the results of the dataset split. Specifically, `x_train` and `y_train` are the training data for features and targets, respectively, while `x_test` and `y_test` are the testing data.

```
from sklearn.model_selection import train_test_split
x=IPK_skripsi
y=IPK['Label']

rs=42

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=rs)
```

Figure 4. Data splitting coding

2.7. Machine learning algorithm modeling

After completing the aforementioned processes, the next step is to perform modeling using ML algorithms. The algorithms used in this research are KNN, SVM, and Naïve Bayes. Additionally, this study employs parameter optimization using the Optuna method. To further improve accuracy, this research also incorporates ensemble learning with the XGBoost algorithm. A total of 12 models are tested in this study, as shown in Table 1.

Table 1. Model testing

Algorithm	Dataset	Testing
KNN	Academic	KNN Based
		KNN + Optuna
		KNN + XGBoost (ensemble)
		KNN Hyte
	Diabetes	KNN Based
		KNN + Optuna
		KNN + XGBoost (ensemble)
		KNN Hyte
	Twitter (X)	KNN Based
		KNN + Optuna
		KNN + XGBoost (ensemble)
		KNN Hyte
SVM	Academic	SVM Based
		SVM + Optuna
		SVM + XGBoost (ensemble)
		SVM Hyte
	Diabetes	SVM Based
		SVM + Optuna
		SVM + XGBoost (ensemble)
		SVM Hyte
	Twitter (X)	SVM Based
		SVM + Optuna
		SVM + XGBoost (ensemble)
		SVM Hyte
GNB	Academic	GNB Based
		GNB + Optuna
		GNB + XGBoost (ensemble)
		GNB Hyte
	Diabetes	GNB Based
		GNB + Optuna
		GNB + XGBoost (ensemble)
		GNB Hyte
	Twitter (X)	GNB Based
		GNB + Optuna
		GNB + XGBoost (ensemble)
		GNB Hyte

2.8. Experimentation and evaluation

The experiments involve applying the models to the datasets and evaluating their performance. The accuracy, precision, recall, and F1-score are calculated for each model to determine their effectiveness. The model demonstrating the highest classification accuracy is selected for further analysis.

2.9. Model improvement

To improve model performance, hyperparameter tuning is performed using Optuna. Optuna automates the parameter tuning process, offering an efficient search algorithm that improves search effectiveness. Table 2 is the rationale for the selection of algorithms used in the study.

Table 2. Reasons for selecting the algorithm that will be developed

No	Researcher	Testing	Accuracy (%)
1	[29]	KNN Algorithm	85
2	[17]	KNN + Hyperparameter (Optuna)	80.82
3	-	KNN + XGBoost	-
4	-	KNN + XGBoost + Hyperparameter (Hyte)	-
5	[18]	SVM Algorithm	85.40
6	[19]	SVM + Hyperparameter (Optuna)	63.81
7	[20]	SVM + XGBoost	79
8	-	SVM + XGBoost + Hyperparameter (Hyte)	-
9	[21]	GNB Algorithm	96
10	[22]	GNB + Hyperparameter (Optuna)	93.2
11	[23]	GNB + XGBoost	81.55
12	-	GNB + XGBoost + Hyperparameter (Hyte)	-

2.10. Model improvement

This comprehensive method section ensures that the study is valid and reproducible. By providing detailed steps and code snippets, a knowledgeable reader can repeat the experiment and validate the findings. References to previously published procedures are highlighted to provide additional context and support.

3. RESULTS AND DISCUSSION

3.1. Result

Several libraries will be utilized in this research. The panda's library will be imported for data analysis, providing data structures such as DataFrame. The numpy library will be employed for array operations, and Seaborn will be used for data visualization. The K-neighbors classifier class, from sklearn.neighbors, will be implemented for classification tasks based on the KNN algorithm.

Following the implementation of the necessary libraries, the next step involves labeling the academic data from the Faculty of Engineering at the Islamic UIR and the diabetes data sourced from Kaggle. Given that the data is numeric, this research will involve removing irrelevant variables. Once the data is cleansed, standardization will be performed using 'StandardScaler' to enhance the algorithm's performance. The StandardScaler will be used to standardize the data by adjusting the distribution of values in each feature to have a mean of 0 and a standard deviation of 1.

Additionally, this research incorporates data from Twitter, which requires a slightly different preprocessing approach. The preprocessing steps include data cleaning, case folding, text normalization, tokenizing, filtering, and stemming. The Twitter data will then be weighted using term frequency-inverse document frequency (TF-IDF). After data cleaning, the next step is to test the designed model using a 70:30 data split. In the first experiment, academic data will be used. Table 3 presents the classification report for modeling using the base KNN model, KNN with Optuna, KNN with XGBoost, and KNN hybrid models.

Table 3. Accuracy results with academic data

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	92.90	93	93	93
KNN + XGBoost	93.40	93	93	93
KNN + Optuna	92.26	92	92	92
KNN Hyte	90.38	85	90	87

Using the KNN algorithm alone, an accuracy of 92.90% was achieved, with precision, recall, and F1 scores each at 93%. When KNN was combined with XGBoost, the precision, recall, and F1 scores remained at 93%, but the accuracy slightly increased to 93.40%. Employing KNN with Optuna resulted in an accuracy of 92.26%, with precision, recall, and F1 scores each at 92%. In contrast, the KNN-Hyte variation achieved an accuracy of 90.38%, with precision at 85%, recall at 90%, and an F1 score of 87%. Overall, the combination of ANN and XGBoost yielded the highest accuracy, with each model demonstrating a different balance between precision, recall, and accuracy.

Table 3 also reveals that the use of KNN-Hyte did not improve accuracy. In this experiment, KNN-Hyte showed a decline compared to the KNN-based model. Additionally, this study utilized another dataset, specifically the diabetes data from Kaggle. Table 4 presents the results of applying the KNN algorithm to the diabetes data.

Table 4. Accuracy results with diabetes data

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	80	81	80	80
KNN + XGBoost	81	81	81	81
KNN + Optuna	80	82	80	80
KNN Hyte	79	80	79	79

Table 4 presents the performance results of several algorithms used to analyze diabetes data. The KNN algorithm alone achieved an accuracy of 80%, with a precision of 81%, recall of 80%, and F1-Score of 80%. When KNN was combined with XGBoost, the accuracy increased to 81%, with precision, recall, and F1-Score all at 81%. Using KNN with Optuna resulted in an accuracy of 80%, with a slightly higher precision of 82%, while recall and F1-Score remained at 80%. The KNN-Hyte variation showed an accuracy of 79%, with a precision of 80%, recall of 79%, and F1-Score of 79%. Overall, the combination of KNN with XGBoost provided the best results in terms of accuracy and balance between precision, recall, and F1-Score.

With the diabetes data, the accuracy decreased compared to the academic data. Subsequently, the model was tested using the Twitter dataset. Table 5 presents the accuracy results obtained using the Twitter dataset.

Table 5. Accuracy results with twitter data

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	91	91	91	91
KNN + XGBoost	90	91	90	90
KNN + Optuna	66	73	67	66
KNN Hyte	89	90	89	89

Using the KNN algorithm with the Twitter dataset, the base algorithm achieved an accuracy of 91%, with precision, recall, and F1-Score all at 91%. This demonstrates that KNN can consistently classify data with a low error rate. When KNN was combined with XGBoost, the accuracy slightly decreased to 90%, although precision remained at 91%, while recall and F1-Score each slightly dropped to 90%. This indicates that despite a slight decrease in recall and F1-Score, precision remained high, indicating the model's effectiveness at producing correct results.

Using KNN with Optuna showed a significant decline in performance, with accuracy only reaching 66%, precision at 73%, recall at 67%, and F1-Score at 66%. These results suggest that this combination is less effective for Twitter data, leading to numerous classification errors. The KNN-Hyte variation achieved an accuracy of 89%, with precision at 90%, recall at 89%, and F1-Score at 89%. Although slightly lower than pure KNN, these results still demonstrate good and consistent performance in data classification.

Overall, the KNN algorithm alone provided the best results in terms of accuracy and consistency across other metrics for Twitter data. The combination with XGBoost also delivered nearly comparable performance, but Optuna was ineffective in this context. KNN-Hyte still provided good performance, albeit slightly below pure KNN. After processing with the KNN algorithm, the next step is to test with other algorithms, namely SVM and GNB as shown in Table 6.

From all the tests carried out, the results obtained show that not all algorithms can be improved with XGBoost, Optuna, or both (Hyte). The highest accuracy using Hyte was obtained in testing with the SVM algorithm with the Twitter dataset, namely 97%. The results show better performance compared to previous research as seen in Table 7.

In Table 7, it is evident that the study significantly improves when the SVM algorithm is tested using the Twitter dataset. However, the results are less satisfactory with the KNN and GNB algorithms, as their performance decreases compared to the base algorithm. Conversely, in the academic and diabetes datasets, the GNB with Hyte algorithm successfully enhances the model's performance. For SVM and KNN, their performance decreases when compared to the base algorithm. This indicates that for nominal data, GNB Hyte is highly recommended as it can improve model performance. However, for text datasets like Twitter, SVM Hyte is highly recommended, as it achieved the highest accuracy in the study.

Table 6. Accuracy results with SVM and GNB algorithms

No	Algorithm	Dataset	Model	Accuracy (%)
1	SVM	Academic	SVM	92,26
2			SVM + XGBoost	93,40
3			SVM + Optuna	91,63
4			SVM Hyte	90,11
5		Twitter	SVM	94
6			SVM + XGBoost	93
7			SVM + Optuna	96
8			SVM Hyte	97
9		Diabetes	SVM	78
10			SVM + XGBoost	73
11			SVM + Optuna	80
12			SVM Hyte	77
13	GNB	Academic	GNB	85,65
14			GNB + XGBoost	91
15			GNB + Optuna	87,45
16			GNB Hyte	92,26
17		Twitter	GNB	68
18			GNB + XGBoost	81
19			GNB + Optuna	50
20			GNB Hyte	54
21		Diabetes	GNB	75
22			GNB + XGBoost	73
23			GNB + Optuna	73
24			GNB Hyte	80

Table 7. Comparison of accuracy with previous research on SVM algorithms

Researcher	Dataset	Model	Accuracy (%)
[20]	Twitter	SVM + XGBoost	79
[30]		SVM + Adaboost	92
[31]		SVM + Particle Swarm Optimization (PSO)	91,63
[32]		SVM + Query Expansion Ranking	82
This Research		SVM Hyte	97

3.2. Discussion

This study explores the integration of ensemble methods and hyperparameter tuning to improve ML performance across diverse datasets, addressing key gaps in the existing literature. Previous studies have extensively examined the benefits of ensemble methods like XGBoost and hyperparameter tuning using techniques such as Optuna; however, their impact on datasets from multiple domains, particularly in academic, health, and social media contexts, remains underexplored. By applying these techniques to datasets with distinct characteristics-academic records, diabetes data, and Twitter sentiment data-this study contributes novel insights into the adaptability and robustness of ML models across different types of data.

Our results demonstrate that combining KNN with XGBoost consistently enhances performance, as evidenced by the improved classification accuracy observed across datasets. For instance, the accuracy achieved with XGBoost-enhanced KNN was 93.40% for academic data and 81% for diabetes data, reflecting a notable improvement over the base KNN algorithm. These results align with the findings of [17], which highlighted the effectiveness of ensemble methods in boosting classification performance. Notably, we found that XGBoost contributed to a more balanced performance across precision, recall, and F1-score, indicating its capacity to improve not only accuracy but also the consistency of model predictions.

Interestingly, despite the success of XGBoost, the application of hyperparameter tuning using Optuna did not yield the same level of improvement, particularly with the Twitter dataset, where accuracy dropped significantly to 66%. This finding contrasts with prior research [18], where hyperparameter optimization generally enhanced performance across various datasets. The discrepancy suggests that while Optuna may offer substantial gains for structured data like academic or medical datasets, it might be less effective for unstructured data such as social media posts, where noise and variability are more pronounced. The decline in performance for the KNN-Hyte model, specifically, underscores the importance of context-specific tuning, as not all algorithms or optimization techniques translate effectively across domains.

Moreover, our findings reveal the robustness of ensemble methods like XGBoost when applied to diverse datasets. The improvements observed in classification accuracy for academic, diabetes, and Twitter datasets suggest that XGBoost is highly adaptable and resilient, even when dealing with varying data structures and characteristics. This supports the argument that ensemble techniques, when properly implemented, can provide substantial performance gains across multiple domains. Our study contributes to the growing body of evidence that ensemble learning is an effective strategy for handling imbalanced and complex datasets.

However, there are limitations to this study that warrant consideration. First, while the datasets used in this research represent different domains, they remain limited in size and scope. Larger datasets or those from other fields, such as finance or e-commerce, could provide further validation of these findings. Additionally, the computational cost associated with hyperparameter tuning and ensemble methods is non-trivial, posing potential challenges for real-time applications where rapid processing is critical. Future research should explore more efficient optimization techniques or hybrid approaches that balance accuracy with computational efficiency.

The implications of this research extend beyond the datasets and algorithms explored. Our findings suggest that the versatility of XGBoost makes it a strong candidate for broader applications, particularly in areas where data is both structured and unstructured. The performance of XGBoost across different contexts highlights its potential for widespread adoption in ML tasks requiring reliable, scalable solutions. Moreover, the limited success of Optuna in optimizing certain datasets invites further investigation into the development of more domain-specific optimization techniques, particularly for unstructured data.

In conclusion, this study provides conclusive evidence that ensemble methods, particularly XGBoost, offer significant improvements in ML performance across diverse datasets. The findings emphasize the value of testing ML models across multiple domains to ensure their robustness and generalizability. While hyperparameter tuning remains a valuable tool for enhancing performance, its effectiveness may be contingent on the specific characteristics of the dataset in question. Future research should continue to investigate the intersection of ensemble learning and optimization techniques, with a focus on improving scalability and efficiency for real-world applications.

4. CONCLUSION

This research successfully demonstrates the effectiveness of integrating advanced techniques like hyperparameter tuning with Optuna and ensemble learning with XGBoost to enhance the performance of ML models. The primary objective was to improve the accuracy of ML models for classification tasks, and the findings confirm that advanced techniques can significantly enhance model performance. The GNB algorithm consistently outperformed the KNN and SVM algorithms across different datasets, achieving the highest accuracy, precision, recall, and F1-score. This indicates that GNB is particularly well-suited for classification tasks in diverse domains, suggesting its potential for broader applications in real-world scenarios. One of the key takeaways from this study is the importance of hyperparameter tuning and ensemble learning in improving ML model performance. Optuna's hyperparameter tuning significantly improved model accuracy, demonstrating the value of systematic optimization in ML. However, this study also highlights some limitations and raises several questions for future research. The scope of the datasets used in this study was limited to academic, diabetes, and Twitter data. Future research should explore the application of these techniques to a wider range of datasets to validate the robustness and generalizability of the findings. Additionally, while this study focused on KNN, SVM, and GNB, other ML algorithms could be evaluated using similar optimization techniques to determine their effectiveness.

The findings of this study have significant implications for the research field and the broader community. For researchers, this study underscores the importance of advanced optimization techniques in developing high-performing ML models. For practitioners, particularly in fields like education, healthcare, and social media analysis, the use of robust algorithms like GNB can lead to more accurate predictions and better decision-making. In conclusion, this study provides a foundation for further exploration into the optimization of ML algorithms. The successful application of hyperparameter tuning and ensemble learning techniques demonstrates their potential to enhance model performance across various domains. Future research should continue to explore these and other advanced techniques to push the boundaries of what ML models can achieve, ultimately benefiting both the research community and practical applications in various fields.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support provided by Riau Islamic University and Putra Indonesia University YPTK Padang for facilitating this research through the provision of institutional support, research facilities, and funding. The authors also appreciate the academic environment and administrative assistance that contributed to the successful completion of this study.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

The authors contributed to this work according to the CRediT taxonomy as follows. Akmar Efendi contributed to conceptualization, methodology, software development, formal analysis, investigation, data curation, visualization, and writing the original draft. Iskandar Fitri contributed to conceptualization, validation, formal analysis, resources, writing review and editing, supervision, and project administration. Gunadi Widi Nurcahyo contributed to conceptualization, methodology, validation, investigation, writing review and editing, supervision, project administration, and funding acquisition. All authors reviewed and approved the final manuscript.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Akmar Efendi	✓	✓	✓	✓	✓	✓		✓	✓		✓			✓
Iskandar Fitri	✓			✓	✓		✓			✓		✓	✓	
Gunadi Widi Nurcahyo	✓	✓		✓		✓				✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

Ethical approval was not required for this study because it did not involve interventions or experiments with human participants or animals. The study used secondary data sources. Institutional permission to use the academic dataset was obtained from Riau Islamic University (UIR). Additional datasets were collected from publicly available repositories (Kaggle) and publicly accessible content from the X platform. All data were handled in accordance with applicable institutional policies and relevant national regulations. Where applicable, identifiers were removed and the data were analyzed and reported only in aggregate form to minimize privacy risks.

DATA AVAILABILITY

The data that support the findings of this study were obtained from multiple sources. An academic dataset was provided by Riau Islamic University (UIR) and was used with institutional permission; restrictions apply to the availability of these data, and they are not publicly available. Additional datasets were obtained from publicly available sources, including Kaggle repositories and publicly accessible content from the X platform, in accordance with their respective terms of use. Processed and derived data supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, pp. 54–70, 2023, doi: 10.1016/j.cogr.2023.04.001.

[2] M. M. Taye, "Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions," *Computers*, vol. 12, no. 5, 2023, doi: 10.3390/computers12050091.

[3] Y. K. Dwivedi *et al.*, "'So what if ChatGPT wrote it?' Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy," *International Journal of Information Management*, vol. 71, 2023, doi: 10.1016/j.ijinfomgt.2023.102642.
[4] S. Badillo *et al.*, "An introduction to machine learning," *Clinical Pharmacology and Therapeutics*, vol. 107, no. 4, pp. 871–885, 2020, doi: 10.1002/cpt.1796.
[5] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, 2021, doi: 10.1007/s42979-021-00592-x.
[6] L. K. Vora, A. D. Gholap, K. Jetha, R. R. S. Thakur, H. K. Solanki, and V. P. Chavda, "Artificial intelligence in pharmaceutical technology and drug delivery design," *Pharmaceutics*, vol. 15, no. 7, 2023, doi: 10.3390/pharmaceutics15071916.
[7] Z. Chen, L. J. Zhou, X. Da Li, J. N. Zhang, and W. J. Huo, "The Lao text classification method based on KNN," *Procedia Computer Science*, vol. 166, pp. 523–528, 2020, doi: 10.1016/j.procs.2020.02.053.
[8] P. H. Putra, M. S. Novelan, and M. Rizki, "Analysis K-nearest neighbor method in classification of vegetable quality based on color," *Journal of Applied Engineering and Technological Science*, vol. 3, no. 2, pp. 126–132, 2022, doi: 10.37385/jaets.v3i2.763.
[9] R. Bakri, N. P. Astuti, and A. S. Ahmar, "Machine learning algorithms with parameter tuning to predict students' graduation-on-time: A case study in higher education," *Journal of Applied Science, Engineering, Technology, and Education*, vol. 4, no. 2, pp. 259–263, 2022, doi: 10.35877/454RI.asci1581.
[10] C. Kaensar and W. Wongnin, "Predicting new student performances and identifying important attributes of admission data using machine learning techniques with hyperparameter tuning," *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 19, no. 12, 2023, doi: 10.29333/ejmste/13863.
[11] A. A. Shaikh, A. Kumar, K. Jani, S. Mitra, D. A. García-Tadeo, and A. Devarajan, "The role of machine learning and artificial intelligence for making a digital classroom and its sustainable impact on education during Covid-19," *Materials Today: Proceedings*, vol. 56, pp. 3211–3215, 2022, doi: 10.1016/j.matpr.2021.09.368.
[12] A. O. Alsayed *et al.*, "Selection of the right undergraduate major by students using supervised learning techniques," *Applied Sciences (Switzerland)*, vol. 11, no. 22, 2021, doi: 10.3390/app112210639.
[13] M. K. Anam, T. A. Fitri, A. Agustin, L. Lusiana, M. B. Firdaus, and A. T. Nurhuda, "Sentiment analysis for online learning using the lexicon-based method and the support vector machine algorithm," *ILKOM Jurnal Ilmiah*, vol. 15, no. 2, pp. 290–302, 2023, doi: 10.33096/ilkom.v15i2.1590.290-302.
[14] S. Rani and N. S. Gill, "Hybrid model for twitter data sentiment analysis based on ensemble of dictionary based classifier and stacked machine learning classifiers-SVM, KNN and C5.0," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 4, pp. 624–635, 2020, [Online]. Available: www.jatit.org.
[15] C. M. S. Ramdani, A. Rachman Nur, and R. Setiawan, "Comparison of the multinomial naive Bayes algorithm and decision tree with the application of AdaBoost in sentiment analysis reviews PeduliLindungi application," *International Journal of Information System & Technology*, vol. 6, no. 4, pp. 419–430, Aug. 2022.
[16] A. Zamsuri, S. Defit, and G. W. Nurcahyo, "Classification of multiple emotions in Indonesian text using the K-nearest neighbor method," *Journal of Applied Engineering and Technological Science*, vol. 4, no. 2, pp. 1012–1021, 2023, doi: 10.37385/jaets.v4i2.1964.
[17] S. Hadianti, "Optimization of the machine learning approach using optuna in heart disease prediction," *Journal Medical Informatics Technology*, pp. 59–64, 2023, doi: 10.37034/medinftech.v1i3.15.
[18] R. Haque, N. Islam, M. Tasneem, and A. K. Das, "Multi-class sentiment classification on Bengali social media comments using machine learning," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 21–35, 2023, doi: 10.1016/j.ijcce.2023.01.001.
[19] H. S. Kang, Y. S. Choi, J. S. Yu, S. W. Jin, J. M. Lee, and Y. J. Kim, "Hyperparameter tuning of OC-SVM for industrial gas turbine anomaly detection," *Energies*, vol. 15, no. 22, 2022, doi: 10.3390/en15228757.
[20] R. Rahmaddeni, M. K. Anam, Y. Irawan, S. Susanti, and M. Jamaris, "Comparison of support vector machine and XGBSVM in analyzing public opinion on Covid-19 vaccination," *ILKOM Jurnal Ilmiah*, vol. 14, no. 1, pp. 32–38, 2022, doi: 10.33096/ilkom.v14i1.1090.32-38.
[21] M. Vedaraj, C. S. Anita, A. Muralidhar, V. Lavanya, K. Balasaranya, and P. Jagadeesan, "Early prediction of lung cancer using Gaussian naive Bayes classification algorithm," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 6s, pp. 838–848, 2023, [Online]. Available: www.ijisae.org.
[22] I. F. Ashari and M. C. Untoro, "Hyperparameter tuning feature selection with genetic algorithm and Gaussian naïve Bayes for diabetes disease prediction," *Jurnal Telematika*, vol. 17, no. 1, pp. 17–23, Oct. 2022, doi: 10.61769/telematika.v17i1.488.
[23] T. R. Mahesh, V. Vinoth Kumar, V. Muthukumaran, H. K. Shashikala, B. Swapna, and S. Guluwadi, "Performance analysis of XGBoost ensemble methods for survivability with the classification of breast cancer," *Journal of Sensors*, vol. 2022, 2022, doi: 10.1155/2022/4649510.
[24] N. Matondang and N. Surantha, "Effects of oversampling SMOTE in the classification of hypertensive dataset," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 4, pp. 432–437, 2020, doi: 10.25046/AJ050451.
[25] N. G. Ramadhan, "Comparative analysis of ADASYN-SVM and SMOTE-SVM methods on the detection of type 2 diabetes mellitus," *Scientific Journal of Informatics*, vol. 8, no. 2, pp. 276–282, 2021, doi: 10.15294/sji.v8i2.32484.
[26] J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere, and S. Hussain, "Effective class-imbalance learning based on SMOTE and convolutional neural networks," *Applied Sciences (Switzerland)*, vol. 13, no. 6, 2023, doi: 10.3390/app13064006.
[27] S. S. Dhaliwal, A. Al Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information (Switzerland)*, vol. 9, no. 7, 2018, doi: 10.3390/info9070149.
[28] J. Jang, W. Jeong, S. Kim, B. Lee, M. Lee, and J. Moon, "RAID: Robust and interpretable daily peak load forecasting via multiple deep neural networks and shapley values," *Sustainability (Switzerland)*, vol. 15, no. 8, 2023, doi: 10.3390/su15086951.
[29] N. Hidayati and A. Hermawan, "K-nearest neighbor (K-NN) algorithm with Euclidean and Manhattan in classification of student graduation," *Journal of Engineering and Applied Technology*, vol. 2, no. 2, 2021, doi: 10.21831/jeatech.v2i2.42777.
[30] M. K. Anam, M. I. Mahendra, W. Agustin, R. Rahmaddeni, and N. Nurjayadi, "Framework for analyzing netizen opinions on BPJS using sentiment analysis and social network analysis (SNA)," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 6, no. 1, pp. 11–28, 2022, doi: 10.29407/intensif.v6i1.15870.
[31] A. D. N. W. Susanto and Hari Suparwito, "SVM-PSO algorithm for Tweet sentiment analysis #BesokSenin," *Indonesian Journal of Information Systems*, vol. 6, no. 1, pp. 36–47, 2023, doi: 10.24002/ijis.v6i1.7551.
[32] H. Atsqalani, N. Hayatin, and C. S. K. Aditya, "Sentiment analysis from Indonesian Twitter data using support vector machine and query expansion ranking," *Jurnal Online Informatika*, vol. 7, no. 1, pp. 116–122, 2022, doi: 10.15575/join.v7i1.669.

BIOGRAPHIES OF AUTHORS

Akmar Efendi He is a lecturer in informatics engineering, engineering faculty, Riau Islamic University. He completed his undergraduate education at STMIK YPTK Padang in 1988. Then he continued his master's education at the Information Technology study program, Putra Indonesia University YPTK Padang in 2007. Currently he is listed as a doctoral student in information technology at Putra Indonesia University YPTK Padang. His current research focus is in the field of data science. He can be contacted at email: akmarefendi@eng.uir.ac.id.

Iskandar Fitri holds the esteemed position of Professor at Universitas Putra Indonesia YPTK Padang, serving as a lecturer in the information technology doctorate program within the Faculty of Computer Science. His academic journey includes completing his undergraduate education (S1) in the electronics engineering study program at Universitas Nasional, followed by Master's (S2) and Doctoral (S3) degrees in the electronics engineering study program at Universitas Indonesia. Presently, he is recognized as a professor specializing in the microwave field at UPI YPTK Padang. Offering a diverse array of courses in the information technology study program, such as microwaves, research methodology, and artificial intelligence, for communication. He can be contacted at email: if@upiyptk.ac.id.

Gunadi Widi Nurcahyo He is a Senior Lecturer in Information Technology at Putra Indonesia University YPTK Padang. He completed his undergraduate education at STMIK YPTK Padang in 1988 majoring in Information Management. Then he continued his master's and doctoral education at the University of Technology Malaysia (UTM). His research field focuses on data science. He can be contacted at email: gunadiwidi@yahoo.co.id.