ICIC Express Letters
Volume **19**, Number **6**, June **2025**

ICIC International ©2025 ISSN 1881-803X
pp. **617–625**

OPTIMIZATION OF GB CLASSIFIER FOR PREDICTING CORONARY HEART DISEASE WITH SMOTE AND TPE

Emil Hardiansyah[1], Zahir Zainuddin[1,*] and Syafaruddin[2]

[1]Department of Informatics
[2]Department of Electrical Engineering
Universitas Hasanuddin
Jl. Poros Malino Km. 6 Bontomarannu, Gowa 92171, Sulawesi Selatan, Indonesia
hardiansyahe22d@student.unhas.ac.id; syafaruddin@unhas.ac.id
*Corresponding author: zahir@unhas.ac.id

Received July 2024; accepted October 2024

Abstract. *Cardiovascular disease is the highest reason of sudden dying worldwide, claiming an assessed 17.9 million lives each year. Accurate diagnosis of cardiovascular diseases is essential, but clinical data often exhibit significant variability, noise, and imbalance, and these issues can impact the model's performance. This study handles the imbalanced data problem using the Synthetic Minority Oversampling Technique (SMOTE) to improve the model's performance in detecting coronary heart disease. In addition, we utilize a Gradient Boosting algorithm optimized with a Tree-structured Parzen Estimator (TPE) for hyperparameter tuning. The efficiency of TPE in narrowing the search space based on previous results enables faster identification of optimal hyperparameters than traditional methods such as Grid Search and Random Search. Using SMOTE and TPE improved model accuracy while maintaining computational efficiency for predicting cardiovascular disease using clinical data.*

Keywords: Machine learning, Cardiovascular disease, Clinical data analysis, Synthetic minority oversampling technique, Tree-structured Parzen estimators

1. **Introduction.** Suitable data for heart disease classification should have relevant features, clear labels, a sufficiently large dataset size, and a balanced class distribution. In addition, the data should be rid of noise, missing values, and have sufficient variation to help the model classify between classes. With this reliable data, Machine Learning (ML) models can produce more accurate predictions [1]. Clinical records often exhibit significant variability, noise, and imbalanced [2, 3, 4]. These issues can impact the model's performance. The dataset used in this research consisted by utilizing valid data obtained from Kaggle's repository, Starter: Cardiac Data Nhanes [5]. The dataset showed an imbalance of more than 95% difference between participants with and without coronary heart disease. If not addressed, the classification model will likely favour the majority class, resulting in minimal contribution and low sensitivity for the minority class [3, 6, 7]. The dataset used in this study showed an imbalance of more than 95% difference between participants with and without coronary heart disease. If not handled, the classification model will likely favour the majority class, resulting in minimal contribution and low sensitivity for the minority class. Therefore, an appropriate approach is needed to handle this imbalance to ensure that the classification model does not favour the majority class and still significantly contributes to detecting cases in the minority class. In previous studies [8, 9, 10], the problem of imbalanced data was solved using the Synthetic Minority Oversampling Technique (SMOTE) method, which was shown to improve model performance on more representative metrics such as precision, recall, and F1-score, especially for minority classes. Although the SMOTE method proved effective in overcoming imbalanced

DOI: 10.24507/icicel.19.06.617

data and improving model performance, further efforts are needed to optimize the model through hyperparameter tuning. Identifying the optimal hyperparameters is one of the most essential and time-consuming factors of the modelling task, which previous studies on heart disease prediction [11, 12, 13, 14] have tried to handle.

To enhance the generalization capability of Gradient Boosting (GB) model, we employ Bayesian Optimization (BO) using the Tree-structured Parzen Estimator (TPE) algorithm as an optimization technique [15]. This approach enhances the model's capacity to handle complex problems with numerous features by seeking the best combination in the hyperparameter space [16].

Unlike methods like Grid Search and Random Search, TPE continuously learns from earlier iterations by gathering data, ultimately providing an optimal set of hyperparameters [17]. This method sequentially narrows the search space with previous results, so it is suitable for optimizing the hyperparameters of ML algorithms that require long training times [18]. In this research, we propose a heart disease prediction system with high accuracy and efficient computation time, employing the TPE and implementing it on the GB model to enhance the accurateness and interpretability of the ML model. Our proposed predictive model with BO-TPE was compared and outperformed other research, such as Least Absolute Shrinkage Selection Operator with Convolutional Neural Network (LASSO-CNN) [19], and One-Dimensional Convolutional Neural Network (ID-CNN) [14].

Our contribution is to improve the accuracy of the GB model by considering an ideal computational time to perform coronary heart disease classification using SMOTE with TPE. In addition, we also evaluate and test three hyperparameter tuning methods: Random Search (RS), Gaussian Process (GP), and Tree-structured Parzen Estimators (TPE). We show that TPE provides the best accuracy but is no better than RS and no worse than GP.

The research is organized as follows. Section 2 discusses the relative research. Section 3 describes the methodology used in this study. Section 4 describes the results and discussion, and the conclusions are presented in Section 5.

2. **Related Studies.** Several studies have explored various Coronary Heart Disease (CHD) prediction techniques using data analysis and ML methods. For instance, Ishaq [10] used data mining techniques to enhance the accurateness of survival prediction for heart failure patients. By using SMOTE to manage imbalanced data and pinpoint crucial features that influence machine learning algorithm performance, the study found that the Extra Tree Classifier achieved the highest accuracy at 0.9262 for predicting patient survival. Future work may investigate various combinations of machine learning models and feature selection methods. However, further development is needed to make predictions from clinical datasets that have imbalanced positive and negative classifications.

On the other hand, Khan Mamun and Elfouly [14] used a One-Dimensional Convolutional Neural Network (1D-CNN) to detect Coronary Heart Disease (CHD) more quickly and accurately than conventional invasive methods. Data from the National Health and Nutrition Examination Survey (NHANES) was used with undersampling techniques to overcome imbalanced data and feature selection using minimum Redundancy Maximum Relevance (mRMR), Recursive Feature Elimination (RFE), and Elastic Net. The model outperformed other algorithms, such as Random Forest, with an accuracy of 80.1% (no-CHD) and 76.9% (CHD). However, this study faced limitations such as the size of the dataset being less diverse, the challenge of imbalanced data, and the optimization of features and hyperparameters that need to be improved. Further research is needed to expand the detection of various heart conditions and integrate more clinical data.

Likely, Dutta et al. [19] focused on Coronary Heart Disease (CHD) prediction using imbalanced clinical data. They developed a Convolutional Neural Network (CNN) with the Least Absolute Shrinkage Selection Operator (LASSO), resulting in CHD prediction

accuracy of 77.3% and non-CHD of 81.8%. These results outperformed traditional methods such as Support Vector Machines and Random Forests. Although effective, this study does not explain how the network copes with imbalanced data and uses a simple architecture to prevent overfitting. Further research is needed to generalize the model to more complex datasets.

On the other hand, Jafar and Lee [20] developed a heart disease prediction system, HypGB, using a Gradient Boosting (GB) with Least Absolute Shrinkage Selection Operator (LASSO) and HyperOpt optimization to improve prediction accuracy. They addressed the problem of variable, noisy, and imbalanced clinical data through preprocessing and feature selection, thus enhancing accuracy to 97.32% on the Cleveland dataset and 97.72% on the Kaggle dataset. Despite success in improving prediction, this research still needs to be expanded to public datasets. It opens up opportunities for further development, such as applying deep learning techniques and broader clinical applications.

Another way to optimize heart disease prediction models is through hyperparameter tuning. Fatih et al. [21] used Bayesian Optimization with both Gaussian Process (BO-GP) and Tree-structured Parzen Estimator (BO-TPE) to improve the accuracy of various algorithms such as Random Forest, Support Vector Macnihe, K-Nearest Neighbors, and Logistic Regression. BO-GP achieved an accuracy of 89.01% on Logistic Regression, while BO-TPE showed similar performance on SVM. However, this study was limited by small datasets and long computation times for BO-GP.

3. Methodology.

3.1. Synthetic minority oversampling technique.

It was found that the disparity between participants with CHD and those without CHD exceeded 95%. Therefore, sampling was required to manage the imbalance. Without this adjustment, the classification model would likely favour the majority class, resulting in minimal contribution or sensitivity from the minority class [6]. Therefore, addressing the imbalance requires the use of the SMOTE. This technique employs an oversampling strategy for the minority class by generating synthetic data specifically for the minority category, in this case, individuals with coronary heart disease. SMOTE operates based on the Nearest Neighbors using the Euclidean distance between data points within the feature space. The generation of synthetic samples is represented in Equation (1) as follows:

$$x_{\text{synthetis}} = x_i + \lambda \times (x_j - x_i) \tag{1}$$

$x_{\text{synthetis}}$ is the synthetic sample to be generated, x_i is the sample from the minority class under consideration, λ is a random number between 0 and 1, which determines how far $x_{\text{synthetis}}$ is between x_i and x_j, and x_j is one of the K-Nearest Neighbors (KNN) of x_i.

3.2. Tree-structured Parzen estimators.

This research used Tree-structured Parzen Estimators (TPE) as the surrogate model for algorithm searching and GB model optimization. Figure 1 illustrates a systematic method for enhancing GB performance using the TPE algorithm and determining the optimal hyperparameter combinations. Split $p(x|y)$ by y^* will form conditional probability in Equation (2) as follows:

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \tag{2}$$

where x is value of single hyperparameter and y is loss. y^* defines threshold that determines whether the value of y is good/bad; a hyperparameter determines this value. TPE models two separate distributions for hyperparameters, namely $l(x)$ a distribution for hyperparameter values that provide good performance and $g(x)$ a distribution for values that provide poor performance [15].

- A distribution for bad values: $p(x|y > y^*) = p(x|bad)$
- A distribution for good values: $p(x|y \leq y^*) = p(x|good)$

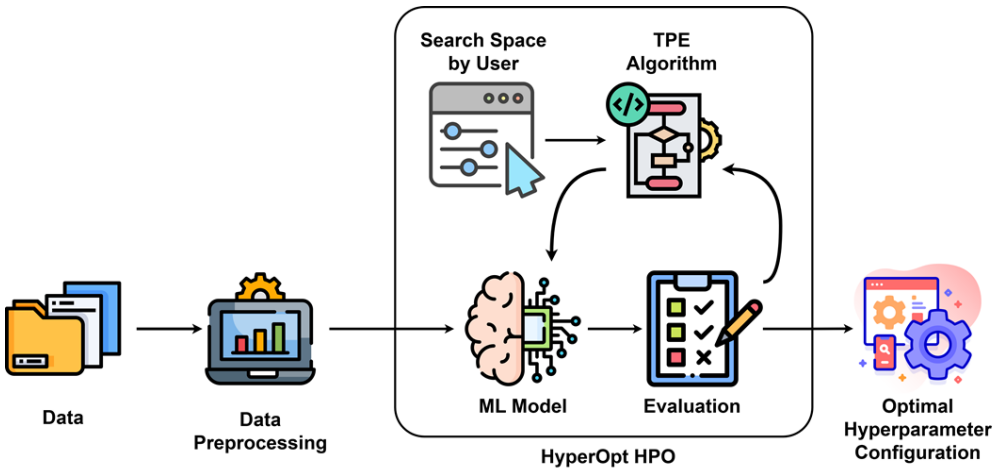


FIGURE 1. TPE optimization process

TPE can more efficiently select hyperparameters that are more likely to give good model performance. This permits the algorithm to concentrate on more relevant domains of the search space and avoid unproductive areas, thus speeding up the optimization process and increasing the chances of finding the optimal hyperparameter combination. The goal is to improve the distribution of $l(x)$: the higher the likelihood, the better the parameters, while for $g(x)$: the lower the likelihood, the better the parameters. So, take the minimum argument in Equation (3) of the lower distribution divided by the upper distribution.

$$\text{argmin}_{\text{params}} = \frac{g(\text{params})}{l(\text{params})} \tag{3}$$

The GB hyperparameters are refined through Bayesian optimization. The best set of parameters extracted from the Bayesian optimization is used to train the GB to obtain the highest classification accuracy. In the hyperparameter optimization process, parameter values must be determined in advance. A wide range of parameters were run in the Bayesian optimization presented in Table 1.

TABLE 1. Parameters and parameter value range

Parameter	Data type	Value range
n_estimators	Integer	50-500, $q = 1$
max_depth	Integer	1-10, $q = 1$
max_leaf_nodes	Integer	1-10, $q = 1$
min_samples_leaf	Integer	1-25, $q = 1$
learning_rate	Float	-5-0
min_samples_split	Integer	2-10, $q = 1$
subsample	Numerical	0.5-1
max_features	String	sqrt, log 2
loss	String	deviance, exponential

The number of trees parameter determines the number of boosting iterations used to fix mistakes made by the earlier trees. The maximum tree depth parameter handles the maximum depth of individual trees in the GB process, which is crucial in preventing overfitting. The limit on the number of leaf nodes caps the number of nodes in the trees, helping to manage the model's complexity. The lowest number of samples per leaf node

parameter sets the lowest number of samples required in a leaf, which helps to avoid overfitting. Finally, the learning rate parameter affects the contribution of each tree in the boosting process, ensuring that updates are gradual and enhancing the model's robustness. Unlike Grid Search, Random Search, and Gaussian Process, which require significant effort and disregard past optimizations, TPE enhances efficiency by learning from previous iterations and consistently improving its recommendations for optimal hyperparameters [17]. Optimization of the classification model in this study uses the Hyperopt open-source framework [16], which automates the optimization process. Hyperopt allows us to automatically find the best combination of hyperparameters to enhance the performance of ML models. Hyperopt uses Bayesian-based optimization techniques to select hyperparameters. Hyperopt, alternatively, learns from previous optimizations by continually collecting data from earlier iterations and supplying a set of excellent hyperparameters. Hyperopt process can be divided into several steps:

1) Describe **search space** via a stochastic program (domain knowledge)
2) Define an **objective function**
3) **Optimize** that objective ($f_{\min}$)
4) **Analyze** results (*Trials*)

Initially, data undergoes pre-processing to prepare it for training. The processed data is then used to train the GB model. The TPE algorithm optimizes the GB model's hyperparameters within a predefined search space, iteratively refining them to enhance model training. This cycle continues until the optimal hyperparameter configuration is achieved, resulting in the best model performance.

4. **Results and Discussions.** SMOTE works by randomly picking K-Nearest Neighbors (KNN) from each instance in the minority class and creating a new (synthetic) instance between that instance and the k randomly chosen Nearest Neighbors as shown in Figure 2. With the SMOTE approach, it can be ensured that there is no data duplication problem, so it is more immune to the overfitting problem. The algorithm selects random samples from the minority class and selects random neighbors using KNN. A synthetic example is created between two examples in the feature space. Synthetics in SMOTE indicate no data duplication in the resampled data. SMOTE increases the sample of minority classes while evading overfitting. This is done by rendering new synthetic samples near other points (belonging to the minority class) in the attribute space.

The authors chose the number of trials as 100, as shown in Figure 3, and this number will determine how many times the optimization algorithm will try different hyperparameter combinations. Each evaluation means that a set of hyperparameters is selected

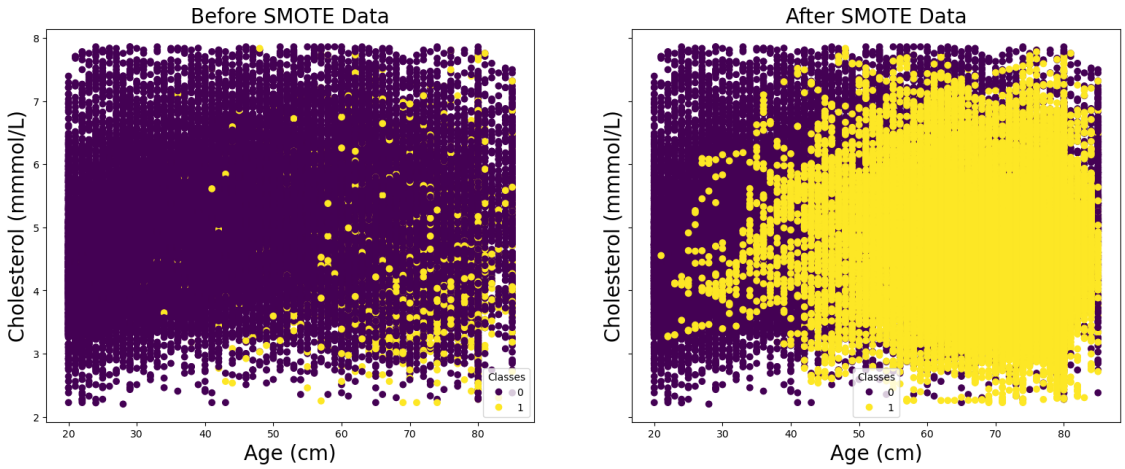


FIGURE 2. Before and after SMOTE

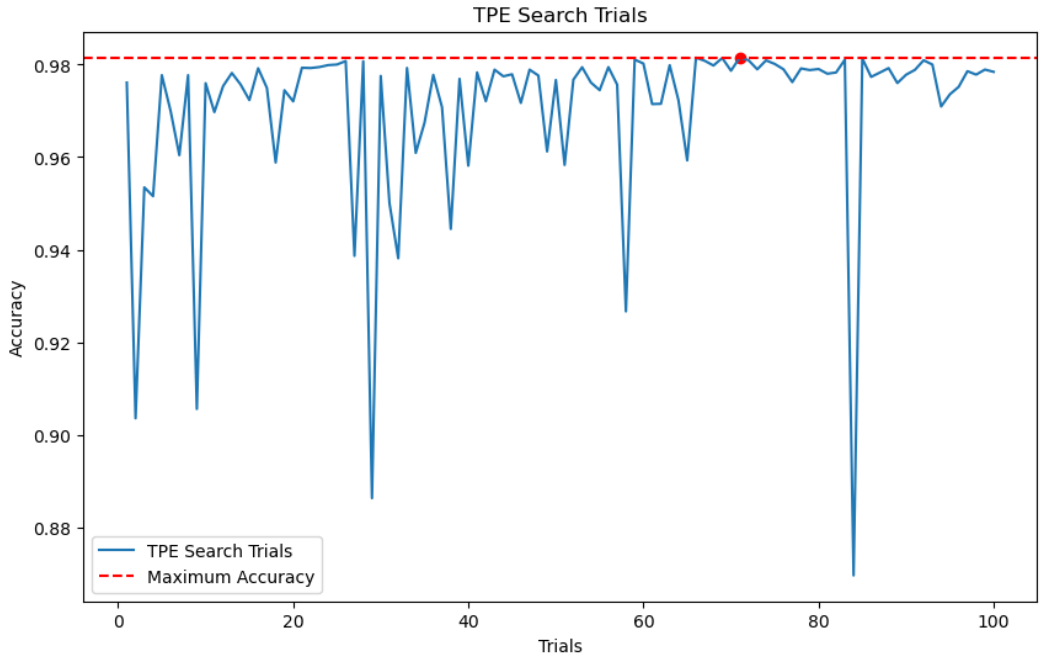


FIGURE 3. Trial search space for 100 iteration

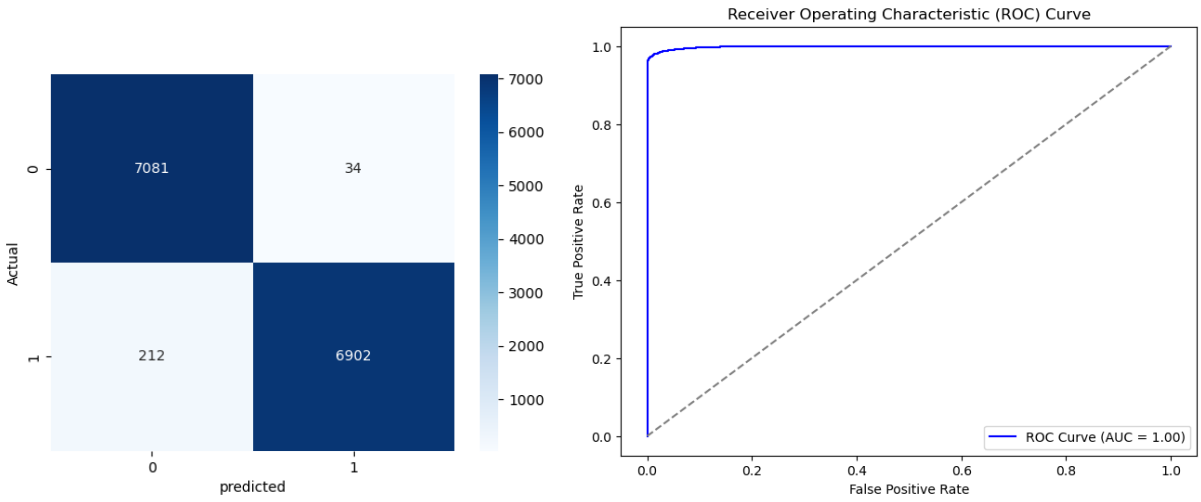
from the search space, and then the model is trained and evaluated using that set of hyperparameters. Hyperopt will stop the search after 100 trials.

Figure 3 gives an overview of how TPE explores the hyperparameter space and how the model performance fluctuates during the process. The model performance shows that most trials have very high accuracy, but some trials produce much lower accuracy. This is because the hyperparameters selected in those trials are not optimal. The fluctuations in the accuracy show that TPE is exploring different parts of the hyperparameter search space, some of which result in lower accuracy. Trials with low accuracy are caused by hyperparameter combinations that are not very suitable for this dataset. It is essential to check the hyperparameters generated by these trials to avoid similar configurations in the future. The objective function will be minimized since the authors used the hyperopt library. Therefore, the objective function should return a smaller value for a better solution. In optimization focusing on minimization, the metric's value is inverted by multiplying it by -1. This turns the minimization problem into a maximization problem. Based on the trials above, it is known that the 71st trial is the best, with a test accuracy of 0.9712 and a computation time of 3040.32 s.

Figure 4(a) describes the confusion matrix results for the TPE model performance. From this confusion matrix, it can be concluded that the model performs well in classifying the data, with the number of correct predictions (TN and TP) being much higher than the number of incorrect predictions (FP and FN). This indicates that the model has a high level of accuracy in detecting the correct class.

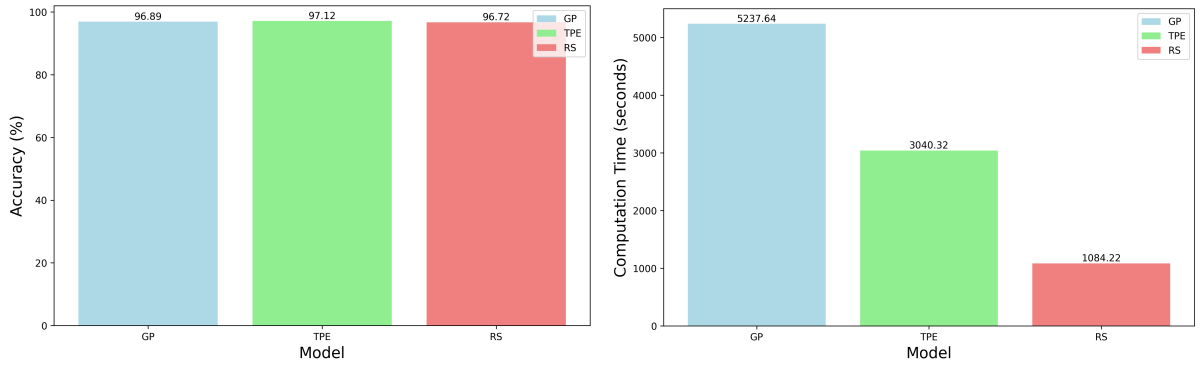
Figure 4(b) shows the Receiver Operating Characteristic (ROC) curve used to evaluate the performance of the classification model. The Area Under the Curve (AUC) was recorded at 1.00, indicating the model has high predictive ability. An AUC close to 1 indicates that the model can almost always distinguish well between positive and negative classes, while an AUC close to 0.5 indicates a performance comparable to random guessing. With an AUC of 1.00, the model can be considered highly effective in performing its classification task.

Comparisons were made on three hyperparameter optimization models, namely Random Search (RS), Tree-structured Parzen Estimators (TPE), and Gaussian Process (GP), by measuring the accuracy and computation time of each model, the results of which are



(a) (b)

FIGURE 4. Result of the proposed model to (a) confussion matrix, and (b) Receiver Operating Characteristic (ROC) curve



(a) (b)

FIGURE 5. Comparison of (a) accuracy, and (b) computation time for each model

shown in Figure 5(a) and Figure 5(b). Figure 5(a) compares the accuracy of the three hyperparameter optimization models: RS, TPE, and GP. From the test results, the TPE model showed the highest accuracy of 97.12%, followed by the RS and GP models, which achieved an accuracy of 96.72% and 96.89%.

Based on Figure 6, we can see the comparison results between the proposed TPE method and the LASSO-CNN and 1D-CNN methods [14]. The results show that the TPE method provides the highest accuracy of 98.27%, better than 1D CNN which has an accuracy of 83.93% and LASSO-CNN which has an accuracy of 81.78%, which means that the TPE method can improve the performance of the model and provide better accuracy compared to CNN and 1D-CNN methods.

5. **Conclusions.** Based on this research, several conclusions were obtained, including that applying SMOTE on clinical heart disease datasets proved effective in handling imbalanced data, increasing minority class representation, and improving classification model performance, especially in recall. Using TPE as a hyperparameter optimization method in the Gradient Boosting algorithm successfully found the optimal hyperparameter combination, improved model accuracy, and reduced computation time compared to

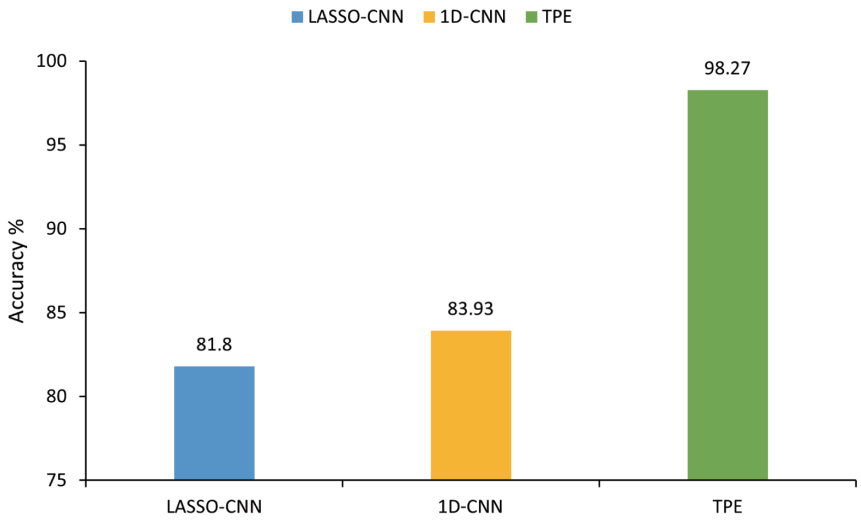


FIGURE 6. Comparison with previous work

the Grid Search and Random Search methods. TPE proved more effective and efficient in improving heart disease classification performance, making it a superior choice for hyperparameter optimization in this model. This research is still limited to single-objective optimization. Future research can adopt MOTPE to handle multiobjective optimization, where not only one performance metric but also multiple objectives are optimized. Integrating MOTPE can help improve prediction accuracy and computation time efficiency simultaneously, which is particularly relevant in systems that require fast response, such as real-time disease detection systems.

REFERENCES

[1] S. Shehzadi, M. A. Hassan, M. Rizwan, N. Kryvinska and K. Vincent, Diagnosis of chronic ischemic heart disease using machine learning techniques, *Computational Intelligence and Neuroscience*, vol.2022, pp.1-9, 2022.

[2] J. A. Sáez, B. Krawczyk and M. Woźniak, On the influence of class noise in medical data classification: Treatment using noise filtering methods, *Applied Artificial Intelligence*, vol.30, pp.590-609, 2016.

[3] I. Araf, A. Idri and I. Chairi, Cost-sensitive learning for imbalanced medical data: A review, *Artificial Intelligence Review*, vol.57, 80, 2024.

[4] M. Salmi, D. Atif, D. Oliva, A. Abraham and S. Ventura, Handling imbalanced medical datasets: Review of a decade of research, *Artificial Intelligence Review*, vol.57, 273, 2024.

[5] *National Center for Health Statistics*, https://wwwn.cdc.gov/nchs/nhanes/ContinuousNhanes/Questionnaires.aspx?BeginYear=2015, 2015.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, vol.16, pp.321-357, 2002.

[7] J. M. Johnson and T. M. Khoshgoftaar, Survey on deep learning with class imbalance, *Journal of Big Data*, vol.6, 27, 2019.

[8] A. Fernandez, S. Garcia, F. Herrera and N. V. Chawla, SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary, *Journal of Artificial Intelligence Research*, vol.61, pp.863-905, 2018.

[9] A. Bisri and R. Rachmatika, Integration of gradient boosted trees with SMOTE and bagging for student graduation prediction, *Indonesian Journal of Electrical Engineering and Informatics*, vol.8, 309, 2019.

[10] A. Ishaq, S. Sadiq, M. Umer, S. Ullah, S. Mirjalili, V. Rupapara and M. Nappi, Improving the prediction of heart failure patients' survival using SMOTE and effective data mining techniques, *IEEE Access*, vol.9, pp.39707-39716, 2021.

[11] S. M. S. Shah, F. A. Shah, S. A. Hussain and S. Batool, Support vector machines-based heart disease diagnosis using feature subset, wrapping selection and extraction methods, *Computers & Electrical Engineering*, vol.84, 106628, 2020.

[12] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan and A. Saboor, Heart disease identification method using machine learning classification in e-healthcare, *IEEE Access*, vol.8, pp.107562-107582, 2020.
[13] R. Valarmathi and T. Sheela, Heart disease prediction using Hyper Parameter Optimization (HPO) tuning, *Biomedical Signal Processing and Control*, vol.70, 103033, 2021.
[14] M. M. R. Khan Mamun and T. Elfouly, Detection of cardiovascular disease from clinical parameters using a one-dimensional convolutional neural network, *Bioengineering*, vol.10, 796, 2023.
[15] S. E. Suryana, B. Warsito and S. Suparti, Application of gradient boosting with hyperopt to predict bank telemarketing success, *Gaussian Journal*, vol.10, pp.617-623, 2021.
[16] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins and D. D. Cox, Hyperopt: A Python library for model selection and hyperparameter optimization, *Computational Science & Discovery*, vol.8, 014008, 2015.
[17] A. Jafar and M. Lee, Comparative performance evaluation of state-of-the-art hyperparameter optimization frameworks, *The Transactions of the Korean Institute of Electrical Engineers*, vol.72, pp.607-620, 2023.
[18] J. Bergstra, D. Yamins and D. Cox, Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms, *Python in Science Conference*, Austin, TX, pp.13-19, 2013.
[19] A. Dutta, T. Batabyal, M. Basu and S. T. Acton, An efficient convolutional neural network for coronary heart disease prediction, *Expert Systems with Applications*, vol.159, 2019.
[20] A. Jafar and M. Lee, HypGB: High accuracy GB classifier for predicting heart disease with HyperOpt HPO framework and LASSO FS method, *IEEE Access*, vol.11, pp.138201-138214, 2023.
[21] F. Fatih, Z. En-Naimani and K. Haddouch, Comparative study of Bayesian optimization process for the best machine learning hyperparameters, in *Proceedings of the 6th International Conference on Big Data and Internet of Things*, M. Lazaar, E. M. En-Naimi, A. Zouhair, M. Al Achhab and O. Mahboub (eds.), Cham, Springer International Publishing, 2023.