The Effect of SMOTE and Optuna Hyperparameter Optimization on TabNet Performance for Heart Disease Classification

Danang Wijayanto[1]*, Robert Marco[2], Acihmah Sidauruk[3], Mulia Sulistiyono[4]
Department of Computer Science [1], [2], [3],[4]
University of AMIKOM
Yogyakarta, Indonesia
danangwijayanto507@gmail.com[1], robertmarco@amikom.ac.id[2], acihmah@amikom.ac.id[3] ,
muliasulistiyono@amikom.ac.id[4]

***Abstract*— Heart disease is a medical condition affecting the cardiovascular system, disrupting blood circulation and reducing cardiac function efficiency, which can lead to severe health complications. Early diagnosis of heart disease has become increasingly crucial as delayed detection can significantly impact patient outcomes and survival rates. While numerous studies have explored various approaches for heart disease classification, challenges related to data imbalance and improper parameter settings remain persistent issues that affect model performance. This research evaluated the effectiveness of combining TabNet with SMOTE and optuna hyperparameter optimization for heart disease classification. We conducted four experimental scenarios using a heart disease dataset with 303 instances: baseline TabNet, baseline TabNet with SMOTE, TabNet with Optuna, and TabNet with both SMOTE and Optuna. Results demonstrated that applying SMOTE alone to TabNet decreased model performance (accuracy from 85.24% to 77.04%, AUC from 0.89 to 0.83). However, when combining SMOTE with Optuna hyperparameter optimization, we achieved optimal performance with 90.16% accuracy, 93.33% precision, 87.50% recall, 90.32% F1-score, and 0.93 AUC. This represented a significant improvement over other configurations and several previous classification approaches. The integration of SMOTE with Optuna optimization provided an effective framework for heart disease classification that outperformed traditional methods particularly in discriminative capability as evidenced by the superior AUC score.**

Keywords*— TabNet, SMOTE, Optuna, Classification, Heart Disease

I. INTRODUCTION

Global cardiovascular disease (CVD) data from 2015 recorded 422.7 million cases and 17.92 million deaths. Ischemic heart disease emerged as the primary cause of CVD-related health loss worldwide, with stroke being the second most common cause. While high-income and some middle-income countries showed declining age-standardized CVD death rates between 1990-2015, mortality patterns shifted from women to men in regions with higher social development indices [1].

Within the context of heart disease research, Artificial Intelligence technology made substantial advances across multiple domains, especially in healthcare applications. As a specialized field within Artificial Intelligence, machine learning showed remarkable utility in various health-related cases, particularly in heart disease classification systems. Machine learning encompassed the development of computer systems that could autonomously improve their capabilities through accumulated experience [2].

Numerous previous studies extensively explored heart disease classification utilizing machine learning and deep learning methodologies. Nevertheless, challenges pertaining to data imbalance persisted, and the performance of machine learning models, especially deep learning architectures, depended heavily on appropriate hyperparameter configurations, which were difficult to determine manually [3].

Research conducted by Yogianto et al. [4] demonstrated the implementation of the K-Nearest Neighbors (KNN) algorithm in heart disease classification, yielding an accuracy rate of 64.03%. Masruriyah et al. [5] employed the SMOTE technique to address class imbalance issues and conducted a comparative analysis of multiple algorithms, producing varying classification accuracies: C4.5 achieved 70%, Random Forest 87%, K-Nearest Neighbors 86%, and Logistic Regression 73%.

The TabNet architecture, introduced by Arik and Pfister [6], offered several theoretical advantages for heart disease classification that addressed the limitations of previous approaches. Unlike conventional neural networks that processed all features simultaneously, TabNet employed sequential attention mechanisms that systematically identified and prioritized significant features throughout each decision-making phase. This approach was particularly suited to medical diagnostics, where certain features carried varying importance for different patient profiles.

To address the identified research gaps, this study proposed a comprehensive approach that combined TabNet with SMOTE for handling class imbalance and Optuna for hyperparameter optimization. This integration specifically targeted the dual challenges that limited previous heart disease classification models: data imbalance and suboptimal parameter selection

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

[7],[8],[9]. By systematically evaluating different combinations of these techniques, we aimed to determine their individual and combined effects on classification performance.

II. METHODOLOGY

In this study, we proposed a heart disease classification methodology using TabNet model, as illustrated in Figure 1. Our approach aimed to evaluate different combinatios of techniques using TabNet model, which offered excellent interpretability capabilities and efficiently handled tabular data.

The methodology followed a systematic workflow comprising four main stages: heart disease dataset, pre-processing, model implementation with experimental scenarios, and evaluation. We evaluated our approach using multiple performance metrics including accuracy, precision, recall, F1-score, and area under the ROC curve, and compared the results across different experimental scenarios to determine the individual and combined effects of SMOTE and Optuna Optimization.

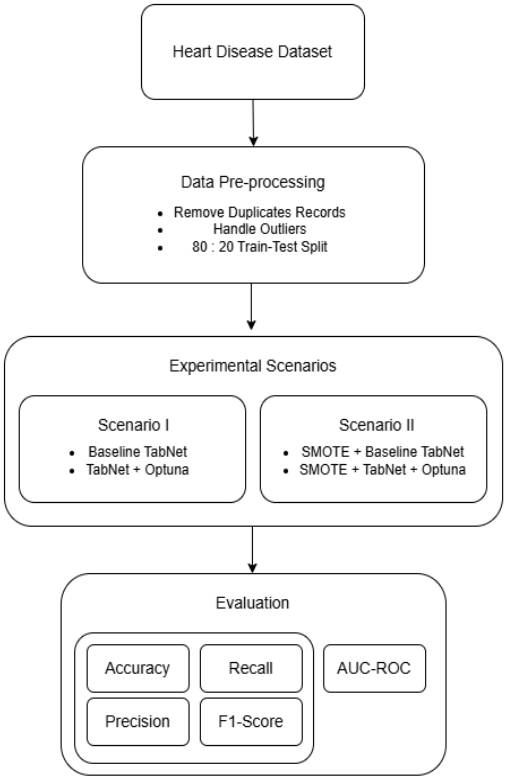

Figure 1. Methodology Research

A. Heart Disease Dataset

The dataset used in this research was obtained from a public dataset avalilable on Kaggle.com (https://www.kaggle.com/datasets/yasserh/heart-disease-dataset) which is derived from the UCI Heart Disease dataset, a widely used benchmark in medical classification research. The dataset contained a total of 303 data points comprising 13 features and 1 target variable. The target variable contained two values: 1 indicating the presence of heart disease and 0 indicating normal condition. The detailed description of each feature and data type is presented in Table I.

TABLE I. DATASET DESCRIPTION

No	Feature Name	Description	Data Type
1	Age	Patient's age in years	Numeric
2	Sex	Gender of patient (1:male; 2:Female)	Categorial
3	Cp	Type of chest pain experienced (0: asymptomatic, 1: atypical angina, 2 : non-anginal pain, 3: typical angina	Numeric
4	Tresbps	Resting blood pressure in mmHg	Numeric
5	Chol	Serum cholesterol level in mg/dl	Numeric
6	Fbs	Fasting blood sugar > 120 mg/dl (1:true, 0:false)	Categorial
7	Restecg	Resting electrocardiogram	Numeric
8	Thalach	Maximum heart rate achieved	Numeric
9	Exang	Exercise-induced angina (1:yes, 0:no)	Categorial
10	Oldpeak	ST depression induced by exercise relative to rest	Numeric
11	Slope	Slope of peak exercise ST segment	Numeric
12	Ca	Number of major vessels colored by fluoroscopy	Numeric
13	Thal	Thalassemia type	Numeric
14	Target	Heart disease diagnosis (1: heart disease, 0:normal)	Categorial

B. Pre-Processing

The preprocessing stage consisted of several steps to ensure the quality of data used in this research. Handling duplicate data aimed to identify and manage duplicated data entries. This was crucial for improving data quality before use and reducing false positives in the results [10].

Handling outliers focused on detecting and managing values that fall significantly outside the dataset's normal range. These anomalous values can substantially bias statistical calculations, particularly affecting mean values through under or overestimation. Thus, addressing outliers through modification or value substitution was essential before conducting data analysis [11]. We split the dataset into training and testing sets with an 80:20 ratio, which was a standard practice widely adopted in previous heart disease classification studies [7], [9], [19].

C. Experimental Scenarios

This research divided the experiments into two main scenarios to obtain more comprehensive evaluation results and enable more detailed comparative analysis. The details of both experimental scenarios are presented in Table II.

TABLE II. SCENARIOS

Scenario	Method
I	TabNet
	TabNet + Optuna
II	SMOTE + TabNet
	SMOTE + TabNet + Optuna

In both scenarios, we implemented two variants of TabNet: a baseline TabNet model and an optimized TabNet model using

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

Optuna for hyperparameter tuning. The first scenario used standard data splitting, while the second scenario incorporated SMOTE to address class imbalance issues. Sampling techniques such as SMOTE were only applied to the training dataset, not to validation or test sets, ensuring model evaluation occurred on data distributions that truly represented the actual problem domain, thus avoiding bias in performance assessment [12].

D. TabNet

TabNet is a deep learning algorithm specifically designed to process tabular data by combining sequential attention and neural networks concepts. TabNet employed sequential attention to select feature subsets, enabling efficient learning of the most prominent features, and its architecture consisted of sequential multi-step processing, where each step contributed to the decision based on selected features [6].

According to paper [6], TabNet architecture consisted of several main components: feature transformer that converted input features into more meaningful representations, attentive transformer that determined feature masks for each decision step, and feature masking that implemented sparse feature selection. At each decision step, TabNet used a learnable mask to select the most important features with the formula:

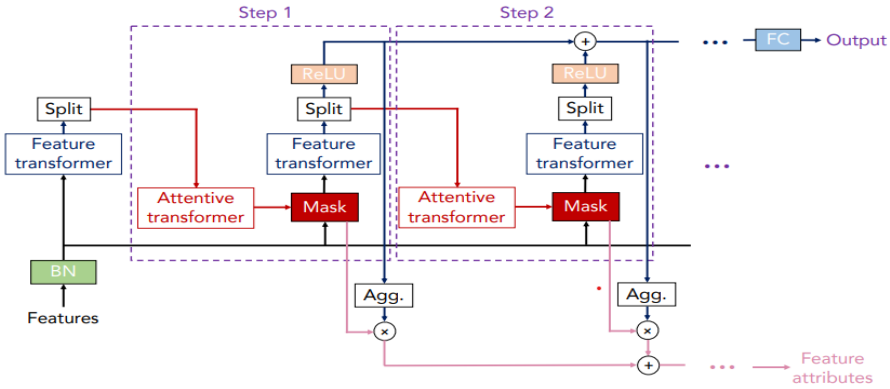

Figure 2. TabNet Architecture[6]

$$M[i] \cdot f \quad (1)$$

Where :
M[i] = Mask for step i
f = Input features

This mask was obtained using an attentive transformer with the formula:

$$M[i] = sparsemax(P[i-1] \cdot hi(a[i-1])) \quad (2)$$

Where:
M[i] = mask for step i
hi = trainable transformation function
a[i-1] = processed features from the previous step sparsemax = normalization that produces sparse weights

P[i] is the prior scale term indicating how much a feature has been previously used:

$$P[i] = \sum_{j=1}^{i} (\gamma - M[j]) \quad (3)$$

Where:
P[i] = prior scale at step i
γ = relaxation parameter (γ ≥ 1)
M[j] = mask from previous steps
P[0] = initialized as a 1B×D matrix

After features are selected, TabNet uses a feature transformer to process these features. The output of this process was divided into two parts:

$$[d[i], a[i]] = fi(M[i] \cdot f) \quad (4)$$

Where:
d[i] = output for the current decision step
a[i] = output information to be used in the next step
fi = the feature transformer function
M[i] = mask for step i
f = input feature

d[i] was the decision step output and a[i] was the information for the next step. To produce the final decision, TabNet aggregated the output from all decision steps using the formula:

$$dout = \sum_i ReLU(d[i]) \quad (5)$$

Where:
Dout = final decision output
ReLU = rectified Linear Unit activation function
d[i] = output from decision step i
Σi = summation over all steps

For interpretability, TabNet used an aggregate feature importance mask that was calculated by:

$$Magg-b,j = \sum_i \eta b[i] Mb,j[i] / normalization \quad (6)$$

Where:
Magg_b,j := Aggregated importance mask

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

nb[i] = Feature importance score at step i
Mb,j[i] = Mask for batch b and feature j at step i
Normalization= Normalization factor for value standardization

The contribution score ηb[i] was determined by:

$$\eta b[i] = \sum c \; ReLU(db, c[i]) \quad (7)$$

Where:
nb[i] = Feature importance score at step i
Σc = Summation over all classes
db,c[i] = Decision output for batch b and class c at step i
ReLU = Rectified Linear Unit activation function

E. Optuna

Optuna was a hyperparameter optimization framework developed by Akiba et al. in 2019. Optuna was designed with a "define-by-run" principle that allowed users to dynamically construct parameter search spaces, offering efficient implementation of search and pruning strategies, a flexible and versatile architecture for various purposes, and equipped with Tree-structured Parzen Estimators (TPE) in its optimization process which was useful for learning from previous optimization trials [13].

Through Optuna optimization, The parameter ranges were determined through preliminary experiments. We deliberately selected narrower, more focused ranges rather than broader exploration to maximize optimization efficiency given computational constraints are detailed in Table III.

TABLE III. TABNET OPTIMIZATION PARAMETERS

Parameter	Range Value
n_d	8 - 32
n_a	8 – 32
n_steps	5 – 8
n_independent	1 - 2
learning_rate	0.01 – 0.1
gamma	1.0 – 2.0
lambda_sparse	0.0001 – 0.01

F. Evaluation

In this study, the model evaluation was conducted using confusion matrix and AUC-ROC curve analysis. confusion matrix, as described by [14], was a fundamental evaluation tool in machine learning that displayed the relationship between predicted and actual classifications. It utilized a two-dimensional structure where one axis represented the true class labels while the other showed the model's predictions.

TABLE IV. CONFUSION MATRIX

		Actual	
		Positive	**Negative**
Prediction	**Positive**	TP	FP
	Negative	FN	TN

The structure of the binary classification confusion matrix implemented in this study consisted of four key components.

- True Positive (TP) : represented the number of correctly classified positive instances.
- True Negative (TN) : indicated the number of correctly classified negative instances.
- False Positive (FP) : also known as Type I error, represented negative instances incorrectly classified as positive
- False Negative (FN) : Type II error, indicated positive instances incorrectly classified as negative.

These components and their relationships are illustrated in Table IV. From the confusion matrix components, several key performance indicators can be calculated to evaluate the model's performance, including accuracy, precision, recall, and F1-score [15]. These evaluation metrics are calculated using the following formulas :

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

$$F1 - score = \frac{2*presisi*recall}{presisi+recall} \quad (11)$$

In addition to the confusion matrix, this study also utilized the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), a popular evaluation metric used to measure binary classification performance [16]. AUC-ROC was employed to analyze model performance in greater depth, particularly in identifying areas where the model struggled to separate positive and negative labels, which ultimately helped identify the classifier's decision boundary and potential AUC improvements.

III. RESULT AND DISCUSSION

In this study, we used a heart disease dataset containing 303 records with 13 features and 1 target variable. The features included patient characteristics and medical measurements such as age, sex, chest pain type, blood pressure, cholesterol, and other cardiac indicators, as shown in Figure 3. These features were selected based on their established clinical relevance to cardiac health assessment and diagnostic procedures in medical literature. The preprocessing phase began with duplicate detection, where one duplicate record was identified and removed, reducing the dataset to 302 records. This elimination of duplicates was an essential step to ensure the integrity of our analysis and prevent potential bias in model training and evaluation outcomes.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

Figure 3. Heart Disease Dataset

In the next step, we continued with outlier detection for the continuous features, which identified several outliers as shown in Figure 4. Considering the small dataset size, these outliers were replaced with their respective upper and lower bounds for each feature.

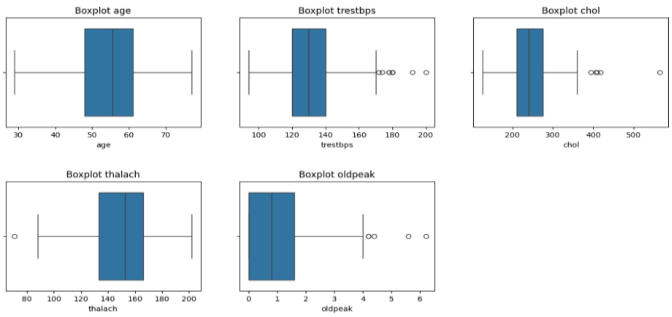

Figure 4. Outliers Data

After preprocessing, the dataset was duplicated for use in two scenarios. The data was split into training and testing sets with an 80:20 ratio. The data distribution was shown in Table V. For scenario II, SMOTE was applied to the training data to achieve balanced class distribution during model training to investigate potential performance improvements.

TABLE V. DISTRIBUTION OF HEART DISEASE DATASET

Scenario	Dataset	
	Training	Testing
I	241	61
II	264	61

The evaluation of TabNet performance on heart disease classification was conducted through two scenarios. Each scenario examined two model variations: baseline TabNet and TabNet with Optuna hyperparameter optimization. For both variations, we established initial parameters including a patience value of 20 for early stopping when no learning improvement was observed, batch and virtual_batch sizes of 32 to accommodate the small dataset, and n_trials of 20 in Optuna for the number of optimization attempts.

Figure 5 presents the training loss curves for all four experimental configurations across both scenarios. Statistical analysis was conducted on the final 20 epochs, chosen because at this stage all models had surpassed their initial rapid learning phase and entered more stable convergence patterns. providing a more reliable representation of each model's final learning characteristics. This analysis revealed significant differences in convergence patterns among the models.

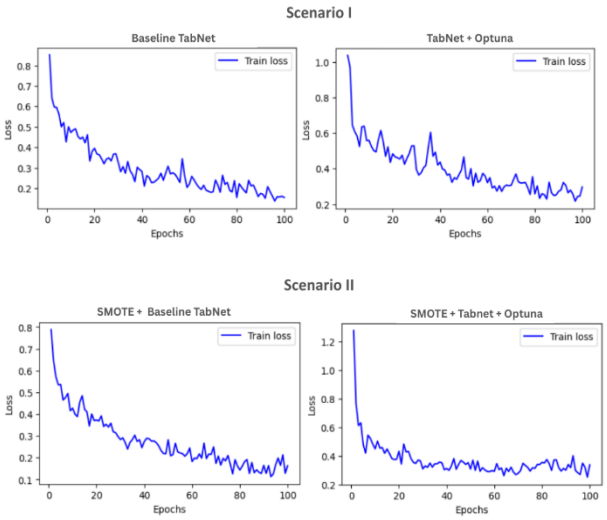

Figure. 5 Training Loss Over Epoch Scenario I and II

For Scenario I, the baseline TabNet demonstrated efficient early learning with a rapid decrease in training loss from 0.8 to 0.4 within the first 10 epochs, eventually reaching stable convergence around 0.2. When Optuna optimization was applied, the model showed higher initial loss (~1.0) but stabilized around 0.3 after epoch 60. Statistical comparison between these models revealed a significant difference in training behavior ($t = -9.74$, $p < 0.0001$), with baseline TabNet consistently maintaining lower loss values across the final 20 epochs. The variance analysis showed TabNet+Optuna exhibited slightly higher fluctuations ($\sigma^2 = 0.000863$) compared to baseline TabNet ($\sigma^2 = 0.000722$), supporting our observation that the optimized model explored a more diverse feature space.

For Scenario II with SMOTE application, the baseline TabNet achieved smoother convergence to approximately 0.15 by epoch 90. When SMOTE was combined with Optuna optimization, the model began with higher initial loss (~1.2) but stabilized between 0.3-0.4 after epoch 40. Statistical analysis of the final 20 epochs revealed an extremely significant difference between these models ($t = -15.83$, $p < 0.0001$), indicating SMOTE+TabNet consistently maintained lower training loss compared to SMOTE+TabNet+Optuna. The variance in SMOTE+TabNet+Optuna ($\sigma^2 = 0.001421$) was notably higher than SMOTE+TabNet ($\sigma^2 = 0.000757$), suggesting that Optuna optimization introduced beneficial regularization effects that prevented the model from minimizing training loss too aggressively.

Cross-scenario comparison revealed significant differences between baseline and SMOTE implementations ($t = 2.67$, $p = 0.015$), with SMOTE consistently resulting in lower training loss. Similarly, comparison between both Optuna-optimized models showed significant differences ($t = -5.93$, $p < 0.0001$), with SMOTE+TabNet+Optuna maintaining higher training loss. These statistical findings provided strong evidence that while SMOTE facilitated easier optimization of the loss function during training, Optuna's hyperparameter optimization introduced effective regularization effects that prevented overfitting to synthetic samples, explaining the superior test performance observed in subsequent evaluations despite higher

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

training loss.

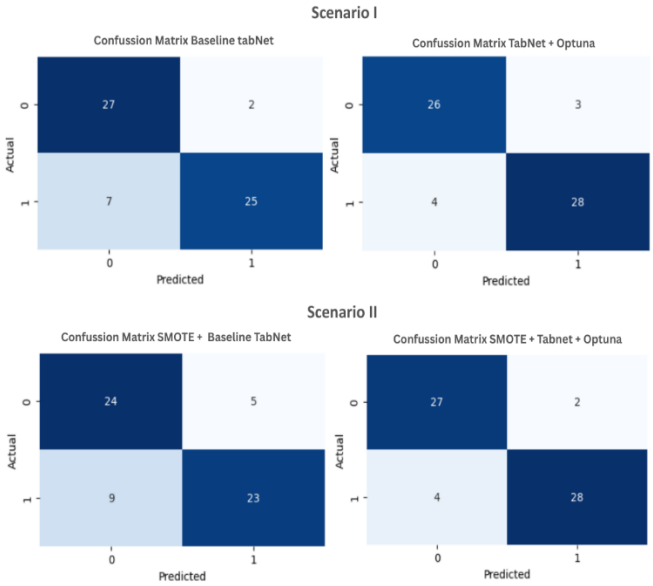

Figure 6. Confusion Matrix Scenario I and II

Figure 6 presents the confusion matrices for all four model configurations, providing a detailed view of classification performance. In Scenario I, the baseline TabNet correctly identified 27 negative cases (true negatives) and 25 positive cases (true positives), while misclassifying 2 negative cases as positive (false positives) and 7 positive cases as negative (false negatives). When Optuna optimization was applied to TabNet in Scenario I, the model demonstrated improved performance with 26 true negatives and 28 true positives. The optimization reduced misclassifications to 3 false positives and 4 false negatives.

In Scenario II, the application of SMOTE alone to the baseline TabNet unexpectedly decreased performance, with the model achieving 24 true negatives and 23 true positives, while showing increased misclassification rates with 5 false positives and 9 false negatives. This performance degradation could be attributed to several factors. First, the synthetic samples generated by SMOTE might have introduced noise in the feature space rather than meaningful patterns, given the relatively small original dataset size. Second, the TabNet's default parameters might not have been optimal for learning from the modified data distribution, causing the model to overfit to synthetic patterns that did not generalize well to the test set.

However, when SMOTE was combined with Optuna optimization, the model achieved the best overall performance with 27 true negatives and 28 true positives, while reducing misclassifications to 2 false positives and 4 false negatives. This demonstrated that while SMOTE alone might disrupt the original data distribution, Optuna's hyperparameter optimization effectively mitigated this issue by adapting the model architecture specifically to the characteristics of the balanced dataset.

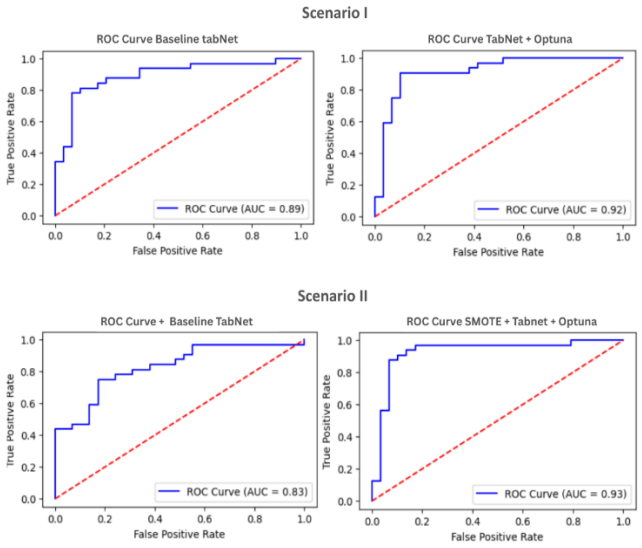

Figure 7. AUC-ROC for Scenario I and II

Figure 7 presents the ROC curves for all four model configurations, providing insights into their discriminative capabilities across different classification thresholds. In Scenario I, the baseline TabNet achieved an AUC score of 0.89, indicating strong overall classification ability. The Optuna-optimized version showed improved performance with an AUC of 0.92, demonstrated by a curve that rose more sharply at low false positive rates and maintained higher true positive rates throughout the threshold spectrum.

In Scenario II with SMOTE implementation, the baseline model showed a decreased performance with an AUC of 0.83, further confirming that class balancing alone negatively affected the model's discriminative ability. However, when combined with Optuna optimization, the model achieved the highest AUC of 0.93, characterized by a steep initial rise and consistently high true positive rates across different false positive rate thresholds.

TABLE VI. COMPARISON OF TABNET MODEL PERFORMANCE

Scenario	Method	Evaluation				
		Accuracy	Presisi	Recall	F1-score	AUC
I	TabNet	85.24%	92.59%	78.12%	84.74%	0,89
	TabNet + Optuna	88.52%	90.32%	87.50%	88.88%	0,92
II	SMOTE + TabNet	77.04%	82.14%	71.87%	76.66%	0,83
	SMOTE + tabnet + Optuna	90.16%	93.33%	87.50%	90.32%	0,93

The experimental results presented in Table VI show the evaluation metrics for all model variations across both scenarios. In Scenario I, the baseline TabNet achieved good performance with 85.24% accuracy, 92.59% precision, 78.12% recall, 84.74% F1-score, and 0.89 AUC. The Optuna-optimized

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

version showed improvement across all metrics, reaching 88.52% accuracy, 90.32% precision, 87.50% recall, 88.88% F1-score, and 0.92 AUC.

In Scenario II, the SMOTE-enhanced baseline TabNet initially showed decreased performance with 77.04% accuracy, 82.14% prerecision, 71.87% recall, 76.66% F1-score, and 0.83 AUC. However, when combined with Optuna optimization, the model achieved the best overall performance with 90.16% accuracy, 93.33% precision, 87.50% recall, 90.32% F1-score, and 0.93 AUC.

The results demonstrate that while SMOTE alone may reduce model performance, its combination with Optuna optimization leads to superior results across all evaluation metrics. The progression from baseline to optimized models in both scenarios highlights the significant impact of proper hyperparameter tuning, particularly when implementing class balancing techniques. A comparative visualization of these performance metrics across all model configurations can be seen in Figure 8. The optimal parameter configurations determined by Optuna that led to these improvements for both scenarios are presented in Table VII.

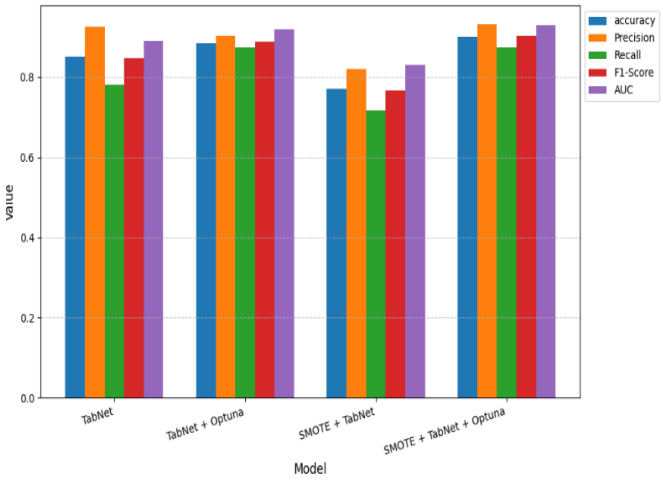

Figure 8. Comparative Visualization of Model Performance Metrics

TABLE VII. OPTIMIZED PARAMETERS FOR DIFFERENTS SCENARIOS

Parameter	Range Value	Scenario I	Scenario II
n_d	8 - 32	21	8
n_a	8 – 32	15	8
n_steps	5 – 8	8	8
n_independent	1 - 2	2	2
learning_rate	0.01 – 0.1	0.024196550894727036	0.02114274661219965
gamma	1.0 – 2.0	1.8305314575602554	1.0251532561037215
lambda_sparse	0.0001 – 0.01	0.0014497065638336094	0.00026367649497584590

TABLE VIII. COMPARATIVE ANALYSIS OF TABNET MODEL PERFORMANCE WITH RELATED RESEARCH

Author	Best Model	Evaluation				
		Accuracy	Precision	Recall	F1-Score	AUC
Hirwono et al. [17]	Naïve Bayes	86.64%	85.07%	89.36%	91.94%	-
Nawawi et al. [18]	Neural Network	84.52%	85.31%	98.85%	-	0.60
Firdaus et al. [19]	MLP	97.50%	97.55%	97.50%	97.48%	-
Baliani et al. [20]	Gradient Boosting	89.50%	-	-	-	-
Ratnasari et al. [21]	Naïve Bayes	84.67%	-	-	-	0.50
Proposed Method	**TabNet**	**90.16%**	**93.33%**	**87.50%**	**90.32%**	**0.93**

Compared to previous studies on heart disease classification Table VIII, our TabNet model with SMOTE and Optuna optimization demonstrated several significant advancements. While some previous approaches have achieved comparable accuracy, our integrated methodology addresses critical limitations in existing methods and offers distinct advantages for real-world clinical applications.

The Naïve Bayes model implemented by Hirwono et al. [17] achieved respectable accuracy (86.64%) but suffered from significant limitations in discriminative capability as evidenced by its unreported AUC values. Similarly, Ratnasari et al. [21] reported a comparable accuracy of 84.67% using Naïve Bayes, but their study revealed an extremely low AUC of only 0.50, effectively equivalent to random guessing in terms of ranking capability. These findings highlight a critical limitation in many previous studies: the overreliance on accuracy as the sole performance metric, which can be misleading in medical diagnostics where false negatives carry serious consequences.

Neural Network approaches, such as that employed by Nawawi et al. [18], showed particularly poor discriminative ability with an AUC of only 0.60 despite reasonable accuracy

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

(84.52%). This substantial performance gap compared to our approach underscores the limitations of conventional neural networks when handling tabular medical data without appropriate attention mechanisms and hyperparameter optimization. The attention mechanism in TabNet provides a critical advantage by focusing on the most relevant features for each individual case, unlike traditional neural networks which process all features equally. Research by Firdaus et al. [19] reported high accuracy (97.55%) using MLP, but this result was achieved using a 90:10 train-test split ratio, which can artificially inflate performance metrics compared to our more robust 80:20 split. Their extreme split ratio likely led to overly optimistic results with limited test samples, whereas our approach with a larger test set provides a more realistic assessment of generalization capability. Additionally, their study lacked comprehensive evaluation across diverse metrics beyond accuracy, particularly AUC, which our research demonstrates is crucial for clinical applications

The Gradient Boosting approach by Baliani et al. [20] achieved 89.5% accuracy using manual parameter tuning with fixed incremental values for learning rate and estimators, whereas our approach leveraged Optuna's Bayesian optimization to systematically explore the parameter space, achieving superior performance (90.16% accuracy). This difference highlights the advantage of our automated optimization strategy over predefined parameter testing, enabling discovery of optimal configurations that manual experimentation likely missed.

IV. CONCLUSION

The implementation of SMOTE technique alone in the context of heart disease classification did not necessarily lead to improved model performance. This was evidenced by the decrease in accuracy from 85.24% to 77.04% in the baseline TabNet implementation. Statistical analysis of the training loss patterns ($p < 0.0001$) revealed that applying SMOTE without appropriate parameter adjustments actually caused the model to overfit to synthetic samples rather than learning generalizable patterns from the data. This phenomenon was particularly pronounced in our relatively small dataset, where synthetic samples generated by SMOTE failed to adequately capture the complexity of real patient data.

However, when SMOTE was combined with parameter optimization using Optuna, the model achieved its best overall performance with 90.16% accuracy, 93.33% precision, 87.50% recall, 90.32% F1-score, and an AUC of 0.93. This significant improvement across all metrics demonstrated the synergistic effect of combining appropriate data balancing techniques with hyperparameter optimization, where Optuna effectively counteracted potential overfitting by fine-tuning regularization parameters.

The key contributions of this research included:

- Identification of the potential negative impact of SMOTE when applied in isolation
- Demonstration of Optuna's effectiveness in optimizing TabNet parameters
- Achievement of state-of-the-art discriminative capability with an AUC of 0.93, representing a substantial improvement over previous approaches

These findings have important implications for clinical applications, where improved classification accuracy and reliability could support earlier and more accurate diagnosis of heart disease, potentially improving patient outcomes through timely interventions. The sequential attention mechanism of TabNet, when properly optimized, provides an interpretable model that could help clinicians understand the factors contributing to a particular diagnosis.

Future research could extend this work by investigating the application of different optimization techniques to TabNet model, and evaluation on larger datasets to further validate the generalizability of our approach. Additionally, exploring the interpretability aspects of the optimized TabNet model could provide valuable insights for medical practitioners in understanding the factors contributing to heart disease classification.

REFERENCES

[1] G.A. Roth, C. Johnson, A. Abajobir, F. Abd-Allah, S.F. Abera, G. Abyu, et al., "Global, regional, and national burden of cardiovascular diseases for 10 causes, 1990 to 2015," J. Am. Coll. Cardiol., vol. 70, no. 1, pp. 1-25, 2017.

[2] M.I. Jordan and T.M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349, no. 6245, pp. 255-260, 2015.

[3] S. Falkner, A. Klein, and F. Hutter, "Practical hyperparameter optimization for deep learning," in AutoML: Methods, Systems, Challenges, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Cham: Springer, pp. 3-25, 2018.

[4] A. Homaidi and Z. Fatah, "Implementasi metode K-nearest neighbors (KNN) untuk klasifikasi penyakit jantung," G-Tech: Jurnal Teknologi Terapan, vol. 8, no. 3, pp. 1720-1728, 2024.

[5] A. Masruriyah, H. Novita, C. Sukmawati, A. Ramadhan, S. Arif, and B. Dermawan, "Pengukuran kinerja model klasifikasi dengan data oversampling pada algoritma supervised learning untuk penyakit jantung," Computer Science (CO-SCIENCE), vol. 4, no. 1, pp. 62-70, 2024.

[6] S.Ö. Arik and T. Pfister, "TabNet: Attentive interpretable tabular learning," in Proc. AAAI Conf. Artif. Intell., vol. 35, no. 8, pp. 6679-6687, May 2021.

[7] A.R. Raharja, A. Pramudianto, and Y. Muchsam, "Penerapan algoritma decision tree dalam klasifikasi data 'Framingham' untuk menunjukkan risiko seseorang terkena penyakit jantung dalam 10 tahun mendatang," Technol. J., vol. 1, no. 1, 2024.

[8] D. Nasien, et al., "Klasifikasi penyakit jantung menggunakan decision tree dan KNN menggunakan ekstraksi fitur PCA," JEKIN-Jurnal Teknik Informatika, vol. 4, no. 1, pp. 18-24, 2024.

[9] T. Indriyani, et al., "Metode decision tree C4.5 untuk klasifikasi penyakit jantung," Prosiding Seminar Nasional Sains dan Teknologi Terapan, no. 1, 2024.

[10] J.J. Tamilselvi and C.B. Gifta, "Handling duplicate data in data warehouse for data mining," Int. J. Comput. Appl., vol. 15, no. 4, pp. 7-15, 2011.

[11] S.K. Kwak and J.H. Kim, "Statistical data preparation: Management of missing values and outliers," Korean J. Anesthesiol., vol. 70, no. 4, pp. 407, 2017.

[12] J. Brownlee, Imbalanced Classification with Python: Better Metrics, Balance Skewed Classes, Cost-Sensitive Learning, Machine Learning Mastery, 2020.

[13] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 2623-2631, July 2019.

[14] C. Sammut and G. I. Webb, Eds., Encyclopedia of Machine Learning, Springer Science & Business Media, 2011.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025

[15] S. Sathyanarayanan and B. R. Tantri, "Confusion matrix-based performance evaluation metrics," Afr. J. Biomed. Res., vol. 4023, pp. 4023-4031, 2024.

[16] A. Tafvizi, B. Avci, and M. Sundararajan, "Attributing AUC-ROC to analyze binary classifier performance," arXiv preprint arXiv:2205.11781, 2022.

[17] B. Hirwono, A. Hermawan, and D. Avianto, "Implementasi metode Naïve Bayes untuk klasifikasi penderita penyakit jantung," J. JTIK (Jurnal Teknol. Inf. Komun.), vol. 7, no. 3, pp. 450-457, 2023.

[18] H. M. Nawawi, J. J. Purnama, and A. B. Hikmah, "Komparasi algoritma neural network dan Naïve Bayes untuk memprediksi penyakit jantung," J. Pilar Nusa Mandiri, vol. 15, no. 2, pp. 189-194, 2019.

[19] R. Firdaus, D. Mualfah, and J. S. Hasanah, "Klasifikasi multi-class penyakit jantung dengan SMOTE dan Pearson's correlation menggunakan MLP," J. CoSciTech (Comput. Sci. Inf. Technol.), vol. 4, no. 1, pp. 262-271, 2023.

[20] M. D. I. Baliani, R. R. Huizen, and G. A. Pradipta, "Perbandingan performa data penyakit jantung menggunakan pendekatan klasifikasi boosting methods," in *Seminar Hasil Penelitian Informatika dan Komputer (SPINTER)*, Institut Teknologi dan Bisnis STIKOM Bali, 2024, pp. 894-899.

[21] A. J. Wahidin, A. E. Setiawan, and P. Bintoro, "Machine learning untuk klasifikasi penyakit jantung," *Aisyah J. Inf. Electr. Eng. (AJIEE)*, vol. 6, no. 1, pp. 145-150, 2024.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2348, Copyright ©2025
Submitted : April 30, 2025, Revised : May 12, 2025, Accepted : May 14, 2025, Published : May 26, 2025