Machine Learning Algorithm Optimization using Stacking Technique for Graduation Prediction

Herianto[1,*], Bambang Kurniawan[2], Zupri Henra Hartomi[3,], Yuda Irawan[4,], M. Khairul Anam[5,]

[1, 2,3, 4]*Computer Science, Universitas Hang Tuah Pekanbaru, Pekanbaru, Indonesia*

[5]*Informatics, Universitas Samudra, Langsa, Indonesia*

(Received: May 30, 2024; Revised: June 21, 2024; Accepted: July 29, 2024; Available online: August 15, 2024)

Abstract

Graduating on time is crucial for academic success, impacting time, costs, and education quality. Hang Tuah University Pekanbaru (UHTP) is currently struggling to meet its goal of achieving a 75% on-time graduation rate. This study introduces an innovative approach using machine learning techniques, particularly ensemble learning with Stacking Machine Learning Optuna SMOTE (SMLOS), to address this issue. Our primary objective is to enhance data classification accuracy to predict student graduation timelines effectively. We employ algorithms such as K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (C4.5), Random Forest (RF), and Naive Bayes (NB). These were combined with meta-models, including Logistic Regression (LR), Adaboost, XGBoost, LR+Adaboost, and LR+XGBoost, to create a robust prediction model. To address class imbalance, we applied the Synthetic Minority Over-sampling Technique (SMOTE) and utilized Optuna for hyperparameter tuning. The findings reveal that SMLOS with the Adaboost meta-model achieved the highest accuracy of 95.50%, surpassing previous models' performances, which averaged around 85%. This contribution demonstrates the effectiveness of using SMOTE for class imbalance and Optuna for hyperparameter optimization. Integrating this model into UHTP's academic information system facilitates real-time monitoring and analysis of student data, offering a novel solution for promoting a Smart Campus through more accurate student performance predictions. This technique is not only beneficial for predicting student graduation but can also be applied to various machine learning tasks to improve data classification accuracy and stability.

Keywords: Sparsity Graduation Prediction, Machine Learning, Meta Models, SMLOS, Stacking

1. Introduction

Education is a deliberate attempt to realize the passing down of culture from one generation to the next [1]. A college is a type of educational institution that offers higher education and might be an academy, polytechnic, institute, or university. Higher education can be divided into two categories based on the program or discipline it manages: professional higher education and academic higher education. Academic higher education promotes enhancing quality and broadening scientific insights [2]. The amount of semester credit units (SKS) required at Indonesian universities varies. Strata 1 typically requires 144 to 148 credits with a minimum grade point average (GPA) of 2.00, which can be finished in 3.5 to 7 years. Students who graduate in 3.5 to 4 years are considered to be on time [3].

UHTP faces a significant challenge in achieving a 75% on-time graduation rate. UHTP defines on-time graduation as completing the required credits within four years or fewer. Despite various efforts, the university has struggled to meet this target. Therefore, understanding and predicting student graduation patterns have become crucial for implementing effective strategies to improve on-time graduation rates. UHTP is constantly taking different steps to ensure that its students graduate on time. Before implementing the policy, a basic study of students who have graduated, whether on time or not, must be conducted. Student patterns can be found in their grades and GPA while attending UHTP each semester. In addition, the number of credits has a significant impact on graduation time. Data science, which employs a variety of machine learning methods, is required to facilitate the development of these patterns [4], [5]. Several other researches have predicted college graduation. Previous research used the Naïve Bayes algorithm to predict graduation based on variables such as marital status, GPA per semester, GPA, and graduation status, achieving an accuracy of 85% [6]. Subsequent research used multiple machine learning algorithms to predict graduation, which was enhanced

***Corresponding author: Herianto (herianto.sy@gmail.com)**

DOI: https://doi.org/10.47738/jads.v5i3.316

with the ensemble method, namely boosting, resulting in greater accuracy [7]. Another study employed the ensemble method and bagging techniques to predict graduation and achieved an accuracy of 90.9% [8]. Furthermore, predictions made using various machine learning techniques demonstrate that the random forest algorithm has the highest accuracy of 77% [9].

Several articles have discussed how using a single algorithm generates lesser accuracy than the ensemble learning method. Ensemble methods are machine learning algorithms that aggregate predictions from many models to increase overall prediction accuracy [10]. Ensemble approaches include boosting, voting, bagging, and stacking [11]. This study will employ stacking to combine multiple algorithms into a model that will predict graduating on time. The stacking technique uses a meta-learner model to aggregate prediction results from many machine-learning models [12]. Previous research suggests that stacking can boost accuracy. For example, in research that employs stacking for predictions, with the highest single algorithm reaching 90%, the accuracy increases to 91% after employing the stacking technique [13]. Another study used the stacking technique to detect diabetes mellitus and obtained an accuracy of 83% [14]. Another study employed stacking to assess credit scoring and oversampling to overcome imbalanced classes, achieving an accuracy of 83.21% [15].

This research used a stacking technique with five basic algorithms, namely KNN, SVM, C4.5, RF, and NB. The meta-models used include LR, Adaboost, XGboost, LR+Adaboost, and LR+XGBoost. Apart from that, this research also uses the Synthetic Minority Over-sampling Technique (SMOTE) to overcome class imbalance [16], as well as hyperparameter tuning with Optuna to automate the search for the best parameters [17]. The combination of five algorithms using various meta-models, as well as performance enhancements with SMOTE and Optuna, is known as "SMLOS." It is intended that by conducting trials with a range of meta-models and other methodologies, we would be able to improve the performance of all models used. To ensure predictions of on-time graduation can be implemented effectively, this model will be integrated with the UHTP academic information system. This integration enables real-time monitoring and analysis of student academic data, allowing for faster and more precise implementation of relevant policies and interventions. It is envisaged that this academic information system, which incorporates machine learning, would provide more accurate suggestions to students and campus management, allowing efforts to raise the number of graduates on time to be better improved.

2. Research Methodology

This research began by collecting a dataset from the UHTP academic information system, focusing on student data relevant to graduation timelines. The process involved several key steps in developing the SMLOS (Stacking Machine Learning Optuna SMOTE) model, as shown in figure 1.

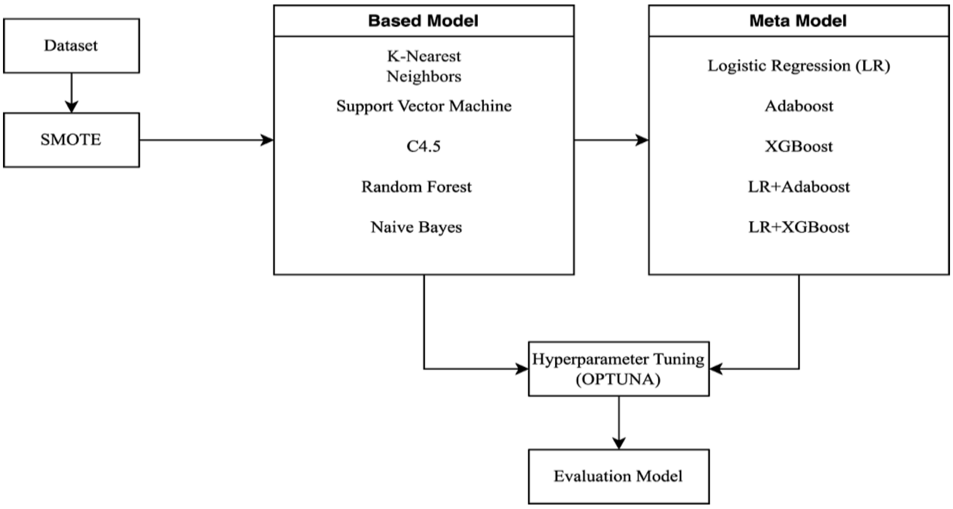
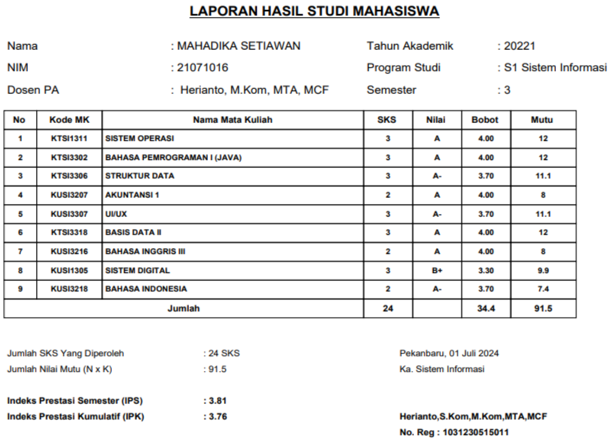


Figure 1. Development of SMLOS Model

2.1. Dataset

The data used in this study was obtained from an academic information system. Every semester, students complete a Study Plan Card containing credits, and at the end of each semester, a Study Results Card is produced that includes a summary of the student's learning results. Figure 2 shows an example of credits taken by KHS (Study Results Card) students.



LAPORAN HASIL STUDI MAHASISWA

Nama	: MAHADIKA SETIAWAN	Tahun Akademik	: 20221
NIM	: 21071016	Program Studi	: S1 Sistem Informasi
Dosen PA	: Herianto, M.Kom, MTA, MCF	Semester	: 3

No	Kode MK	Nama Mata Kuliah	SKS	Nilai	Bobot	Mutu
1	KTSI1311	SISTEM OPERASI	3	A	4.00	12
2	KTSI3302	BAHASA PEMROGRAMAN I (JAVA)	3	A	4.00	12
3	KTSI3306	STRUKTUR DATA	3	A-	3.70	11.1
4	KUSI3207	AKUNTANSI 1	2	A	4.00	8
5	KUSI3307	UI/UX	3	A-	3.70	11.1
6	KTSI3318	BASIS DATA II	3	A	4.00	12
7	KUSI3216	BAHASA INGGRIS III	2	A	4.00	8
8	KUSI1305	SISTEM DIGITAL	3	B+	3.30	9.9
9	KUSI3218	BAHASA INDONESIA	2	A-	3.70	7.4
		Jumlah	24		34.4	91.5

Jumlah SKS Yang Diperoleh : 24 SKS
Jumlah Nilai Mutu (N x K) : 91.5

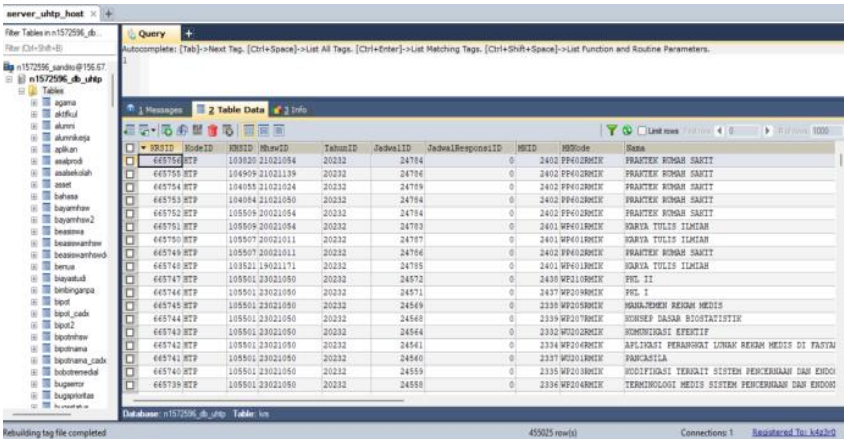
Indeks Prestasi Semester (IPS) : 3.81
Indeks Prestasi Kumulatif (IPK) : 3.76

Pekanbaru, 01 Juli 2024
Ka. Sistem Informasi

Herianto,S.Kom,M.Kom,MTA,MCF
No. Reg : 1031230515011

Figure 2. The Card of Students' Study Result

The credits taken are then entered into a database, and the data in the database is then used as a dataset. Figure 3 is the database used as the dataset for this research. In this research, the data used includes students from the Class of 2016-2019 who have already graduated, and it can be determined whether students graduated on time or not on time. The dataset consists of features such as student ID and academic performance, including grades from semesters 6 to 8. To prepare the data for analysis, the target variable, which indicates whether a student graduated on time, was encoded using a label encoder. This process involves converting categorical data into numerical values that can be used by machine learning algorithms. The target variable was encoded as 0 for not graduating on time and 1 for graduating on time. All features were encoded to ensure they were in a format suitable for model training and evaluation. By including these additional details, the research provides a clearer understanding of the data used and the preprocessing steps, such as label encoding, which are crucial for preparing the dataset for effective machine learning model development.



Autocomplete: [Tab]->Next Tag. [Ctrl+Space]->List All Tags. [Ctrl+Enter]->List Matching Tags. [Ctrl+Shift+Space]->List Function and Routine Parameters.

KRSID	KodeID	KHSID	MhswID	TahunID	JadwalID	JadwalResponsiID	MKID	MKKode	Nama
665756	HTP	103820	21021054	20232	24784	0	2402	PP602RMIK	PRAKTEK RUMAH SAKIT
665755	HTP	104909	21021139	20232	24786	0	2402	PP602RMIK	PRAKTEK RUMAH SAKIT
665754	HTP	104055	21021024	20232	24789	0	2402	PP602RMIK	PRAKTEK RUMAH SAKIT
665753	HTP	104084	21021050	20232	24784	0	2402	PP602RMIK	PRAKTEK RUMAH SAKIT
665752	HTP	105509	20021054	20232	24784	0	2402	PP602RMIK	PRAKTEK RUMAH SAKIT
665751	HTP	105509	20021054	20232	24783	0	2401	WP601RMIK	KARYA TULIS ILMIAH
665750	HTP	105507	20021011	20232	24787	0	2401	WP601RMIK	KARYA TULIS ILMIAH
665749	HTP	105507	20021011	20232	24786	0	2402	PP602RMIK	PRAKTEK RUMAH SAKIT
665748	HTP	103521	19021171	20232	24785	0	2401	WP601RMIK	KARYA TULIS ILMIAH
665747	HTP	105501	23021050	20232	24572	0	2438	WP210RMIK	PKL II
665746	HTP	105501	23021050	20232	24571	0	2437	WP209RMIK	PKL I
665745	HTP	105501	23021050	20232	24569	0	2338	WP205RMIK	MANAJEMEN REKAM MEDIS
665744	HTP	105501	23021050	20232	24568	0	2339	WP207RMIK	KONSEP DASAR BIOSTATISTIK
665743	HTP	105501	23021050	20232	24564	0	2332	WU202RMIK	KOMUNIKASI EFEKTIF
665742	HTP	105501	23021050	20232	24561	0	2334	WP206RMIK	APLIKASI PERANGKAT LUNAK REKAM MEDIS DI FASYA
665741	HTP	105501	23021050	20232	24560	0	2337	WU201RMIK	PANCASILA
665740	HTP	105501	23021050	20232	24559	0	2335	WP203RMIK	KODIFIKASI TERKAIT SISTEM PENCERNAAN DAN ENDO
665739	HTP	105501	23021050	20232	24558	0	2336	WP204RMIK	TERMINOLOGI MEDIS SISTEM PENCERNAAN DAN ENDOK

Figure 3. The Database Display of Academic Information System

2.2. SMOTE

SMOTE (Synthetic Minority Over-sampling technique) is a data preparation approach that addresses class imbalance in datasets [18]. SMOTE generates new synthetic samples for minority classes by interpolating existing minority samples [19]. The primary purpose of SMOTE is to improve minority class representation in machine learning models, allowing for better training and more accurate predictions. This strategy reduces bias towards the majority class while increasing the model's generalizability to the minority class [16]. In this study, the ratio of minority to majority class samples after oversampling was adjusted to 1:1, ensuring equal representation of both classes. This balanced ratio helps to demonstrate the extent of class balancing achieved and its impact on model performance.

2.3. Modeling With Based Algorithm

This study assessed all of the algorithms that were used as basis models to determine their performance. The results of the testing are shown in table 1.

Table 1. The Algorithm Based Used

No	Algorithm	SMOTE	OPTUNA	SMOTE+OPTUNA
1	KNN	√	√	√
2	SVM	√	√	√
3	C4.5	√	√	√
4	RF	√	√	√
5	NB	√	√	√

Table 1 shows that all algorithms use various methods to improve accuracy, such as SMOTE, Optuna, or a combination of the two.

2.4. Modeling With SMLOS

After testing the based algorithm, the next step is to integrate all of the algorithms. In this study, the algorithms are combined using a stacking technique known as SMLOS (Stacking Machine Learning Optuna SMOTE). Stacking is a strategy that employs a meta-model augmented by a based algorithm. Several prior researches employed the Logistic Regression (LR) technique as a metamodel, whereas others used something else. Table 2 shows past research that used the stacking technique.

Table 2. The previous research related to stacking

Researcher	Based Algorithm	Meta Model	Accuracy
A. Ghasemieh et al. [12]	LR, KNN, DT, RF, SVM, and XGBoost	XGBoost	88.23%
Ren, Junyu et al. [20]	LightGBM, KNN, LR, SVM, and ANN	LR	92.73%
Gupta, Aditya et al. [21]	DT, RF, SVM, and ANN	LR	93.23%
Mohapatra et al. [22]	RF, NB, LR, DT, Adaboost, KNN, and Gradient Boosting	Gradient Boosting	94.67%
Krishna et al. [13]	CNN, LSTM	LR	93.50%
Santoso et al. [23]	RF, NB, and SVM	LR	87.05%

Stacking is used because it leverages the strengths of multiple algorithms by combining their predictions through a meta-model, leading to improved predictive performance. The key advantage of stacking over other ensemble methods, such as bagging and boosting, is its ability to combine different types of base models, rather than just variations of the same model. This diversity in the models helps capture different aspects of the data, reducing overfitting and improving generalization.

In contrast, our study focuses on using only the grades from semesters 6 to 8 as variables. This decision was based on the assumption that these semesters are critical in determining a student's likelihood of graduating on time. By concentrating on these specific variables, we aimed to simplify the model while maintaining high accuracy. The results of our study show that using grades from semesters 6 to 8 provides sufficient predictive power, achieving an accuracy of 95.50% with the SMLOS technique. This is a significant improvement over previous studies that used a broader range of variables.

This study draws on various papers and employs multiple meta models to achieve the best performance. This study differs from earlier studies in several significant ways. For starters, this study used a variety of meta-models to achieve the best results, whereas past studies have tended to utilize a single model. This study also examines the effectiveness of SMOTE, Optuna, and a combination of the two to improve model performance. Table 3 displays the method and meta-model employed in this study:

Tabel 3. The Stacking Model Used

Algorithm	Metamodel
SMLOS	LR
	Adaboost
	XGBoost
	LR+Adaboost
	LR+XGBoost

Another significant difference is that this study uses the Ensemble Boosting technique as a meta-model. The boosting algorithms employed in this study are Adaboost and XGBoost. These two algorithms are not only employed as a boosting strategy but also as a single meta-model in the stacking process. This differs from earlier studies, which often employed Logistic Regression as the primary meta-model.

Thus, this research shows innovation in the use of various combinations of algorithms and optimization techniques to improve the accuracy and stability of models in text classification. This approach makes a significant contribution to improving the performance of the prediction model compared to the approach used in previous research.

2.5. Optuna

Optuna is an automatic and efficient parameter optimization tool that significantly aids in the hyperparameter tuning process in machine learning [17]. It allows users to specify a hyperparameter search space, which Optuna then explores automatically to identify the optimal set of hyperparameters that maximize or minimize a specified objective, such as model accuracy or prediction error [24]. In this study, Optuna was employed to optimize specific hyperparameters for several machine learning algorithms. For K-Nearest Neighbors (KNN), the number of neighbors was optimized within the range of 1 to 20, and the weight function was varied between uniform and distance. For the Support Vector Machine (SVM), hyperparameters such as C were tuned within the range of 0.1 to 100, with kernel options including linear, poly, rbf, and sigmoid, and gamma ranging from 0.001 to 1. The Decision Tree (C4.5) algorithm had its maximum depth optimized within a range of 1 to 50, and the minimum samples split ranged from 2 to 20. In the case of the Random Forest (RF) algorithm, the number of trees was optimized between 10 and 200, the maximum depth from 1 to 50, and the minimum samples split from 2 to 20. For Naive Bayes (NB), no hyperparameters were optimized since it is a parameter-free algorithm. Listing these hyperparameters and their ranges provides a detailed insight into the tuning process, illustrating the comprehensive search performed to identify the optimal configurations for each algorithm, which is crucial for achieving improved model performance.

2.6. Model Evaluation

Model evaluation using a confusion matrix is a highly effective method for assessing the performance of classification models [25]. The confusion matrix provides detailed information about the correct and incorrect predictions made by the model, offering a more comprehensive view of the model's performance than relying solely on metrics like accuracy [26]. It is $n \times n$ matrix, where n is the number of classes, illustrating the number of correct and incorrect predictions across different classes. For binary classification problems, the confusion matrix comprises four main components: True Positive (TP), which indicates the number of positive samples correctly predicted as positive; True Negative (TN), representing the number of negative samples correctly predicted as negative; False Positive (FP), which is the number of negative samples incorrectly predicted as positive (also known as Type I error); and False Negative (FN), which is the number of positive samples incorrectly predicted as negative (Type II error). Utilizing the confusion matrix allows for the calculation of various evaluation metrics, including accuracy, precision, recall, and F1 score, providing a nuanced understanding of model performance. This deeper insight is crucial for identifying areas where the model performs well and where it may require improvement, ultimately leading to more robust and reliable classification outcomes.

3. Result and Discussion

The data obtained from academics was then analyzed to extract information. The total number of students in the Computer Science faculty class of 2017 is 810, with 593 students graduating on time and 217 students not graduating on time. The on-time graduation rate is still considered to fall short of the targeted goal of 75%, with the class of 2017 achieving only 73%. Currently, UHTP is striving to improve this condition. UHTP has identified that student grades are a key indicator affecting on-time graduation. If this issue is left unaddressed, UHTP is concerned about the potential decline in the campus's reputation, which could reduce public trust in enrolling their high school graduates at UHTP. Figure 4 depicts the distribution of non-punctual and punctual pupils in a bar graph.

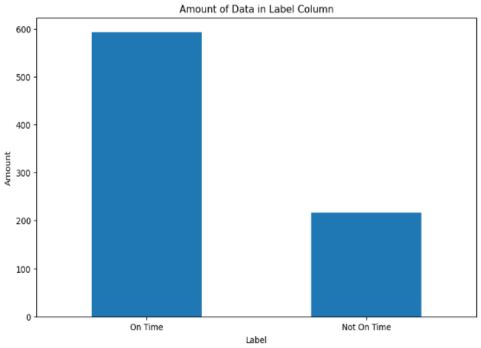


Figure 4. The Real Label Graph

Several previous studies used various criteria such as repeating courses, taking leave, and overall grade point average (GPA). However, the accuracy achieved using the decision tree algorithm was less than optimal, at 75.95% [27]. Another study used criteria such as gender, region of origin, university entrance pathway, type of tuition funding, school origin, and family finances to determine on-time graduation. This study used Bagging CART and achieved an accuracy of 85.71% [28]. These two studies show that using different sets of criteria can affect model performance.

In contrast, our study focuses on using only the grades from semesters 6 to 8 as variables. This decision was based on the assumption that these semesters are critical in determining a student's likelihood of graduating on time. By concentrating on these specific variables, we aimed to simplify the model while maintaining high accuracy. The results of our study show that using grades from semesters 6 to 8 provides sufficient predictive power, achieving an accuracy of 95.50% with the SMLOS technique. This is a significant improvement over previous studies that used a broader range of variables.

The advantage of focusing on semester grades lies in the reduction of data complexity and the avoidance of potential biases introduced by socio-economic and demographic variables. For example, while marital status and socio-economic factors can influence academic performance, they may not be as directly relevant to the prediction of on-time graduation as academic performance indicators like semester grades. By simplifying the model to include only the most relevant academic variables, our approach reduces noise and potential bias, leading to more accurate and reliable predictions.

The imbalance in the dataset poses a significant challenge, as shown in figure 5. To address this, we employed the SMOTE to generate synthetic samples for the minority class, thereby balancing the dataset. This approach helps to improve the performance and generalizability of the machine learning models by ensuring that both classes are adequately represented during training. The initial imbalance ratio was significant, with 217 students not graduating on time compared to 593 students graduating on time. After applying SMOTE, the class distribution was balanced, resulting in equal representation of both classes in the training dataset. Providing specific numbers for the class distribution before and after applying SMOTE clarifies the extent of the imbalance problem and demonstrates the effectiveness of SMOTE in addressing this issue.

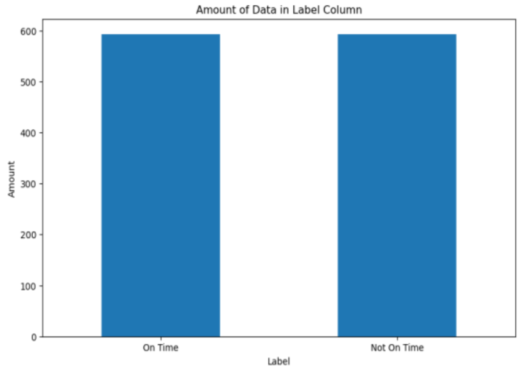


Figure 5. The Label Graph After Using SMOTE

Following the data balancing phase, the classification process is carried out utilizing the stacking technique. Based on the algorithm (KNN, SVM, C4.5, RF, and NB) with Adaboost meta-model and Optuna/SMOTE. Table 4 shows a report classification using the Adaboost meta-model.

Table 4. The Stacking

	Precision	Recall	F1-score	Support
On time	0.92	0.99	0.95	163
Not on time	0.99	0.93	0.96	193
Accuracy			0.96	356
Macro avg	0.95	0.96	0.95	356
Weight avg	0.96	0.96	0.96	356

Classification reports provide a detailed study of model performance. This report contains measures such as precision, recall, and F1-score for each class, as well as total accuracy. In this situation, the model had an overall accuracy of 96%. The 'On Time' class has a precision of 0.92, a recall of 0.99, and an F1-score of 0.95; the 'Not on Time' class has a precision of 0.99, a recall of 0.93, and an F1-score of 0.96. These metrics suggest that the model is quite good at predicting both classes, with high precision and recall. Figure 6 depicts the results of the confusion matrix.

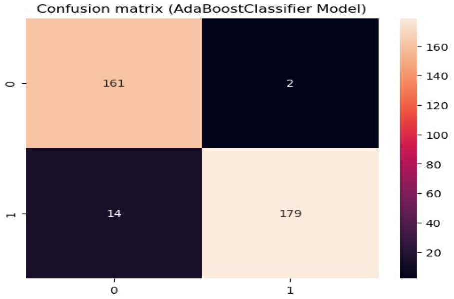


Figure 6. Confusion Matrix

Confusion Matrix provides a visual representation of model performance in terms of actual versus predicted classifications. This matrix shows that of the 163 'On Time' instances, the model correctly predicted 161 and incorrectly classified 2 as 'Not on Time'. Of the 193 'Not on Time' instances, the model correctly predicted 179 and incorrectly classified 14 as 'On Time'. This results in a high number of true positives and true negatives, indicating the power of the model in classification. Next, figure 7 is the result of the Receiver Operating Characteristic (ROC).

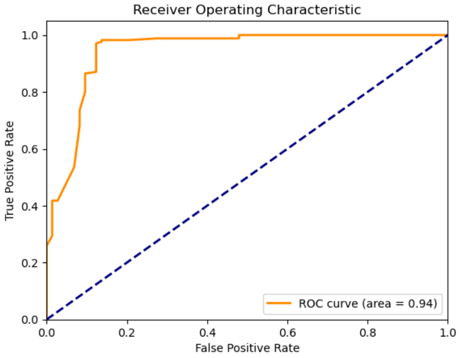


Figure 7. ROC Graph

The ROC curve and AUC of 0.98 indicate that the model performs well in distinguishing between the two classes. The ROC curve depicts the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) at various thresholds. An AUC near 1 suggests that the model is very good at distinguishing between positive and negative classes, which enhances the effectiveness of the classification model used in this study.

The confusion matrix provides further insight into the model's performance, showing the number of true positives, true negatives, false positives, and false negatives. However, certain misclassifications require detailed interpretation. False positives (students predicted to graduate on time but do not) occur due to anomalies in students' grades in the last semesters or external factors affecting their performance. False negatives (students predicted not to graduate on time but do) are caused by unexpected improvements in students' academic efforts or support received during the final semesters. Addressing these misclassifications in future research could involve incorporating additional variables such as students' attendance records, engagement in extracurricular activities, or personal circumstances affecting their academic performance. Additionally, refining the model with more granular data or employing more sophisticated techniques to handle outliers and anomalies can improve predictive accuracy.

Although the ROC curve is presented, the discussion does not delve into the trade-offs between different threshold settings. This is because the primary focus of this study is on developing and evaluating an effective predictive model using stacking and optimization techniques, rather than on detailed analysis of specific threshold settings. However, choosing an optimal threshold based on the ROC curve involves balancing sensitivity and specificity to match the application's priorities. If the goal is to maximize the identification of students at risk of not graduating on time (high sensitivity), a lower threshold is chosen, accepting a higher rate of false positives. Conversely, if the goal is to ensure that predictions of on-time graduation are highly reliable (high specificity), a higher threshold can be set, reducing false positives but increasing false negatives. Future research should include a detailed analysis of threshold selection to optimize the balance between these trade-offs according to the specific goals and priorities of the institution.

After processing with the Adaboost meta-model, table 5 compares the accuracy with various meta-models to demonstrate how the techniques used improve overall model performance.

Table 5. The Result of Stacking Comparison

Algorithm Based	Meta Model	Without SMOTE	SMOTE	Optuna	SMOTE+Optuna (SMLOS)
KNN, SVM, C.45, RF, NB	LR	95.06%	95.22%	95.06%	94.94%
KNN, SVM, C.45, RF, NB	XGBoost	95.06%	94.94%	95.06%	95.22%
KNN, SVM, C.45, RF, NB	Adaboost	95.06%	94.38%	94.23%	95.50%
KNN, SVM, C.45, RF, NB	LR+XGBoost	95.06%	94.94%	95.06%	95.22%
KNN, SVM, C.45, RF, NB	LR+Adaboost	95.06%	94.94%	95.22%	95.22%

Table 5 compares the outcomes of various combinations of fundamental algorithms (KNN, SVM, C.45, RF, and NB) and meta models (LR, XGBoost, Adaboost, LR+XGBoost, LR+Adaboost) utilizing different data processing methods, such as without SMOTE, SMOTE, Optuna, and SMOTE+Optuna combo (SMLOS). Overall, the results reveal that the Adaboost meta model with the SMOTE+Optuna (SMLOS) combination achieves the highest accuracy of 95.50%. This was followed by the combos LR+Adaboost and LR+XGBoost, both of which obtained 95.22% accuracy using the SMLOS method.

Using SMOTE alone does not result in a significant improvement in several combinations, such as LR, XGBoost, and LR+XGBoost, where accuracy remains consistent or slightly declines as compared to not using SMOTE. Using Optuna yielded different outcomes, with some combinations experiencing enhanced accuracy and others not. The combination sans SMOTE and Optuna achieves a reasonably consistent accuracy of 95.06% across all meta-models, indicating that the core algorithm is quite strong. However, the introduction of SMLOS leads to a significant improvement in certain meta-models, particularly Adaboost, indicating that this combination of approaches can produce more optimal results. The effectiveness of SMOTE is particularly evident in the improvement of minority class predictions. Before applying SMOTE, the model struggled to accurately predict the minority class, leading to a high rate of false negatives. After applying SMOTE, the number of correct predictions for the minority class increased significantly, reducing the rate of false negatives and improving the overall balance of the model's predictions. This demonstrates the direct benefits of SMOTE in enhancing the model's ability to correctly identify instances from the minority class, which is critical in the context of this study.

This discussion shows that the choice of meta-model and data processing method has a significant impact on the model's final performance. The combination of SMLOS with the Adaboost meta-model was shown to produce the best outcomes in the circumstances investigated in this study. The research also tested the based algorithm. Table 6 displays the performance of the based algorithm.

Table 6. The Performance Result of Algorithm Based

Algorithm	Without SMOTE	SMOTE	OPTUNA	SMOTE + OPTUNA
KNN	95.06%	95.48%	95.06%	95.22%
SVM	81.06%	69.66%	95.47%	95.22%
C45	93.82%	93.53%	87.85%	95.22%
RF	95.06%	93.82%	95.06%	94.38%
NB	76.13%	70.78%	76.13%	70.78%

Table 6 shows the performance results of various basic algorithms (KNN, SVM, C45, RF, NB) with four different approaches: without SMOTE, with SMOTE, with Optuna, and a combination of SMOTE+Optuna. In the KNN algorithm, the performance results show a small increase when using SMOTE (95.48%) and the combination of SMOTE+Optuna (95.22%), but remain consistent without significant changes with or without Optuna. The SVM algorithm experienced a significant increase in performance with the application of Optuna (95.47%) and the SMOTE+Optuna combination (95.22%), showing the effectiveness of hyperparameter optimization in improving SVM performance.

For the C45 algorithm, the SMOTE+Optuna combination improves performance significantly (95.22%), while using simply Optuna results in a modest performance loss (87.85%). The RF method remained consistent with or without Optuna, but there was a minor performance reduction when utilizing SMOTE and the SMOTE+Optuna combination. Finally, the MNB algorithm shows suboptimal results with the use of SMOTE, either alone or in combination with Optuna, with the best performance without SMOTE or Optuna (76.13%).

When compared to the stacking results in table 4, where the combination of algorithms employs multiple meta-models such as Logistic Regression, XGBoost, and Adaboost, the stacking performance is more consistent and produces higher results. For example, the Adaboost meta-model combined with SMLOS receives the highest performance with an accuracy of 95.50%, whereas the XGBoost meta-model achieves 95.22%. This demonstrates that the stacking method, which combines many basic algorithms, can produce better and more consistent results than utilizing a single basic algorithm optimized using SMOTE and Optuna.

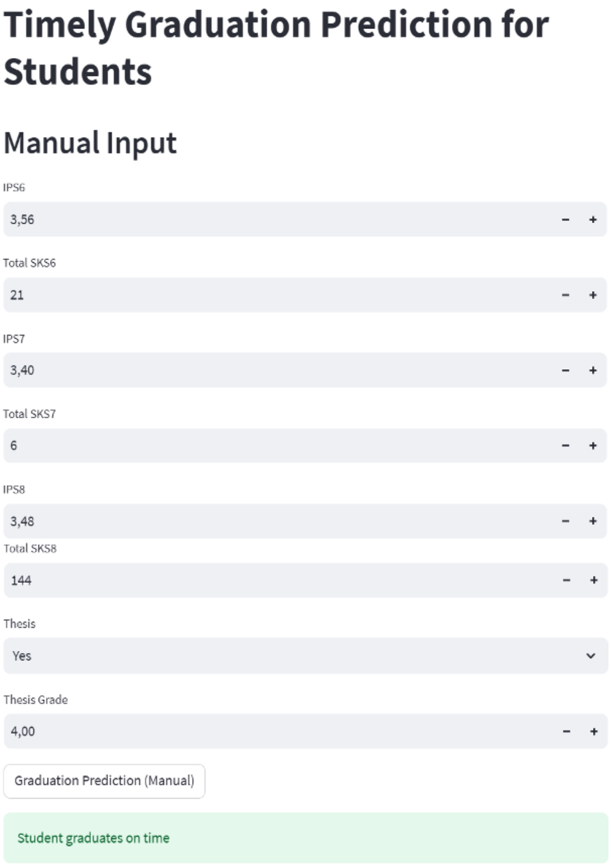
Overall, this study demonstrates that the utilization of stacking approaches, particularly with proper meta-models, can produce better outcomes than individually optimized fundamental algorithms. The combination of SMOTE and Optuna improves the performance of several fundamental algorithms, but the stacking method provides a more comprehensive approach to enhancing model accuracy and stability in text data classification. Then this study compares to past research and is superior. Table 7 presents a comparison to past research.

Table 7. The Comparison with Previous Research

No	Researcher	Based Algorithm	Meta Model	Accuracy
1	A. Ghasemieh et al. [12]	LR, KNN, DT, RF, SVM, and XGBoost	XGBoost	88.23%
2	Ren, Junyu et al. [20]	LightGBM, KNN, LR, SVM, and ANN	LR	92.73%
3	Gupta, Aditya et al. [21]	DT, RF, SVM, and ANN	LR	93.23%
4	Mohapatra et al. [22]	RF, NB, LR, DT, Adaboost, KNN, and Gradient Boosting	Gradient Boosting	94.67%
5	Santoso et al. [23]	RF, NB, and SVM	LR	87.05%
6	Almohimeed [29]	RF, DT, SVM, LR, KNN, and NB	RF	92.08%
7	Zhao et al. [30]	KNN, LR, SVM, DT, RF, ET, GNB, Adaboost, GBDT, XGBoost, LightGBM, and CatBoost	LR	87.00%
8	Samreen [31]	NB, KNN, LR, DT, SVM, GB, Adaboost, and RF	RF	94.40%
9	Muslim et al. [32]	KNN, SVM, and RF	XGBoost	91.43%
10	Kumar [33]	LR, SVM, RF, and DT	XGBoost	83.45%
11	This Research (SMLOS)	KNN, SVM, C4.5, RF, and NB	Adaboost	95.50%

Table 7 compares the findings of this study to prior studies that used various fundamental algorithms and metamodels for data classification. This study attained a maximum accuracy of 95.50% by combining fundamental KNN, SVM, C4.5, RF, and NB algorithms, as well as the Adaboost meta-model in the SMLOS method. This demonstrates that the method utilized in this study outperformed earlier research, which had an accuracy range of 83.45% to 94.67%. This study provides a substantial contribution to enhancing data classification accuracy by combining the SMLOS technique with the Adaboost metamodel.

The final stage in this research is to run tests on the best model in Streamlit. Streamlit is an open-source library for developing interactive web apps quickly and effortlessly. The Streamlit application demonstrated offers two key capabilities for forecasting on-time graduation for students: human input and Excel file-based input. Users can manually add student data into the manual input field using many input fields, such as Semester 6 IP, 7th Semester IP, 8th Semester IP, total 6th Semester SKS, 7th Semester SKS, 8th Semester SKS, thesis status, and thesis grades. After entering all of the data, the user can click the "Graduation Prediction (Manual)" button to get a prediction of whether the student will graduate on time or not. In figure 8 the prediction results will be displayed below the button with text such as "Students graduated on time" and "Students graduated not on time".



Timely Graduation Prediction for Students

Manual Input

IPS6
3,56

Total SKS6
21

IPS7
3,40

Total SKS7
6

IPS8
3,48

Total SKS8
144

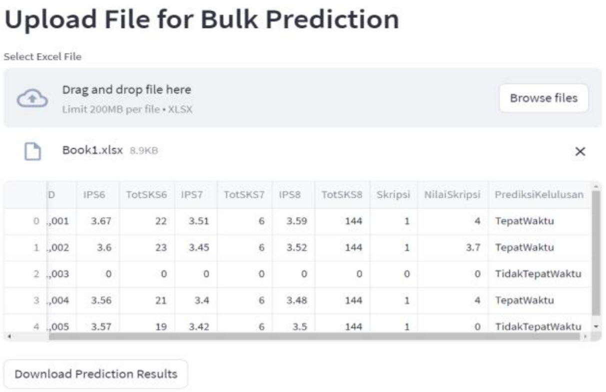
Thesis
Yes

Thesis Grade
4,00

Student graduates on time

Figure 8. Manual Prediction with Streamlit

The Excel file-based input feature allows users to upload Excel files containing student data for bulk predictions. Users can upload Excel files by dragging and dropping them into specific places or by clicking the "Browse files" button and selecting files from their PC. Once the file is submitted, the data will be presented in a table beneath the upload area. This table has columns such as 6th Semester IP, 7th Semester IP, 8th Semester IP, total 6th Semester SKS, 7th Semester SKS, 8th Semester SKS, undergraduate-thesis status, undergraduate-thesis grades, and a graduation prediction column that will be displayed with projected results, such as "On Time" or "Not on Time". After the data is processed, the user can press the "Download Prediction Results" button to download the prediction results in the form of an Excel file as seen in figure 9.



Upload File for Bulk Prediction

Select Excel File

Book1.xlsx 8.9KB

	D	IPS6	TotSKS6	IPS7	TotSKS7	IPS8	TotSKS8	Skripsi	NilaiSkripsi	PrediksiKelulusan
0	.,001	3.67	22	3.51	6	3.59	144	1	4	TepatWaktu
1	.,002	3.6	23	3.45	6	3.52	144	1	3.7	TepatWaktu
2	.,003	0	0	0	0	0	0	0	0	TidakTepatWaktu
3	.,004	3.56	21	3.4	6	3.48	144	1	4	TepatWaktu
4	.,005	3.57	19	3.42	6	3.5	144	1	0	TidakTepatWaktu

Figure 9. Automatic Prediction with Streamlit

With these features, the Streamlit application provides flexibility for users to make graduation predictions either individually via manual input or en masse via Excel file uploads.

4. Conclusion

This study succeeded in inventing the SMLOS technique, which combines various basic algorithms with the Adaboost metamodel to increase data classification accuracy by up to 95.50%. These findings demonstrate that using SMOTE to solve class imbalance and Optuna for hyperparameter adjustment improves classification model performance significantly. The integration of this machine learning model into the UHTP academic information system allows for real-time monitoring and analysis of student academic data, resulting in more accurate recommendations to students and campus management to promote Smart Campus [34]. This technique is not only useful for forecasting student graduation but it can also be used in a variety of other machine learning applications to improve the accuracy and stability of data categorization models.

Future research should look into combining other machine learning algorithms and implementing more complex ensemble learning methods to overcome challenges in more complex data analysis. For instance, exploring the use of neural network-based models such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) could provide deeper insights into time-series or spatial data. Additionally, investigating advanced ensemble methods like Gradient Boosting Machines (GBM) or LightGBM could further enhance model performance. Furthermore, integrating these advanced models with UHTP's academic information system could support more nuanced decision-making processes, including personalized student interventions and predictive maintenance for educational resources.

5. Declarations

5.1. Author Contributions

Conceptualization: H., B.K., Z.H.H., Y.I., and M.K.A.; Methodology: Z.H.H. and M.K.A.; Software: H. and B.K.; Validation: H., Z.H.H., and M.K.A.; Formal Analysis: H., B.K., and M.K.A.; Investigation: B.K.; Resources: Z.H.H.; Data Curation: Y.I.; Writing Original Draft Preparation: H. and B.K.; Writing Review and Editing: Z.H.H., Y.I., and M.K.A.; Visualization: Y.I.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

Acknowledgments are addressed to Direktorat Riset, Teknologi dan Pengabdian kepada Masyarakat Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Indonesia for providing funding in the Fundamental Research Scheme - Regular Research Program 2024, as well as Universitas Hang Tuah Pekanbaru which has provided facilities for conducting research activities.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Seprie, "Contemporary Problems in Early Childhood Education (Case Study at the President Filia Gracia Foundation Puruk Cahu Kindergarten)," *JOURNAL SYNTAX IDEA*, vol. 6, no. 1, pp. 45–61, 2024, doi: 10.46799/syntax-idea.v6i1.2815.

[2] Devaki, "E-Learning: Online Teaching Experiences of Higher Education Teachers in India During the Covid19 Pandemic," *ELS Journal on Interdisciplinary Studies in Humanities*, vol. 6, no. 4, pp. 646–657, 2023, doi: 10.34050/elsjish.v6i4.20298.

[3] M. F. Isbah, W. Kustiningsih, G. R. Wibawanto, O. A. Artosa, N. Kailani, and I. Zamjani, "Strategies to Enhance the

Employability of Higher Education Graduates in Indonesia: A Way Forward," *Society*, vol. 11, no. 2, pp. 398–414, Dec. 2023, doi: 10.33019/society.v11i2.592.

[4] 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. 259–265, Dec. 2022, doi: 10.35877/454ri.asci1581.

[5] Y. Irawan, "Implementation of Data Mining For Determining Majors Using K-Means Algorithm In Students of Sma Negeri 1 Pangkalan Kerinci," *Journal of Applied Engineering and Technological Science*, vol. 1, no. 1, pp. 17–29, 2019, doi: 10.37385/jaets.v1i1.18.

[6] S. Mehta, "Playing Smart with Numbers: Predicting Student Graduation Using the Magic of Naive Bayes," *International Transactions on Artificial Intelligence (ITALIC)*, vol. 2, no. 1, pp. 60–75, 2023, doi: 10.33050/italic.v2i1.405.

[7] A. Desfiandi and B. Soewito, "Student Graduation Time Prediction Using Logistic Regression, Decision Tree, Support Vector, And Adaboost Ensemble Learning," *International Journal of Information System and Computer Science) IJISCS*, vol. 7, no. 3, pp. 195–199, 2023, doi: 10.56327/ijiscs.v7i2.1579.

[8] K. Okoye, J. T. Nganji, J. Escamilla, and S. Hosseini, "Machine learning model (RG-DMML) and ensemble algorithm for prediction of students' retention and graduation in education," *Computers and Education: Artificial Intelligence*, vol. 6, no. 1, pp. 1–13, Jun. 2024, doi: 10.1016/j.caeai.2024.100205.

[9] A. Sadqui, M. Ertel, H. Sadiki, and S. Amali, "Evaluating Machine Learning Models for Predicting Graduation Timelines in Moroccan Universities," *IJACSA (International Journal of Advanced Computer Science and Applications)*, vol. 14, no. 7, pp. 304–310, 2023, doi: 10.14569/IJACSA.2023.0140734.

[10] P. P. Putra, M. K. Anam, S. Defit, and A. Yunianta, "Enhancing the Decision Tree Algorithm to Improve Performance Across Various Datasets," *INTENSIF*, vol. 8, no. 2, pp. 200–212, Aug. 2024, doi: 10.29407/intensif.v8i2.22280.

[11] P. Mahajan, S. Uddin, F. Hajati, and M. A. Moni, "Ensemble Learning for Disease Prediction: A Review," *Healthcare*, vol. 11, no.7, pp. 1-21, Jun. 01, 2023, doi: 10.3390/healthcare11121808.

[12] A. Ghasemieh, A. Lloyed, P. Bahrami, P. Vajar, and R. Kashef, "A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients," *Decision Analytics Journal*, vol. 7, no 1, pp. 1–13, Jun. 2023, doi: 10.1016/j.dajour.2023.100242.

[13] B. L. V. S. R. Krishna, V. Mahalakshmi, and G. K. M. Nukala, "A Stacking Model for Outlier Prediction using Learning Approaches," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 2s, pp. 629–638, 2023.

[14] A. A. Alzubaidi, S. M. Halawani, and M. Jarrah, "Towards a Stacking Ensemble Model for Predicting Diabetes Mellitus using Combination of Machine Learning Techniques," *IJACSA (International Journal of Advanced Computer Science and Applications)*, vol. 14, no. 12, pp. 348–358, 2023, doi: 10.14569/IJACSA.2023.0141236.

[15] R. Rofik, R. Aulia, K. Musaadah, S. S. F. Ardyani, and A. A. Hakim, "Optimization of Credit Scoring Model Using Stacking Ensemble Learning and Oversampling Techniques," *Journal of Information System Exploration and Research*, vol. 2, no. 1, pp. 11–20, Dec. 2023, doi: 10.52465/joiser.v2i1.203.

[16] 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, Mar. 2023, doi: 10.3390/app13064006.

[17] W. A. G. Kodri and S. Hadianti, "Optimization of The Machine Learning Approach using Optuna in Heart Disease Prediction," *Journal Medical Informatics Technology*, vol. 1, no. 3, pp. 59–64, Sep. 2023, doi: 10.37034/medinftech.v1i3.15.

[18] L. L. Van Fc, M. K. Anam, M. B. Firdaus, Y. Yunefri, and N. A. Rahmi, "Enhancing Machine Learning Model Performance in Addressing Class Imbalance," *COGITO Smart Journal*, vol. 10, no. 1, pp. 478–490, 2024.

[19] 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.

[20] J. Ren, H. Wan, C. Zhu, and T. Qin, "Stacking ensemble learning with heterogeneous models and selected feature subset for prediction of service trust in internet of medical things," *IET Inf Secur*, vol. 17, no. 2, pp. 269–288, Mar. 2023, doi: 10.1049/ise2.12091.

[21] A. Gupta, V. Jain, and A. Singh, “Stacking Ensemble-Based Intelligent Machine Learning Model for Predicting Post-COVID-19 Complications,” *New Gener Comput*, vol. 40, no. 4, pp. 987–1007, Dec. 2022, doi: 10.1007/s00354-021-00144-0.

[22] S. Mohapatra, I. Mishra, and S. Mohanty, “Stacking Model for Heart Stroke Prediction using Machine Learning Techniques,” *EAI Endorsed Trans Pervasive Health Technol*, vol. 9, no. 1, pp. 1–5, May 2023, doi: 10.4108/eetpht.9.4057.

[23] D. B. Santoso, A. Munna, and D. H. U. Ningsih, “Improved playstore review sentiment classification accuracy with stacking ensemble,” *Journal of Soft Computing Exploration*, vol. 5, no. 1, pp. 38–45, Mar. 2024, doi: 10.52465/joscex.v5i1.247.

[24] A. Tikaningsih, P. Lestari, A. Nurhopipah, I. Tahyudin, E. Winarto, and N. Hassa, “Optuna Based Hyperparameter Tuning for Improving the Performance Prediction Mortality and Hospital Length of Stay for Stroke Patients,” *Telematika*, vol. 17, no. 1, pp. 1–16, Feb. 2024, doi: 10.35671/telematika.v17i1.2816.

[25] Hamdani, Randi N.A, and M. K. Anam, “Comparison of Support Vector Machine and Random Forest Algorithms for Analyzing Online Loans on Twitter social media,” *JAIA-Journal Of Artificial Intelligence And Applications*, vol. 4, no. 1, pp. 8–16, 2024, doi: 10.33372/jaia.v4i1.1087.

[26] L. K. Ramasamy, S. Kadry, Y. Nam, and M. N. Meqdad, “Performance analysis of sentiments in Twitter dataset using SVM models,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2275–2284, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2275-2284.

[27] O. P. Moerdyanto and I. K. D. Nuryana, “Predicting On-Time Graduation Using Decision Tree Approach Decision Tree Algorithm,” *Journal of Informatics and Computer Science*, vol. 5, no. 1, pp. 90–96, 2023, Accessed: Jul. 26, 2024.

[28] W. Agwil, H. Fransiska, and N. Hidayati, “Analysis of Student Graduation Timeliness Using Bagging CART,” *FIBONACCI*, vol. 6, no. 2, pp. 155–166, Dec. 2020, doi: 10.24853/fbc.6.2.155-166.

[29] A. Almohimeed et al., “Explainable Artificial Intelligence of Multi-Level Stacking Ensemble for Detection of Alzheimer’s Disease Based on Particle Swarm Optimization and the Sub-Scores of Cognitive Biomarkers,” *IEEE Access*, vol. 11, no. 1, pp. 123173–123193, 2023, doi: 10.1109/ACCESS.2023.3328331.

[30] N. Zhao, X. Li, Y. Ma, H. Wang, S. J. Lee, and J. Wang, “Improved stacked ensemble with genetic algorithm for automatic ECG diagnosis of children living in high-altitude areas,” *Biomed Signal Process Control*, vol. 87, no. 1, pp. 1–11, Jan. 2024, doi: 10.1016/j.bspc.2023.105506.

[31] S. Samreen, “Memory-efficient, accurate and early diagnosis of diabetes through a machine learning pipeline employing crow search-based feature engineering and a stacking ensemble,” *IEEE Access*, vol. 9, no. 1, pp. 134335–134354, 2021, doi: 10.1109/ACCESS.2021.3116383.

[32] M. A. Muslim et al., “New model combination meta-learner to improve accuracy prediction P2P lending with stacking ensemble learning,” *Intelligent Systems with Applications*, vol. 18, no. 1, pp. 1–8, May 2023, doi: 10.1016/j.iswa.2023.200204.

[33] C. U. O. Kumar, I. Singh, and M. Suguna, “Optimizing Patient Recruitment for Clinical Trials: A Hybrid Classification Model and Game-Theoretic Approach for Strategic Interaction,” *IEEE Access*, vol. 12, no. 1, pp. 10254–10280, 2024, doi: 10.1109/ACCESS.2024.3351688.

[34] M. K. Anam, A. Yunianta, H. J. Alyamani, Erlin, A. Zamsuri, and M. B. Firdaus, “Analysis and Identification of Non-Impact Factors on Smart City Readiness Using Technology Acceptance Analysis: A Case Study in Kampar District, Indonesia,” *Journal of Applied Engineering and Technological Science*, vol. 5, no. 1, pp. 1–17, 2023, doi: 10.37385/jaets.v5i1.2401.