Indonesian Journal of Electronics, Electromedical Engineering, and Medical Informatics e-ISSN: 2656-8624
Homepage: https://ijeeemi.org/; Vol. 7, No. 2, pp. 415-426, May 2025

RESEARCH PAPER **OPEN ACCES**

Application of Adaboost Algorithm with SMOTE and Optuna Techniques in Sleep Disorder Classification

Muhammad Naufal Anshory, Muhammad Itqan Mazdadi, Triando Hamonangan Saragih, Irwan Budiman, Setyo Wahyu Saputro

Department of Computer Science, Lambung Mangkurat University, Banjarbaru, Indonesia

ABSTRACT

Data imbalance is a serious challenge in developing machine learning models for sleep disorder classification. When models are trained on an uneven distribution of classes, classification performance for minority classes such as insomnia and sleep apnea is often low. As a result, the overall accuracy may seem elevated, yet the sensitivity to important cases to be weak. Therefore, this research aims to design and develop a robust sleep disorder classification model with the AdaBoost algorithm, with improved performance through the integration of two main approaches, namely data balancing technique utilizing SMOTE and hyperparameter optimization using Optuna. This research contributes by showing that the combination of the two approaches can significantly improve model performance, not only in terms of global accuracy, but also accuracy on previously overlooked minority classes. The dataset utilized is the Sleep Health and Lifestyle Dataset which consists of 374 synthesized data and is divided into three categories: insomnia, sleep apnea, and none. This method stages include data preprocessing, data division using train-test split (80:20), application of SMOTE to balance the class distribution, hyperparameter tuning using Optuna, and model training with the AdaBoost algorithm. Evaluation was performed using classification metrics: accuracy, precision, recall, and F1-score. Results showed that mix of SMOTE and Optuna yielded the best results, accuracy 90.6%, F1-score 0.83871 for insomnia, and 0.81250 for sleep apnea. This performance was consistently superior to scenarios with no SMOTE or no tuning. This confirms the importance of using combination strategies to obtain fair and accurate classification on medical data. Future research is recommended to use real datasets as well as test the capabilities of this research on other models such as XGBoost or LightGBM.

PAPER HISTORY

Received March 05, 2025
Accepted May 10, 2025
Published May 21, 2025

KEYWORDS

Sleep Disorder;
Adaboost;
SMOTE;
Optuna

AUTHOR EMAIL

muhammad.naufal210@gmail.com
mazdadi@ulm.ac.id
triando.saragih@ulm.ac.id
irwan.budiman@ulm.ac.id
setyo.saputro@mhs.ulm.ac.id

1. INTRODUCTION

Sleep disorders have become a growing global health issue, affecting nearly 40% of the population and contributing to both physical and mental health problems, such as emotional instability, weakened immunity, and chronic illnesses [1]. These disturbances are common not only in adults but also among adolescents, with an estimated 20–30% experiencing sleep-related issues influenced by genetics, environment, and lifestyle factors [2]. Poor sleep quality has also been linked to behavioral and emotional challenges, especially in younger individuals [3]. Despite its importance, diagnosing sleep disorders remains complex due to symptom overlap with other conditions, emphasizing the need for more efficient and accurate diagnostic methods.

In the last few years, Artificial Intelligence (AI) AI technology already sees widespread implementation in the healthcare sector, particularly for identifying and classifying sleep disorders. A range of machine learning techniques has been utilized to examine patients' physiological signals, such as electroencephalogram (EEG) and polysomnography (PSG), in order to enhance the precision of sleep disorder diagnosis [4]. These techniques enable systems to identify abnormal sleep patterns more quickly and efficiently compared to conventional methods [5]. In addition, AI-based approaches open opportunities for the development of systems capable of operating automatically and independently in detecting sleep pattern abnormalities in patients [6].

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

One of the primary difficulties in utilizing machine learning for classifying sleep disorders lies in the data imbalance commonly found in healthcare datasets. These datasets frequently exhibit a disproportionate distribution, with significantly fewer samples representing certain classes. Such imbalance often leads machine learning models to perform better in detecting the dominant class while struggling with the less-represented ones [7]. To overcome this challenge, several methods have been proposed, involving the use of the Synthetic Minority Over-sampling Technique (SMOTE), which enhances the presence of underrepresented classes to promote more balanced model training [8]. Research by [9] demonstrated the application of SMOTE in sleep disorder classification, resulting in an accuracy rate of 96.88% on imbalanced datasets.

In addition to data balancing techniques, selecting the appropriate algorithm is also crucial for improving the accuracy of sleep disorder classification. One commonly used algorithm is Adaptive Boosting (AdaBoost), which works by combining several weak learners into a stronger model to enhance classification performance [10]. AdaBoost has been proven effective in various healthcare studies and can improve the detection of abnormal sleep patterns [11]. A study conducted by [12] utilized AdaBoost and achieved 90.26% accuracy in pediatric OSA classification. Nevertheless, AdaBoost's effectiveness is significantly influenced by the choice of optimal parameters. Therefore, efficient hyperparameter optimization techniques are required, one of which is Optuna, which automatically searches for the best parameter combinations to enhance machine learning model performance [13]. For example, a study conducted by [14] using Optuna hyperparameter optimization achieved an accuracy of 95.45% for cardiovascular disease classification.

Grounded in the outlined issues and suggested approaches, this research emphasizes implementing AdaBoost integrated with SMOTE alongside Optuna, aiming to enhance classification accuracy in detecting sleep disorders. Through this approach, it is expected that the system will be able to detect sleep disorders more effectively, thereby supporting medical professionals in providing more accurate also efficient diagnoses.

This study offers several things key contributions: 1) introducing conceptual developments within medical machine learning, with a specific focus on utilizing the AdaBoost algorithm for classifying sleep disorders; 2) offering academic insights into improving model accuracy through data resampling techniques based on SMOTE and hyperparameter optimization using Optuna; 3) offering actionable approaches to address data imbalance in medical datasets, a common obstacle that complicates the deployment of machine learning algorithms within clinical environments; 4) facilitating the advancement of intelligent diagnostic technologies by applying a refined integration of resampling techniques and hyperparameter tuning strategies.

The organization of the organization of this paper is as follows: Part II describes the datasets used, the proposed approach, as well as the steps involved in model training and validation. Part III outlines the outcomes related to data distribution before and after applying the resampling technique, the best parameters obtained from the tuning process, as well as the performance metrics of the classification model under various conditions. Part IV provides an in-depth discussion of the results, highlights comparisons with previous studies, and addresses the limitations identified in this work. Finally, Part V concludes the study by summarizing the objectives, key findings, and recommendations for future investigations.

2. MATERIALS AND METHOD

Fig. 1. **Flowchart of the research methodology illustrating each stage of the sleep disorder classification process using SMOTE, Optuna, and AdaBoost.**

Fig. 1 presents the workflow of the research methodology employed in this study, starting from the data collection stage to the model evaluation phase. Each phase is methodically structured to build a highly effective classification model for sleep disorders. This workflow

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

consists of several crucial steps, including data acquisition, preprocessing, dataset partitioning, the utilization of SMOTE to handle class imbalance, hyperparameter optimization with Optuna, model training via the AdaBoost algorithm, and ultimately, performance assessment using appropriate evaluation metrics.

A. Data Collection

The dataset utilized in this research is the Sleep Health and Lifestyle Dataset, sourced from Kaggle. It comprises 374 entries and 13 variables, which encompass factors related to sleep behaviors and lifestyle, including age, gender, sleep duration, sleep quality, physical activity level, stress level, BMI classification, blood pressure, heart rate, daily step count, and the occurrence of sleep disorders (none, insomnia, sleep apnea). Comprehensive information about all dataset attributes is presented at Table 1.

Table 1. A summary of the attributes included in the Sleep Health and Lifestyle dataset.

Features	Description
Personal ID	A unique code assigned to identify each participant.
Gender	The biological sex of the individual (Male or Female).
Age	The person's age represented in years.
Occupation	Type of job or professional role held by the individual.
Sleep Duration	The total amount of time the individual sleeps each day, measured in hours.
Quality of Sleep	Self-rated sleep quality index using a rating scale ranging from 1 to 10.
Physical Activity Level	Duration of an individual's physical activity per day in minutes.
Stress Level	Self-reported measure of stress intensity, rated using a rating scale ranging from 1 to 10.
BMI Category	Categories of an individual's Body Mass Index (BMI), including Underweight, Normal weight, or Overweigh.
Blood Pressure	Individual blood pressure is displayed as systolic versus diastolic values.
Heart Rate	Number of heartbeats per minute at rest.
Daily Steps	Total steps a person takes in a day.
Sleep Disorder	Indicates whether the individual has a sleep disorder, and if so, the type (None, Insomnia, Sleep Apnea).

B. Data Processing

Before developing the model, several data preprocessing steps were conducted as follows:

1) Data Cleaning

Data cleaning, also known as data cleansing or data scrubbing, involves assessing the quality of data by making necessary modifications, altering, or removing data considered irrelevant, incomplete, inaccurate, or improperly formatted within a database, with the goal of producing high-quality data [15]. In this study, data cleaning was performed by removing attributes irrelevant to the classification process, such as Person ID and Gender.

2) Categorical Data Encoding

The process of converting categorical data into a numerical format is known as label encoding. This enables machine learning algorithms to process the data [16]. This research is, Label Encoding was applied to transform categorical fields such as BMI Category, Blood Pressure, Sleep Disorder, and Occupation into numeric format.

3) Feature Scaling

Feature scaling is an essential preprocessing step that standardizes data ranges through techniques such as normalization and standardization [17]. In this study, the method used for feature scaling was MinMaxScaler. MinMaxScaler is a scaling technique that transforms feature values into a specific range, typically between 0 and 1 [18]. The formula for MinMaxScaler can be seen this Eq. (1) [19].

$$X_{std} = \frac{(X - X.\min)}{(X.\max - X.\min)} \quad (1)$$

C. Data Splitting

Dividing data into training, testing, and validation sets is a standard procedure in machine learning, used to support both model building and performance assessment [20]. In this research, the train-test split approach was utilized. This method randomly divides the dataset into two subsets one for training and the other for testing to ensure unbiased and representative model assessment [21]. The dataset was divided using the train_test_split function from the scikit-learn library, specifying a test size of 0.2, which means 20% of the data was reserved for testing while the remaining 80% was used for training. The 80:20 division was selected due to its simplicity and widespread adoption in machine learning studies, particularly when working with datasets of limited size. This ratio offers a practical balance between having enough samples for model learning and maintaining sufficient data for reliable evaluation. Furthermore, this split configuration is well-

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

established in the literature and supports both computational efficiency and evaluation consistency.

D. SMOTE

SMOTE (Synthetic Minority Oversampling Technique) is a widely used oversampling method that addresses class imbalance by generating synthetic data from the minority class within the training set, although it may also amplify noisy instances [22]. SMOTE carries out oversampling through locating instances belonging to the minority class and selecting their k-nearest neighbors for each of them. Rather than simply duplicating existing data points, it generates new synthetic examples. This strategy effectively reduces the risk of overfitting caused by repeated instances [23]. The systematic steps of SMOTE implementation for handling class imbalance are outlined in Table 2 [24]. In this study, the SMOTE technique was applied using the implementation provided by the imblearn library, with the default parameters. Specifically, k_neighbors set to the default of 5, and no additional parameter tuning was performed. The only parameter explicitly defined was random_state=42, in order to maintain consistent and repeatable results.

Table 2. Summary of SMOTE implementation steps for handling class imbalance in a classification dataset. Each phase is essential for building an effective and balanced predictive model.

SMOTE Implementation for Addressing Class Imbalance
Step 1: Data Preparation
Handle missing entries in the dataset
Convert categorical attributes into numerical representations
Normalize or standardize numerical values
Step 2: Class Distribution Analysis
Detect and separate the underrepresented and dominant classes in the dataset
Step 3: SMOTE Application
Load essential libraries
Divide the dataset into two portions: one designated for training and the other for evaluation.
Utilize SMOTE on the training dataset to create synthetic instances for underrepresented classes
Train a machine learning model using the balanced training sets
Employ the test dataset to evaluate the performance of the model
Assess the model's effectiveness through evaluation metrics such as accuracy, precision, recall, and the F1-score.
Step 4: Exploring Other Oversampling Techniques
Investigate alternative oversampling strategies suitable for the dataset

For further clarification, Fig. 2 [25] illustrates the comparison between data distribution before and after applying SMOTE. Prior to resampling, there were several minority instances considered abnormal, as indicated by separate clusters. After the application of SMOTE, the distribution of minority data became more balanced by synthesizing new samples, although an unusual bridging of data between minority points and previously considered abnormal instances can be observed.

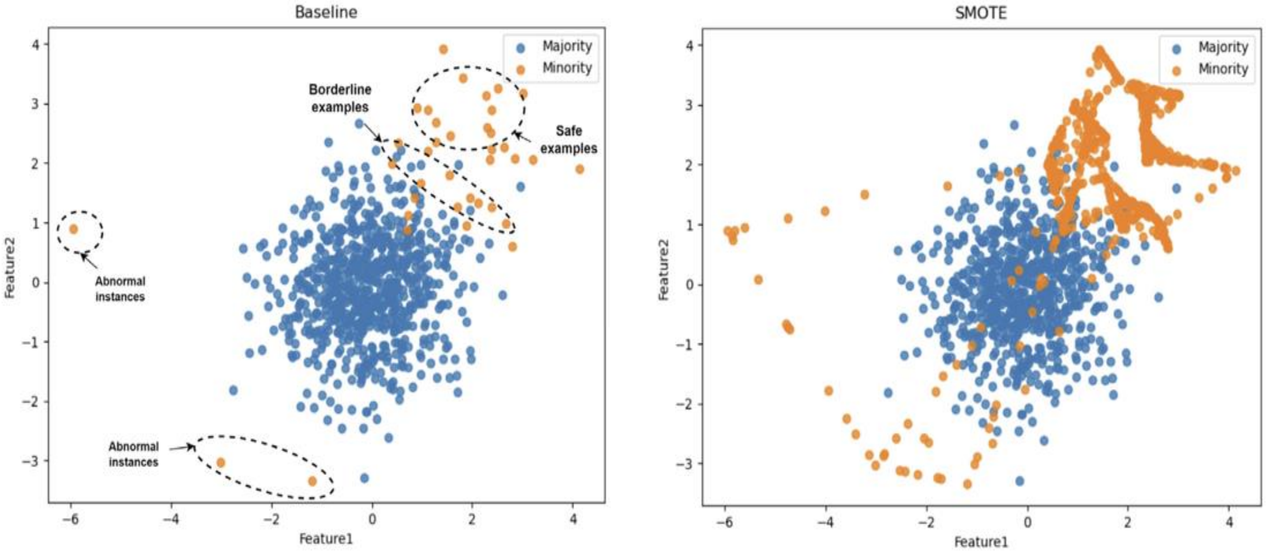

Fig. 2. Illustrates the distribution of data prior to (Baseline) and after applying SMOTE to address the issue of class imbalance in the dataset.

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

E. Optuna

The study included the adjustment of hyperparameters to assess how accurately the model generates results. It's part of the training process using both training and testing data to achieve an optimal level of accuracy [26]. However, many people tend to manually try various combinations of hyperparameters during model training and testing, making the process repetitive every few hours or even days. This method is not ideal for hyperparameter optimization, as it consumes considerable time and effort. This is where Optuna comes in as a solution [27].

Optuna is an optimization approach that provides three key benefits in model selection or hyperparameter tuning. The first benefit is the define-by-run API style, which offers flexibility in configuration. The second benefit is its effective pruning and sampling techniques. The third benefit is the simplicity of setup, making Optuna a convenient option for users. This define-by-run API is influenced by deep learning frameworks, enabling users to define hyperparameter search spaces in a flexible and dynamic way [28].

In this study, Optuna not only serves to optimize hyperparameters but also contributes to reducing the risk of overfitting by systematically exploring the best combinations of n_estimators and learning_rate in AdaBoost. Careful hyperparameter adjustment is important for preventing models that are too complex and chieve high accuracy on training data but struggle to generalize to unseen data [29].

Fig. 3 Describes the primary steps involved in selecting hyperparameters for machine learning models through Optuna.

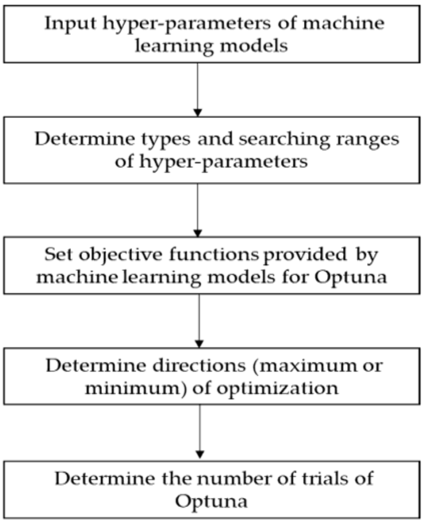

Fig. 3. The basic procedure for using Optuna in the systematic process of hyperparameter selection and optimization for machine learning models [28].

F. AdaBoost

AdaBoost (Adaptive Boosting), first introduced by Freund and Schapire, is a widely used ensemble algorithm that iteratively trains a series of weak classifiers on weighted data, then combines their outputs through a weighted summation to form a stronger final model [30].

The AdaBoost algorithm offers the advantage of combining multiple weak classifiers by assigning higher weights to those with better performance, allowing it to improve overall model accuracy. While it was not originally designed for imbalanced datasets, its ability to assign greater emphasis to underrepresented classes makes it a promising approach in such contexts [31]. The pseudocode for the AdaBoost algorithm is illustrated in Algorithm. 1.

Algorithm 1. Pseudocode of the AdaBoost algorithm illustrating the iterative process of weighting and combining weak learners to form a strong classification model [32].

```
Given: $(x_1, y_1), \dots, (x_m, y_m)$ where $\mathrm{x_i} \in X, y_i \in \{-1, +1\}$
Initialize: $D_1(i) = \frac{1}{m}$ for $i = 1, \dots, \mathrm{m}$
For $t = 1, \dots, T$:
    • Train weak learner using distribution $D_t$
    • Get weak hypothesis $h_t : X \rightarrow \{-1, +1\}$
    • Aim: Select $h_t$ to minimalize the weighted erorr:
          $\in_t = \boldsymbol{Pr}_{i \sim D_t}[h_t(x_i) \neq y_i]$
    • Choose:
          $\alpha_t = \frac{1}{2} \log\left(\frac{1 - \in_t}{\in_t}\right)$
    • Update, for $1 = 1, \dots, \mathrm{m}$:
          $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{\alpha_t} & \text{if } \mathrm{h}_t(\mathrm{x_i} = \mathrm{y_i}) \\ e^{\alpha_t} & \text{if } \mathrm{h}_t(\mathrm{x_i} \neq \mathrm{y_i}) \end{cases}$
          $= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
    Where $Z_t$ is a normalization factor (choosen so that $D_{t+1}$ will be a distribution).
    Output the final hypothesis:
          $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
```

G. Model Evaluation

In this study, the performance of the classification algorithm is assessed using a confusion matrix, a commonly used tool in machine learning to evaluate prediction quality across various classifiers such as neural networks, decision trees, and support vector machines and many more [33]. As stated by [34], True Positive (TP) refers to the number of documents from class 1 that are correctly classified as class 1. On the other hand, True Negative (TN) represents the number of documents from class 0 that are accurately categorized as class 0. False Positive (FP) refers to the number of documents from class 0 that are mistakenly classified as class 1. Finally, False Negative (FN) denotes the number of documents from class 1 that are misclassified as class 0. The confusion matrix used includes accuracy, recall, precision, and F1-Score.

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

These evaluation measures were chosen due to their ability to thoroughly assess classification outcomes, especially when dealing with datasets that exhibit class imbalance. Accuracy gives a general overview of performance, while precision and recall are essential to evaluate the model's effectiveness addressing both types of misclassification: false positives also false negatives. F1-Score integrates these two aspects, offering reliable insight when dealing with data imbalance.

Accuracy (ACC) represents the proportion of correctly predicted instances to the overall number of samples within the evaluation dataset. Its value ranges between 0 and 1, where a score of 1 means that both positive and negative instances are entirely classified correctly, whereas a score of 0 indicates that none of the instances have been correctly predicted [35]. The computation for accuracy is presented in Eq. (2) [36].

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} = \frac{\mathrm{TP} + \mathrm{TN}}{P + N} \tag{2}$$

Recall, also referred to as sensitivity or True Positive Rate (TPR), indicates the proportion of actual positive instances that are correctly identified by the model. It's computed by dividing the number of true positive predictions by the total number of actual positive cases. The value of recall ranges between [0, 1], where a value of 1 implies that all positive instances are accurately detected, and a value of 0 means that none of the positive instances were identified [35]. The formula for recall can be seen in Eq. (3) [36].

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} = \frac{TP}{P} \tag{3}$$

Precision measures the accuracy of positive predictions by evaluating the proportion of true positives among all instances predicted as positive. It is determined by dividing the number of correctly predicted positive instances by the total number of instances assigned to the positive class. The value of precision ranges from 0 to 1, where a score of 1 denotes perfect classification of the class, and a score of 0 signifies that none of the predicted samples were correct [35]. The computation for precision is presented in Eq. (4) [36].

$$\mathrm{PREC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

The F1-Score represents the harmonic mean between precision and recall, effectively discouraging imbalanced contributions from either metric. Since it is influenced by how the positive and negative classes are defined, its outcome can differ based on class labeling. The F1-Score ranges from 0 to 1, where a score of 1 indicates that both precision and recall are perfectly achieved, and score 0 reflects the complete absence of one or both metrics [35]. The computation for F1-Score is presented in Eq. (5) [36].

$$F1 - Score = \frac{2.precision.recall}{precision + recall} \tag{5}$$

H. Implementation Environment

All experimental stages in this study were carried out using Python version 3.11. The main libraries used included scikit-learn version 1.6.1 for machine learning processes as well as Optuna version 4.3.0 for hyperparameter optimization. The experimental procedures were carried out using a personal computer featuring an 11th Generation Intel® Core™ i5-11400H CPU running at 2.70GHz, supported by 16 GB of RAM, a 512 GB solid-state drive, and operating on the Windows 11 platform.

3. RESULTS

A. Data Distribution Analysis Before and After Resampling

The class distribution in the initial dataset shows a significant imbalance among the sleep disorder categories. As shown in Fig. 4, total sample size in class 0 and class 1 are each only 62, while class 2 has 175 samples. Such imbalance may negatively impact the model's ability to accurately identify minority classes.

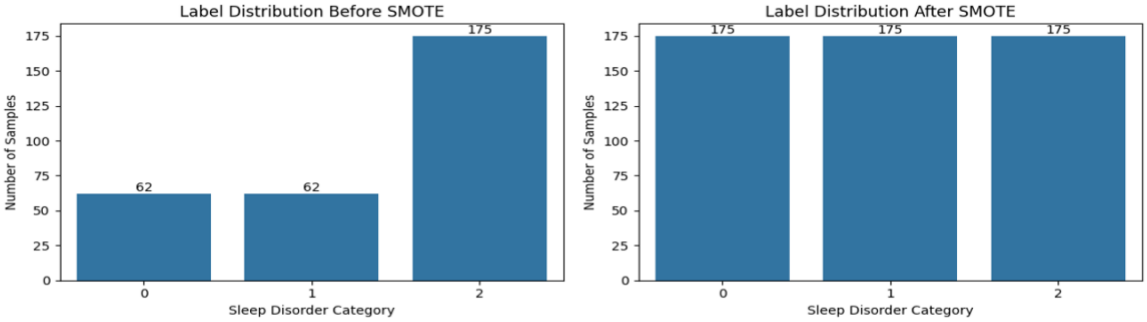

Fig. 4. Illustrates distribution of sample size for each sleep disorder category before and after implementing SMOTE.

To solve this problem, oversampling was performed using the Synthetic Minority Over-sampling Technique (SMOTE). After applying SMOTE, the distribution of the three classes became balanced, each having 175 samples. This change in distribution is clearly shown at the bottom of Fig. 4, which describes the increase in sample size for minimum class, making it equivalent to the majority class.

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

B. Optimal Hyperparameter Selection Using Optuna

In this study, the AdaBoost algorithm was used as the classification model. To boost performance in this model, a powerful combination of hyperparameters was searched for using the Optuna framework. The two main parameters optimized were many estimators (n_estimators) and learning_rate. Specifically, n_estimators was tuned using a categorical distribution with values [50, 100, 200], while learning_rate was optimized as a continuous float ranging from 0.01 to 1.0. The optimization process was performed over 20 trials using the Tree-structured Parzen Estimator (TPE) sampler with a fixed random seed for reproducibility.

Table 3 represents the best parameter exploration results obtained from the tuning process. Before applying SMOTE, the optimal combination was n_estimators = 100 and learning_rate ≈ 0.885. Meanwhile, after resampling using SMOTE, the optimal parameters changed to n_estimators = 200 and learning_rate ≈ 0.999. The difference in these values indicates that the resampling process influenced the characteristics of the optimal model.

Table 3. The results of the best hyperparameter selection using Optuna before and after applying SMOTE to the AdaBoost model.

Sampling Condition	n_estimators	learning_rate
Before SMOTE	100	0.8852
After SMOTE	200	0.9985

C. Model Performance Evaluation

There are four measurements of model performance evaluation: accuracy, precision, recall, and F1-score, for each one of three target classes: class 0 (insomnia), class 1 (sleep apnea), and class 2 (none). The evaluation was performed under four different scenarios to observe the impact of applying SMOTE and parameter tuning on the classification performance of the model.

Table 4 presents results without using SMOTE and without tuning with Optuna. In this condition, the model obtained an accuracy of 0.880, with the best performance demonstrated by class 2 (none), which had an F1-score of 0.9655. In contrast, class 1 (sleep apnea) recorded the lowest F1-score of 0.7500.

Table 4. Model performance evaluation results without SMOTE and without tuning using classification metric.

Class	Precision	Recall	F1-Score
0	0.75000	0.80000	0.77419
1	0.75000	0.75000	0.75000
2	0.97674	0.95455	0.96552
Accuracy			0.88000

After applying SMOTE without parameter tuning, as shown in Table 5, the accuracy increased to 0.893. The F1 scores for class 0 (insomnia) and class 1 (sleep apnea) improved, indicating that the resampling technique effectively improved the model's capacity to identify minority classes.

Table 5. Evaluation results of model performance after applying SMOTE without parameter tuning.

Class	Precision	Recall	F1-Score
0	0.80000	0.80000	0.80000
1	0.76471	0.81250	0.78788
2	0.97674	0.95455	0.96552
Accuracy			0.89333

Table 6 illustrates the model evaluation results after parameter tuning using Optuna, but without SMOTE. The accuracy significantly increased to 0.9067. Precision and recall for class 0 (insomnia) and class 1 (sleep apnea) also became more balanced (0.8125), indicating that parameter tuning improved the stability of classification performance across classes.

Table 6. Model performance evaluation after parameter tuning without using SMOTE.

Class	Precision	Recall	F1-Score
0	0.81250	0.86667	0.83871
1	0.81250	0.81250	0.81250
2	0.97674	0.95455	0.96552
Accuracy			0.90667

Finally, Table 7 shows the model performance results after the combination of SMOTE and tuning was applied. The results are consistent with the previous scenarios, with accuracy remaining at 0.9067 and the F1-score for the minority class remaining high. This combination demonstrates the most stable and optimal results across all classes, including insomnia, sleep apnea, and none.

Table 7. Evaluation of model performance with the combination of SMOTE and Optuna parameter tuning.

Class	Precision	Recall	F1-Score
0	0.81250	0.86667	0.83871
1	0.81250	0.81250	0.81250
2	0.97674	0.95455	0.96552
Accuracy			0.90667

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

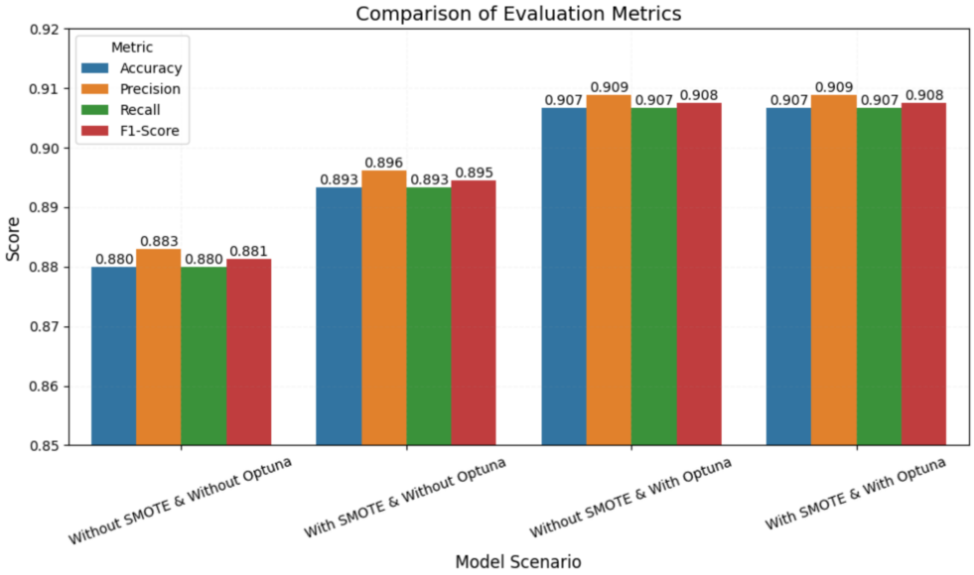

Fig. 5. Comparison of accuracy, precision, recall, and F1-Score in four models scenarios.

Fig. 5 presents a comparison of the comprehensive performance metrics of the model in four different layouts: without SMOTE and Optuna, with SMOTE but without Optuna, without SMOTE but with Optuna, and the combination of along SMOTE and Optuna. The metrics compared include accuracy, precision, recall, and F1-score.

It is observable that scenarios involving the application of Optuna, whether with or without SMOTE, resulted in significant improvements across all evaluation metrics. The scenario without SMOTE and without tuning had the lowest performance, while the last two scenarios—with Optuna tuning, whether with or without SMOTE—showed the highest and most stable metric values (around 0.907–0.909 across all metrics).

Overall, the results in Fig. 6 indicate that parameter tuning using Optuna contributes more significantly to improving model performance compared to applying SMOTE alone. However, the combination of both still yields highly optimal results, making it the recommended configuration.

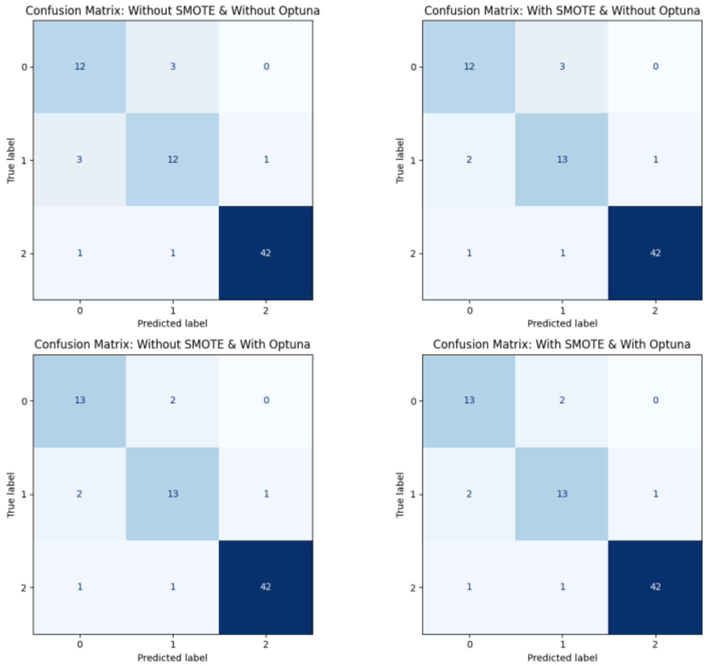

Fig. 6. The confusion matrix visualization across the four experimental setups of the predictive classification framework

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.

DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

Fig. 6 illustrates the confusion matrix visualization across the four experimental setups of the predictive classification framework, which are: (1) without SMOTE and without Optuna tuning, (2) with SMOTE but without tuning, (3) without SMOTE but with Optuna tuning, and (4) with SMOTE and tuning using Optuna. In all four scenarios, class 2, which represents the category of no sleep disorder (none), was consistently classified very well, as seen from the 42 correct predictions across all configurations. On the other hand, performance improvements for class 0 (insomnia) and class 1 (sleep apnea) were more noticeable after tuning with Optuna. This is reflected in the reduction of prediction errors in classes 0 and 1, from three errors each to two errors after tuning. Overall, it is possible to conclude that the application of Optuna as a parameter optimization method plays a significant contribution to enhancing model performance, especially in identifying more complex types of sleep disturbances like insomnia and obstructive sleep apnea.

4. DISCUSSION

On the basis of available information, implementation of SMOTE as well as parameter tuning with Optuna has already been proven to significantly improve classification model performance. Without SMOTE and tuning (Table 4), accuracy reached only 0.88000, with F1-Scores for both class 0 and class 1 of 0.77419 and 0.75000, resp. After applying SMOTE (Table 5), there was an increase in the F1-Scores for class 0 to 0.80000 and class 1 to 0.78788, with accuracy rising to 0.89333. Better results were obtained with Optuna tuning without SMOTE (Table 6), where accuracy reached 0.90667 and the F1-Score for the minority class improved significantly. The combination of SMOTE and Optuna (Table 7) provided the most stable performance across all classes. This suggests that using a mix of rebalancing as well as tuning techniques can increase accuracy while reducing bias towards the majority class.

Table 8. A comparison of this study with several previous studies in terms of methods, accuracy, as well as the advantages and disadvantages of the approaches used.

No	Researcher	Method	Accuracy Result
1	[37]	SVM & Neural Network	SVM: 90,1% Neural Network: 91,2%
2	[40]	Naïve Bayes (Bernoulli & Complement)	Bernoulli NB: 76% → 73% (SMOTE) Complement NB: 79% → 81% (NearMiss)
3	[38]	Random Forest	Random Forest: 88%
4	[39]	Random Forest & KNN	Random Forest: 89,69% KNN: 87,02%
5	[41]	Naïve Bayes Gaussian	Naïve Bayes Gaussian: 85,3%
6	This Study	AdaBoost, SMOTE & Optuna	AdaBoost with SMOTE & Optuna: 90,6%

As shown in Table 8, this study demonstrates optimal model performances with 90.6% accuracy using the AdaBoost algorithm combined with the SMOTE technique and Optuna parameter tuning. These results are in close competition with the study in [37] which applied Neural Networks and achieved an accuracy of 91.2%; however, the study did not use resampling techniques or parameter optimization, so the potential of the model has not been fully maximized. Meanwhile, the studies in [38] and [39] using Random Forest achieved accuracies of only 88% and 89.69%, respectively, without applying balancing approaches or further optimization. This confirms that the combination of SMOTE and Optuna in this study significantly contributes to performance improvement.

The study by [40] shows a decrease in performance for the Bernoulli Naive Bayes algorithm after applying SMOTE, indicating that not all algorithms react positively to resampling techniques. Instead, Complement Naive Bayes approach merged for NearMiss achieves up to 81% accuracy improvement. [41] using Gaussian Naive Bayes, recorded an accuracy of 85.3%, but limitations in preprocessing became a barrier to improving model performance. This highlights that the success of an approach depends not only on the algorithm but also on the proper data balancing and parameter tuning process, as implemented in this research.

Thus, the approach used in this paper proved to make an effective improvement in classification accuracy while providing a balanced performance across all classes, including minority classes, which are often overlooked in previous research.

There are a number of constraints in this research that should be taken into account. One, it was a relatively small dataset, with only 374 data points, which may affect the model's generalizability to new data. Second, this study uses only one type of algorithm, AdaBoost, so it cannot be directly compared with the performance of other models in the same context. Additionally, the dataset used is synthetic data, not real clinical data, which may have different characteristics when applied to real-world data.

From a theoretical perspective, this paper makes a contribution towards the advancement for machine learning in the medical field, particularly in classification of sleep disorders utilizing AdaBoost. SMOTE and Optuna combination approach applied in this study successfully improved model performance and provides academic insights into strategies for handling data imbalance and parameter optimization. Practically, the model generated

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.

DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

can assist medical professionals in enhancing the accuracy of sleep disorder diagnoses, accelerating the detection process, and providing a real solution to the handling of imbalanced medical data. Thus, this study supports the implementation of more optimal artificial intelligence systems within the healthcare industry.

5. CONCLUSION

The objective of this study was to analyze how well the AdaBoost algorithm performs in classification sleep disorders, specifically insomnia, sleep apnea, and none, by integrating the SMOTE data balancing technique and hyperparameter optimization through Optuna. The experimental results show that this combination of techniques consistently improves model performance, with the highest accuracy of 90.6% and significant improvements in F1-Score metrics for the minority classes, namely 0.83871 (insomnia) and 0.81250 (sleep apnea).

The dataset's size is the primary constraint of this study (374 synthetic data points) and the use of a single AdaBoost model. Therefore, further research is recommended to use real-world datasets on a larger scale and to explore the application of the SMOTE and Optuna combination in other machine learning algorithms so that the classification outcomes can be more broadly applicable and relevant in real-world clinical environments.

REFERENCES

[1] X. Qi et al., “Associations between flavonoids intake and sleep disorders: A survey-weighted cross-section study from NHANES,” J. Nutr. Biochem., hal. 109944, 2025, doi: 10.1016/j.jnutbio.2025.109944.

[2] K. Kostev, N. Kaur, C. Vetter, dan M. Konrad, “Sleep disorders are associated with subsequent depression and anxiety disorders in children and adolescents in Germany- a retrospective cohort study,” J. Psychiatr. Res., 2025, doi: 10.1016/j.jpsychires.2025.05.030.

[3] K. Qi et al., “A study on sleep quality under the shadow of school bullying: The interwoven effects of depressed mood, low self-esteem, and negative parenting practices,” Acta Psychol. (Amst)., vol. 253, no. December 2024, hal. 104717, 2025, doi: 10.1016/j.actpsy.2025.104717.

[4] J. Karimi dan B. M. Asl, “Automatic detection of non-apneic sleep arousal regions from polysomnographic recordings,” Biomed. Signal Process. Control, vol. 66, no. December 2020, hal. 102394, 2021, doi: 10.1016/j.bspc.2020.102394.

[5] S. D. Wickramaratne et al., “Physiology Guided Machine Learning Approach to Predict Short and Long Term Outcomes of Obstructive Sleep Apnea,” 2024, doi: https://doi.org/10.1101/2024.11.20.24317571.

[6] S. Sharma dan D. A. Singhal, “A Model based on Borderline 2 Synthetic Minority Oversampling Technique and Extreme Gradient Boosting for predicting Cardiovascular Disease,” Soc. Sci. Res. Netw., hal. 14, 2024, doi: https://doi.org/https://dx.doi.org/10.2139/ssrn.4994847.

[7] K. . Navin, K. H. Nehemiah, N. Jane, dan V. Saroji, “A classification framework using filter–wrapper based feature selection approach for the diagnosis of congenital heart failure,” J. Intell. Fuzzy Syst., vol. 44, no. 4, hal. 1–36, 2023, doi: 10.3233/JIFS-221348.

[8] M. Faheem dan M. A. Al-Khasawneh, “Multilayer cyberattacks identification and classification using machine learning in internet of blockchain (IoBC)-based energy networks,” Data Br., vol. 54, 2024, doi: 10.1016/j.dib.2024.110461.

[9] M. Mostafa Monowar et al., “Advanced sleep disorder detection using multi-layered ensemble learning and advanced data balancing techniques,” Front. Artif. Intell., vol. 7, 2024, doi: 10.3389/frai.2024.1506770.

[10] S. R. Begum, S. Y. Sait, dan A. S. Ramachandran, “AI-based Depression Detection using Profile Information,” 2nd Int. Conf. Autom. Comput. Renew. Syst. ICACRS 2023 - Proc., no. December, hal. 1367–1373, 2023, doi: 10.1109/ICACRS58579.2023.10405040.

[11] R. Chahar, A. K. Dubey, dan S. K. Narang, “Multiclass Classification of Mental Health Disorders Using XGBoost-HOA Algorithm,” SN Comput. Sci., vol. 5, no. 1167, 2024, doi: https://doi.org/10.1007/s42979-024-03525-6.

[12] J. Jiménez-García et al., “Assessment of airflow and oximetry signals to detect pediatric sleep Apnea-Hypopnea syndrome using AdaBoost,” Entropy, vol. 22, no. 6, hal. 670, 2020, doi: 10.3390/E22060670.

[13] R. S. Lalawat, V. Bajaj, dan P. K. padhy, “Optimal variational mode decomposition based automatic stress classification system using EEG signals,” Appl. Acoust., vol. 231, no. October 2024, hal. 110478, 2025, doi: 10.1016/j.apacoust.2024.110478.

[14] S. Dhanka dan S. Maini, “A hybridization of XGBoost machine learning model by Optuna hyperparameter tuning suite for cardiovascular disease classification with significant effect of outliers and heterogeneous training datasets,” Int. J. Cardiol., vol. 420, hal. 132757, 2025, doi: 10.1016/j.ijcard.2024.132757.

[15] D. Darwis, N. Siskawati, dan Z. Abidin, “Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional,” J. Tekno Kompak, vol. 15, no. 1, hal. 131, 2021, doi: 10.33365/jtk.v15i1.744.

[16] M. X. Low et al., “Comparison of Label Encoding and Evidence Counting for Malware Classification,”

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.

DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

J. Syst. Manag. Sci., vol. 12, no. 6, hal. 17–30, 2022, doi: 10.33168/JSMS.2022.0602.

[17] M. Ahmed Ouameur, M. Caza-Szoka, dan D. Massicotte, “Machine learning enabled tools and methods for indoor localization using low power wireless network,” Internet of Things (Netherlands), vol. 12, hal. 100300, 2020, doi: 10.1016/j.iot.2020.100300.

[18] L. B. V. de Amorim, G. D. C. Cavalcanti, dan R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” Appl. Soft Comput., vol. 133, hal. 1–37, 2023, doi: 10.1016/j.asoc.2022.109924.

[19] B. Deepa dan K. Ramesh, “Epileptic seizure detection using deep learning through min max scaler normalization,” Int. J. Health Sci. (Qassim)., vol. 6, no. May, hal. 10981–10996, 2022, doi: 10.53730/ijhs.v6ns1.7801.

[20] D. E. Birba, “A Comparative study of data splitting algorithms for machine learning model selection,” Degree Proj. Comput. Sci. Eng., vol. 2020, no. 1, hal. 1–23, 2020, [Daring]. Tersedia pada: https://www.diva-portal.org/smash/get/diva2:1506870/FULLTEXT01.pdf

[21] Scikit-learn Developers, “train_test_split — dokumentasi scikit-learn 1.6.1,” scikit-learn. [Daring]. Tersedia pada: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[22] D. S. Turan dan B. Ordin, “The incremental SMOTE: A new approach based on the incremental k-means algorithm for solving imbalanced data set problem,” Inf. Sci. (Ny)., vol. 711, no. March, hal. 122103, 2025, doi: 10.1016/j.ins.2025.122103.

[23] D. Elreedy dan A. F. Atiya, “A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance,” Inf. Sci. (Ny)., vol. 505, hal. 32–64, 2019, doi: 10.1016/j.ins.2019.07.070.

[24] B. H. Aubaidan, R. A. Kadir, dan M. T. Ijab, “A Comparative Analysis of Smote and CSSF Techniques for Diabetes Classification Using Imbalanced Data,” J. Comput. Sci., vol. 20, no. 9, hal. 1146–1165, 2024, doi: 10.3844/JCSSP.2024.1146.1165.

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

[26] D. A. Dinathi, E. Ramadanti, dan D. R. Chandranegara, “Diabetes Detection Using Extreme Gradient Boosting (XGBoost) with Hyperparameter Tuning,” Indones. J. Electron. Electromed. Eng. Med. Informatics, vol. 6, no. 2, hal. 78–84, 2024.

[27] P. Srinivas dan R. Katarya, “hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost,” Biomed. Signal Process. Control, vol. 73, no. December 2021, hal. 103456, 2022, doi: 10.1016/j.bspc.2021.103456.

[28] J. P. Lai, Y. L. Lin, H. C. Lin, C. Y. Shih, Y. P. Wang, dan P. F. Pai, “Tree-Based Machine Learning Models with Optuna in Predicting Impedance Values for Circuit Analysis,” Micromachines, vol. 14, no. 2, 2023, doi: 10.3390/mi14020265.

[29] T. Akiba, S. Sano, T. Yanase, T. Ohta, dan M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” in KDD ’19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, hal. 2623–2631. doi: 10.1145/3292500.3330701.

[30] W. Wang dan D. Sun, “The improved AdaBoost algorithms for imbalanced data classification,” Inf. Sci. (Ny)., vol. 563, hal. 358–374, 2021, doi: 10.1016/j.ins.2021.03.042.

[31] A. de Giorgio, G. Cola, dan L. Wang, “Systematic review of class imbalance problems in manufacturing,” J. Manuf. Syst., vol. 71, no. September, hal. 620–644, 2023, doi: 10.1016/j.jmsy.2023.10.014.

[32] R. D. Mendrofa, M. H. Siallagan, J. Amalia, dan D. P. Pakpahan, “Credit Risk Analysis With Extreme Gradient Boosting and Adaptive Boosting Algorithm,” J. Inf. Syst. Hosp. Technol., vol. 5, no. 1, hal. 1–7, 2023, doi: 10.37823/insight.v5i1.233.

[33] I. Düntsch dan G. Gediga, “Indices for rough set approximation and the application to confusion matrices,” Int. J. Approx. Reason., vol. 118, hal. 155–172, 2020, doi: 10.1016/j.ijar.2019.12.008.

[34] D. Normawati dan S. A. Prayogi, “Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter,” J. Sains Komput. Inform. (J-SAKTI, vol. 5, no. 2, hal. 697–711, 2021.

[35] S. A. Hicks et al., “On evaluation metrics for medical applications of artificial intelligence,” Sci. Rep., vol. 12, no. 1, hal. 1–9, 2022, doi: 10.1038/s41598-022-09954-8.

[36] Ž. Vujović, “Classification Model Evaluation Metrics,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 6, hal. 599–606, 2021, doi: 10.14569/IJACSA.2021.0120670.

[37] D. Sari, “Prediksi Gangguan Tidur pada Sleep Health and Lifestyle Menggunakan Support Vector Machine dan Neural Network,” Jav. J. Vokasi Inform., hal. 36–42, 2024, doi: 10.24036/javit.v4i1.168.

[38] I. A. Hidayat, “Classification of Sleep Disorders Using Random Forest on Sleep Health and Lifestyle Dataset,” J. Dinda Data Sci. Inf. Technol. Data Anal., vol. 3, no. 2, hal. 71–76, 2023, doi:

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.

DOI: https://doi.org/10.35882/ijeeemi.v6i3.407

10.20895/dinda.v3i2.1215.

[39] N. Khasanah, D. Uki, E. Saputri, F. Aziz, dan T. Hidayat, “Studi Perbandingan Algoritma Random Forest dan K-Nearest Neighbors (KNN) dalam Klasifikasi Gangguan Tidur,” Comput. Sci., vol. 5, no. 1, hal. 17–25, 2025, doi: https://doi.org/10.31294/coscience.v5i1.5522.

[40] M. R. Fatturrahman dan A. Kurniasih, “Penggunaan Metode NearMiss, SMOTE, dan Naïve Bayes untuk Klasifikasi Gangguan Tidur Berdasarkan Kualitas Tidur dan Gaya Hidup,” Pros. Semin. Nas. Mhs. Bid. Ilmu Komput. dan Apl., vol. 4, no. 2, hal. 567–576, 2023.

[41] G. D. Rahayuningtyas, “Prediksi Sleep Disorder Menggunakan Metode Algoritma Naïve Bayes Gaussian,” J. Tek. Inform. Sains dan Ilmu Komun., vol. 3, hal. 283–291, 2025, doi: https://doi.org/10.59841/saber.v3i1.2375.

AUTHOR BIOGRAPHY

Muhammad Naufal Anshory is an active student in the Computer Science Study Program, Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University, since 2021. He was born in Banjarmasin, and has a particular interest in the field of data science. His study focus includes data processing techniques, machine learning, and their applications in the medical and social fields. Although he has no formal research experience, Naufal has shown a high interest in developing data-driven solutions and aspires to build a professional career in a field that aligns with his interests and skills. Outside of academic activities, he is actively involved in non-academic activities that support the development of soft skills and leadership. He can be contacted via email: muhammad.naufal210@gmail.com.

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

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

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

Setyo Wahyu Saputro, is a lecturer in Computer Science Department, Faculty of Mathematics and Natural Science, Lambung Mangkurat University in Banjarbaru. He received bachelor’s degree also in Computer Science from Lambung Mangkurat University in 2011, and received his master’s degree in Informatics from STMIK Amikom University in 2016. He is active as an information technology practitioner and consultant, being a project manager or systems analyst working on several projects in government and private agencies in South Kalimantan province since 2017. His research interests include software engineering, human computer interaction, and artificial intelligence applications. He can be contacted at email: setyo.saputro@mhs.ulm.ac.id. Orcid ID: 0009-0007-9250-7704.

Corresponding author: Muhammad Itqan Mazdadi, mazdadi@ulm.ac.id, Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Kalimantan Selatan, Indonesia.
DOI: https://doi.org/10.35882/ijeeemi.v6i3.407