JOIV : Int. J. Inform. Visualization, 10(1) - January 2026 262-270

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

# Classification of Heart Disease with Machine Learning: A Comparison of Grid Search, Random Search, and Bayesian Optimization

Tri Andi [a,*], Amelia Ritahani Ismail [b], Andri Pranolo [c], Candra Juni Cahyo Kusuma [d]

[a] Universitas Muhammadiyah Yogyakarta, Yogyakarta, Indonesia
[b] Department of Computer Science, International Islamic University Malaysia, Malaysia
[c] Department of Informatics, Universitas Ahmad Dahlan, Indonesia
[d] Universitas PGRI Yogyakarta, Yogyakarta, Indonesia
Corresponding author: *triandi@umy.ac.id

***Abstract*—This study systematically evaluates the effect of hyperparameter optimization on the performance of predictive models by comparing three main techniques: Grid Search, Random Search, and Bayesian Optimization. Four commonly used machine learning algorithms: Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Gradient Boosting were tested on benchmark datasets from the UC Machine Learning Repository. The results of the study show that hyperparameter optimization significantly improves prediction accuracy compared to baseline models, with the optimal method varying across algorithms. Specifically, Random Search achieved the highest accuracy of 0.883333 for Logistic Regression and 0.833333 for Gradient Boosting, while Bayesian Optimization demonstrated superior performance on SVM with an accuracy of 0.883333. Grid Search proved most effective for KNN, achieving an accuracy of 0.866667. A comprehensive analysis using additional metrics such as precision and recall reinforces these findings, showing that accuracy improvements do not come at the expense of other performance metrics. Bayesian Optimization stands out for computational efficiency, especially for complex models, though Grid Search remains relevant for limited hyperparameter spaces. The main contribution of this research is to provide practical guidance for machine learning practitioners on selecting optimization techniques that align with algorithm characteristics. This study also analyses the trade-off between search thoroughness and computational resources required, and provides recommendations for balancing accuracy and efficiency in model development. Although the research results are promising, several limitations need to be acknowledged, including reliance on a single dataset and the need for further validation across multiple domains. Future research could explore hybrid optimization approaches that combine the strengths of various methods, develop adaptive strategies for ensemble models, or conduct more comprehensive evaluations across diverse datasets. By linking theoretical findings to practical applications, this research provides a valuable framework for efficiently optimizing machine learning models.***

***Keywords*—Hyperparameter optimization; logistic regression; support vector machine; K-Nearest Neighbors; gradient boosting; grid search; random search; bayesian optimization; machine learning; model performance.**

*Manuscript received 7 Feb. 2025; revised 13 Jul. 2025; accepted 24 Sep. 2025. Date of publication 30 Jan. 2026.*


## I. INTRODUCTION

Heart disease is one of the leading causes of death worldwide, with a mortality rate of around 17.9 million people per year, according to the World Health Organization (WHO) [1]–[3]. Early detection and accurate diagnosis are essential to reduce mortality [4]–[7]. In recent years, machine learning has become an effective tool in aiding the diagnosis of heart disease by automatically analyzing patient data [2], [8]–[11]. Previous studies have proposed classifying heart disease using ML [7], [12]. Such as the research conducted by Azis et al using the Logistic Regression method [13], Osei-Nkwantabisa and Ntumy using the KNN method [14].

One way to improve machine learning performance is by doing hyperparameter tuning [15]–[24]. Such as research conducted by Ahamad et al [25], Franceschi et al [26], and Baseer [6]. Although many studies have been conducted, most rely on default hyperparameter configurations or use only a single tuning technique, such as Grid Search or Random Search. In addition, some studies do not consider computational efficiency when selecting tuning methods. Therefore, there remains a research gap in developing an optimal hyperparameter tuning method to improve the

accuracy and efficiency of the heart disease classification model.

This study presents novelty by combining and comparing three hyperparameter tuning methods rarely used simultaneously in previous research: Grid Search, Random Search, and Bayesian Optimization. By evaluating these three techniques on three major ML algorithms (Logistic Regression, SVM, KNN, and Gradient Boosting), this study provides a more comprehensive insight into the best tuning methods for heart disease classification. Another advantage of this study is that it considers not only the model's accuracy but also its computational efficiency when selecting the tuning method. Thus, the results of this study can serve as a reference for practitioners and academics in selecting the optimal combination of models and tuning techniques for ML-based heart disease diagnosis.

## II. Materials and Methods

### A. Overview of Machine Learning Approaches in Heart Disease Classification

The development of machine learning in heart disease classification has shown rapid progress in recent years. Various machine learning algorithms have been applied with a focus on improving accuracy and optimizing hyperparameters. The following comparative analysis examines current approaches to provide context for systematic hyperparameter optimization studies.

TABLE I
CLASSIFICATION BASED ON MACHINE LEARNING APPROACH

| Author & Year | Primary Algorithm | Accuracy | Dataset | Key Strengths | Methodology |
|---|---|---|---|---|---|
| Barus et al. (2024) [27] | Enhanced C4.5 Decision Tree | 73.25% | - | Outperforms kNN and Naïve Bayes; generates 7 classification rules | Pruning techniques for efficiency; uses blood pressure, cholesterol, glucose attributes |
| Boer et al. (2023) [10] | Logistic Regression | 92.61% | - | Best performance among 7 algorithms tested | Comparative analysis with KNN, Random Forest, Gaussian Naive Bayes, XGBoost, SVM, Decision Tree |
| Mohsen et al. (2024) [28] | Random Forest | Not specified | Medical records from Ibb Medical Clinic, Yemen | Best in accuracy, sensitivity, specificity, and F-measure | Implementation of 14 different algorithms using Weka; web system development with ASPX |
| Alfonse (2019) [29] | Ensemble (MLP + K-NN + C4.5) | 99.4% | UCI repository (4 databases) | Highest accuracy with voting mechanism | Comprehensive preprocessing; 10-fold cross-validation; ensemble technique |
| Habelalmateen et al. (2023) [30] | IPSO-SVM | 99.95% | Cleveland dataset | Highest accuracy; addresses unbalanced samples and missing values | Feature optimization with IPSO; feature extraction using ResNet 50 |
| Balamurugan et al. (2021) [31] | Adaptive Harris Hawk Optimization + Enhanced Deep Genetic Algorithm | 98.36% (feature selection) 97.3% (classification) | - | Combines clustering and deep learning; neural network weight optimization | Stochastic gradient boosting with recursive feature elimination |

The comparative analysis shows that recent studies have achieved accuracy ranging from 73.25% to 99.95% using various machine learning approaches. Advanced hybrid methods such as IPSO-SVM and ensemble techniques have demonstrated exceptional performance, with the highest reported accuracy reaching 99.95%. However, most studies have not explicitly compared the effectiveness of various hyperparameter optimization techniques, such as Grid Search, Random Search, and Bayesian Optimization. This gap in the literature prompted this study to conduct a systematic comparison between Grid Search, Random Search, and Bayesian Optimization in optimizing heart disease classification performance. With a specific focus on hyperparameter optimization strategies, this research addresses the critical need for a standardized evaluation of optimization techniques that could improve the performance of existing machine learning algorithms for heart disease classification.

### B. Dataset

The dataset used in this study is a heart disease dataset from the UCI Machine Learning Repository [32]. This dataset consists of 303 samples with 14 attributes, including age, gender, blood pressure, cholesterol levels, and others. The target variable is the diagnosis of heart disease (0 = no disease, 1 = no disease).

TABLE II
HEART DISEASE DATASET FEATURES [32]

| Variable Name | Role | Type | Demographics | Description | Units | Missing Values |
|---|---|---|---|---|---|---|
| Age | Featured | Integer | Age | Age | years | It |
| Sex | Featured | Categorical | Sex | Sex | | It |
| Cp | Featured | Categorical | | Chest pain type | | It |
| trestbps | Featured | Integer | | Resting blood pressure (on admission to the hospital) | mm Hg | It |
| chol | Featured | Integer | | Serum cholesterol | mg/dl | It |
| FBS | Featured | Categorical | | Fasting blood sugar > 120 mg/dl | | It |
| restecg | Featured | Categorical | | | | It |
| Thalach | Featured | Integer | | Maximum heart rate achieved | | It |
| Exang | Featured | Categorical | | Exercise-induced angina | | It |
| oldpeak | Featured | Integer | | ST depression induced by exercise relative to rest | | It |
| Slope | Featured | Categorical | | number of major vessels (0-3) colored by fluoroscopy | It | |
| Ca | Featured | Integer | | | Yes | |
| thal | Featured | Categorical | | | Yes | |
| Num | Target | Integer | | Diagnosis of heart disease | It | |

### C. Preprocessing Data

The data undergoes a preprocessing stage before being applied to an ML model. The first stage is the handling of missing values. Missing values in a dataset are handled using a mean or median-based imputation method, depending on the data distribution of each feature [33]–[35]. The second stage is data normalization. All numerical features are normalized using the Min-Max Scaling method to have a value range between 0 and 1, thus helping the ML model in faster and more stable convergence [36], [37]. The third stage is the distribution of datasets. The dataset is split into 80% training and 20% test data, stratified to ensure balanced class distribution during model evaluation.

### D. Machine Learning Models

This study uses four ML algorithms for heart disease classification: Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Gradient Boosting. The first model is Logistic Regression (LR), which is used because of its simplicity and ability to handle binary classification [38]–[40]. The second model is the Support Vector Machine (SVM). SVM is a margin-based Algorithm that searches for the optimal hyperplane to separate two classes with margin maximization [41]–[45]. The third model is K-Nearest Neighbors (KNN). K-Nearest Neighbors (KNN) is a non-parametric model that classifies samples based on the majority of classes from k-nearest neighbors [46]–[49]. The fourth model is Gradient Boosting. Gradient Boosting is a decision tree-based ensemble algorithm that builds models incrementally to reduce previous prediction errors and improve classification accuracy [50]–[53]. Training will be carried out to produce models without hyperparameter optimization from the four existing models. This model will later serve as a basis for comparison with models optimized for hyperparameters.

### E. Hyperparameter Optimization

To improve the performance of the model, hyperparameter tuning is carried out using three methods, namely

*1) Grid Search*: A brute-force method that tries all combinations of hyperparameters that have been defined in the search grid. While it guarantees optimal search, this method requires high computational time [54], [55].

*2) Random Search*: This method performs random sampling in the hyperparameter search space, which allows for more efficient searches than Grid Search, even if it may not find the optimal combination [26], [56], [57].

*3) Bayesian Optimization*: Uses a probabilistic model based on the Gaussian Process to predict the best combination of hyperparameters based on previous evaluations, making it more efficient than the other two methods [58], [59].

The hyperparameter lookup table for each method is shown below:

TABLE III
HYPERPARAMETER SEARCH OF LOGISTIC REGRESSION METHODS

| Hyperparameters | Search Space | Type |
|---|---|---|
| Regularization (C) | [0.001, 1000] | Continuous |
| Solver | ['newton-cg', 'lbfgs', 'liblinear'] | Discrete |

TABLE IV
HYPERPARAMETER SEARCH METHOD SUPPORT VECTOR MACHINE

| Hyperparameters | Search Space | Type |
|---|---|---|
| Kernel | ['linear', 'rbf', 'poly', 'sigmoid'] | Discrete |
| Regularization (C) | [0.1, 10] | Continuous |
| Gamma | ['scale', 'auto'] | Discrete |

TABLE V
K-NEAREST NEIGHBORS METHOD HYPERPARAMETER SEARCH

| Hyperparameters | Search Space | Type |
|---|---|---|
| Number of Neighbors (k) | [1, 50] | Continuous |
| Distance Metric | ['Euclidean', 'Manhattan', 'Minkowski', 'uniform'] | Discrete |

TABLE VI
GRADIENT BOOSTING METHOD HYPERPARAMETER SEARCH

| Hyperparameters | Search Space | Type |
|---|---|---|
| Number of Estimators | [50, 200] | Continuous |
| Learning Rate | [0.01, 0.3] | Continuous |
| Max Depth | [3, 10] | Continuous |

Hyperparameter searches are performed thoroughly on the predefined parameters for each algorithm. The results of the hyperparameter search for the Logistic Regression method

(Table 6), SVM (Table 7), KNN (Table 8), and Gradient Boosting (Table 9).

TABLE VII
RESULTS OF THE SEARCH FOR HYPERPARAMETERS OF LOGISTIC REGRESSION METHODS

| Hyperparameters | Grid Search | Random Search | Bayesian Optimization |
|---|---|---|---|
| Regularization (C) | 0.01 | 0.017787658410143285 | 0.02048675394458509 |
| Solver | lbfgs | lbfgs | lbfgs |

TABLE VIII
HYPERPARAMETER SEARCH RESULTS OF SUPPORT VECTOR MACHINE METHOD

| Hyperparameters | Grid Search | Random Search | Bayesian Optimization |
|---|---|---|---|
| Kernel | RBF | RBF | RBF |
| Regularization (C) | 0.1 | 0.30584494295802445 | 0.1362241649868 |
| Gamma | scale | Auto | scale |

TABLE IX
HYPERPARAMETER SEARCH RESULTS OF K-NEAREST NEIGHBORS METHOD

| Hyperparameters | Grid Search | Random Search | Bayesian Optimization |
|---|---|---|---|
| Number of Neighbors (k) | 9 | 7 | 15 |
| Distance Metric | uniform | uniform | uniform |

TABLE X
GRADIENT BOOSTING METHOD HYPERPARAMETER SEARCH RESULTS

| Hyperparameters | Grid Search | Random Search | Bayesian Optimization |
|---|---|---|---|
| Number of Estimators | 100 | 174 | 300 |
| Learning Rate | 0.1 | 0.0412037280884873 | 0.027822281883312608 |
| Max Depth | 5 | 5 | 4 |

### *F. Model Evaluation*

This study evaluated the model using the metrics of accuracy, precision, recall, F1-score, and Confusion Matrix [60]–[68]. In addition, cross-validation with a 5-fold is carried out to ensure the model's reliability [69]–[77]. Cross-validation helps reduce the risk of overfitting and provides a more accurate estimate of model performance on data it has never seen before.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (4)$$

Information:
TP: True Positive (correct prediction for positive classes).
TN: True Negative (correct prediction for negative classes).
FP: False Positive.
FN: False Negative.

## III. RESULTS AND DISCUSSION

The experiment was conducted by applying four ML algorithms (Logistic Regression, Support Vector Machine, K-Nearest Neighbors, and Gradient Boosting) to the heart disease dataset using various hyperparameter tuning methods (Grid Search, Random Search, and Bayesian Optimization).

TABLE XI
RESULTS OF BASIC MODEL EVALUATION

| Type | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Logistic Regression | 0.833333 | 0.846154 | 0.785714 | 0.814815 |
| SVM | 0.85 | 0.88 | 0.785714 | 0.830189 |
| KNN | 0.85 | 0.88 | 0.785714 | 0.830189 |
| Gradient Boosting | 0.766667 | 0.769231 | 0.714286 | 0.740741 |

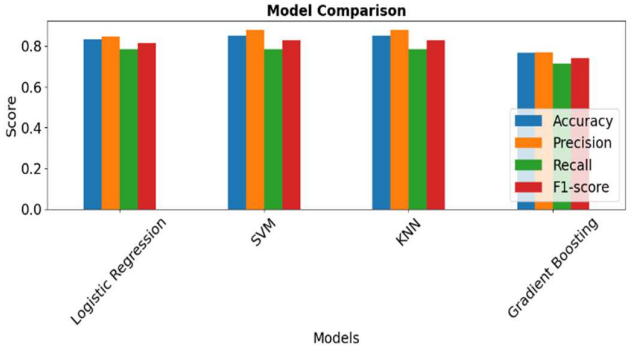


Fig. 1 Basic Model Evaluation Results (without hyperparameter tuning)

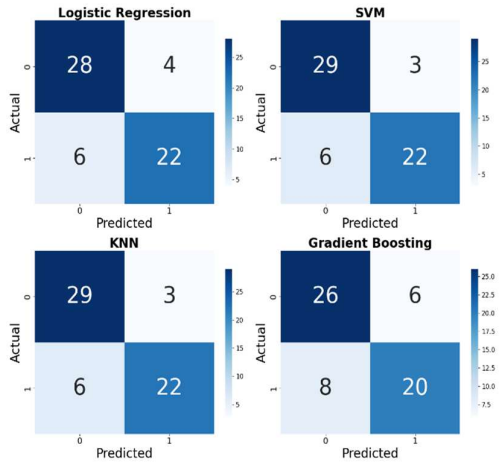


Fig. 2 Basic Confusion Matrix model (without hyperparameter tuning)

The evaluation results for the basic/non-hyperparameter tuning model are shown in Table 10. The basic Logistic Regression model performs well, achieving 83.33% accuracy and an F1-score of 81.48%. The accuracy of this model is 84.62%, indicating its ability to accurately identify positive data. However, the recall of 78.57% indicates that some positive data points are not well detected by this model. Logistic Regression is a solid choice in this case, although it lags slightly behind SVM and KNN.

SVM delivers the best performance of all models with an accuracy of 85% and an F1-score of 83.02%. The precision and recall of this model are both high, namely 88% and 78.57%, respectively. This shows that SVM is not only able to identify positive data well, but also maintains a balance between precision and recall. With these results, the SVM can be considered the most reliable model for this case.KNN performs as well as SVM in terms of accuracy (85%) and precision (88%), and yields an identical F1-score of 83.02%. Nonetheless, these similarities indicate that KNN is a viable alternative to SVM, especially when model interpretability or training speed are key considerations.

Gradient Boosting performs less than other models, with an accuracy of 76.67% and an F1-score of 74.07%. The accuracy of 76.92% and recall of 71.43% indicated that this model was better at identifying positive data than Logistic Regression, but it had weaknesses in detecting overall positive data (recall). Gradient Boosting can be improved through hyperparameter optimization to improve model performance.

Based on the evaluation of the base model, SVM and KNN are the best-performing models, followed by Logistic Regression and Gradient Boosting. SVM excels in terms of precision and F1-score, making it a top choice for this analysis. However, Gradient Boosting is still worth considering for further analysis, especially if it comes with parameter optimization or feature addition.

After tuning the hyperparameters using Grid Search, the model's performance improved significantly across several evaluation metrics. The results of the evaluation after the tuning process are shown in the following Table 11:

TABLE XII
RESULTS OF MODEL EVALUATION WITH HYPERPARAMETER TUNING GRID SEARCH

| Type | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Logistic Regression | 0.866667 | 0.916667 | 0.785714 | 0.846154 |
| SVM | 0.866667 | 0.916667 | 0.785714 | 0.846154 |
| KNN | 0.866667 | 0.884615 | 0.821429 | 0.851852 |
| Gradient Boosting | 0.783333 | 0.8 | 0.714286 | 0.754717 |

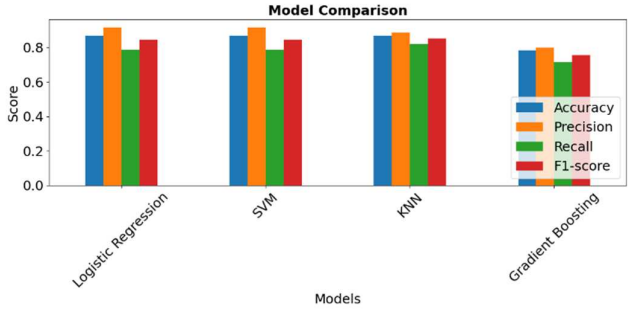


Fig. 3 Results of Model Evaluation with Grid Search

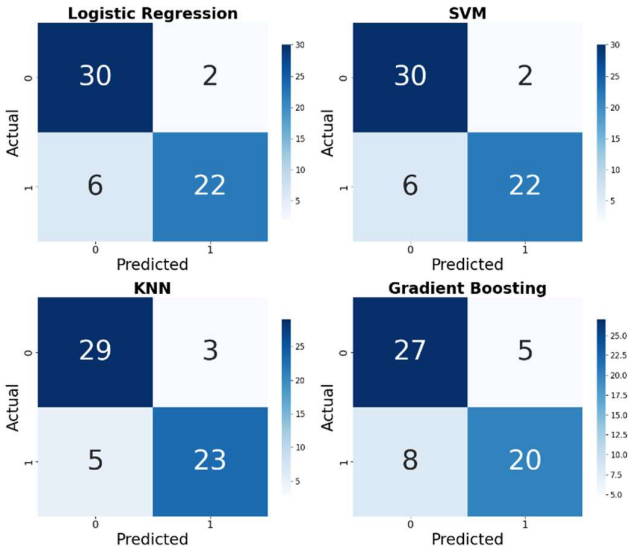


Fig. 4 Confusion Matrix Model with Grid Search

After tuning using Grid Search, Logistic Regression experienced a significant performance improvement, with accuracy increasing to 86.67% and F1-score remaining stable at 84.62%. The model's accuracy also increased to 91.67%, indicating a better ability to correctly identify positive data. The recall remained at 78.57%, indicating that the model still struggles to detect the overall positive data.

Just like Logistic Regression, SVM shows identical performance with an accuracy of 86.67% and an F1-score of 84.62%. High precision (91.67%) and recall (78.57%) indicate that hyperparameter tuning improved accuracy while maintaining a balance across evaluation metrics. SVM remains one of the recommended models based on these results.

KNN has shown a significant improvement, with an accuracy of 86.67% and an F1-score of 85.19%. The recall of this model is the highest among all models at 82.14%, demonstrating KNN's ability to detect positive data more consistently. Precision is slightly lower than Logistic Regression and SVM (88.46%), but still good enough for practical applications.

Gradient Boosting showed a moderate improvement after tuning, achieving 78.33% accuracy and 75.47% F1-score. Precision increased to 80%, but recall remained at 71.43%. This shows that Gradient Boosting still has limitations in detecting positive data overall. Compared to other models, Gradient Boosting remains the worst performer.

After hyperparameter tuning using Grid Search, the three best models (Logistic Regression, SVM, and KNN) achieved the same accuracy of 86.67%. However, KNN has an advantage in recall value (82.14%) and F1-score (85.19%), making it the most balanced model to use in this case. Meanwhile, Logistic Regression and SVM remain strong choices thanks to higher precision (91.67%). Gradient Boosting, although improved, still lags behind the other three models.

After tuning the hyperparameters using the Random Search method, the model's performance also showed a varied improvement. Table 12 presents the model evaluation results for the main metrics: accuracy, precision, recall, and F1-score.

TABLE XIII
RESULTS OF MODEL EVALUATION WITH HYPERPARAMETER TUNING RANDOM SEARCH

| Type | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Logistic Regression | 0.883333 | 0.92 | 0.821429 | 0.867925 |
| SVM | 0.866667 | 0.916667 | 0.785714 | 0.846154 |
| KNN | 0.833333 | 0.846154 | 0.785714 | 0.814815 |
| Gradient Boosting | 0.833333 | 0.846154 | 0.785714 | 0.814815 |

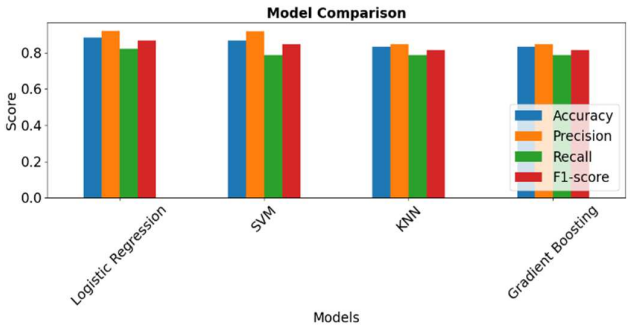


Fig. 5 Results of Model Evaluation with Random Search

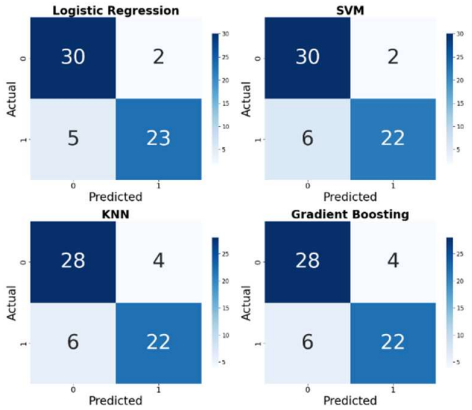


Fig. 6 Confusion Matrix Model with Random Search

Logistic Regression shows the best results among all models after tuning using Random Search. This model has an accuracy of 88.33% and the highest F1-score of 86.79%. Accuracy reached 92%, indicating the model's ability to minimize errors in positive predictions. The recall also increased to 82.14%, indicating that the model has a good balance between correctly detecting positives and reducing false negatives.

SVM maintained excellent performance, achieving 86.67% accuracy and an F1-score of 84.62%. The precision of this model reaches 91.67%, only slightly lower than that of Logistic Regression. However, the recall remained at 78.57%, indicating that SVM is slightly behind Logistic Regression in detecting all positive data. KNN showed stable performance with an accuracy of 83.33% and an F1-score of 81.48%. Precision and recall are 84.62% and 78.57%, respectively, indicating that this model provides good results, although not as optimal as Logistic Regression or SVM.

Gradient Boosting has similar results to KNN, with 83.33% accuracy, 84.62% precision, 78.57% recall, and 81.48% F1-score. While Gradient Boosting maintains balanced evaluation metrics, it doesn't show a significant advantage over other models.

Overall, after hyperparameter tuning with Random Search, Logistic Regression achieves the best results, with the highest accuracy and F1-score. The very high precision (92%) makes this model highly reliable for accurate predictions. Meanwhile, SVM remains a strong alternative with competitive evaluation metrics. On the other hand, KNN and Gradient Boosting show stable performance but are not as optimal as Logistic Regression and SVM.

After tuning the hyperparameters using the Bayesian Optimization method, the results of the model evaluation are shown in Table 13. The evaluation was carried out using four main metrics: accuracy, precision, recall, and F1-score.

TABLE XIV
RESULTS OF MODEL EVALUATION WITH BAYESIAN OPTIMIZATION TUNING HYPERPARAMETER

| Type | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Logistic Regression | 0.866667 | 0.884615 | 0.821429 | 0.851852 |
| SVM | 0.883333 | 0.92 | 0.821429 | 0.867925 |
| KNN | 0.85 | 0.88 | 0.785714 | 0.830189 |
| Gradient Boosting | 0.816667 | 0.84 | 0.75 | 0.792453 |

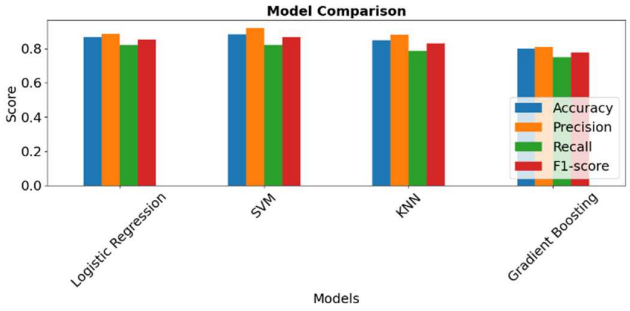


Fig. 7 Results of Model Evaluation with Bayesian Optimization

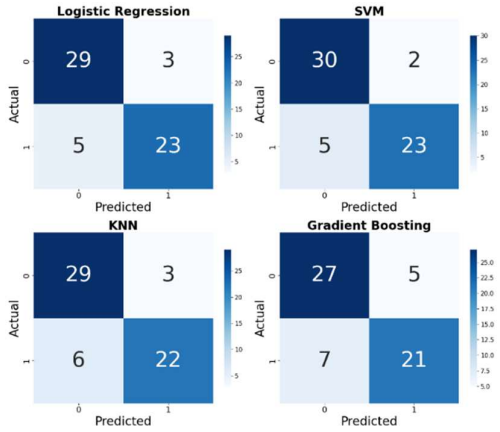


Fig. 8 Confusion Matrix Model with Bayesian Optimization

After hyperparameter tuning using Bayesian Optimization, the Logistic Regression model maintained consistent performance, achieving an accuracy of 86.67%. The precision model reached 88.46%, indicating a fairly good ability to minimize false-positive predictions. The recall of 82.14% shows the model's ability to detect positive data. An F1-score of 85.19% indicates a fairly good balance between precision and recall.

SVM achieved the best performance, with an accuracy of 88.33% and an F1-score of 86.79%. The precision of this model reaches 92%, the highest among all models. A recall of 82.14% also shows a good balance, making SVM a superior model in this scenario. KNN recorded an accuracy of 85% with an accuracy of 88%. A recall of 78.57% shows that this model is slightly lower in detecting positive data than Logistic Regression and SVM. An F1-score of 83.02% ranks the model third in overall performance.

Gradient Boosting has the lowest performance among all models, with an accuracy of 81.67%. Precision and recall were 84% and 75%, respectively, resulting in an F1-score of 79.25%. Although this model has lower performance, the relatively high precision indicates that it is more reliable in positive predictions than in recalls.

From the results of Bayesian Optimization tuning, SVM proved to be the best model for this case, with a superior combination of evaluation metrics. Logistic Regression and KNN are also competitive alternatives, while Gradient Boosting is suboptimal in this scenario. The results of the comparison of the accuracy of each model are shown in Table 10.

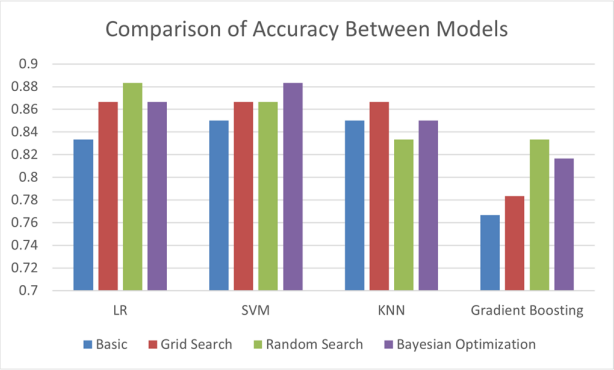


Fig. 9 Comparison of Accuracy Between Models

TABLE XV
RESULTS OF ACCURACY COMPARISON BETWEEN MODELS

| Type | Basic | Grid Search | Random Search | Bayesian Optimization |
|---|---|---|---|---|
| LR | 0.833333 | 0.866667 | **0.883333** | 0.866667 |
| SVM | 0.85 | 0.866667 | 0.866667 | **0.883333** |
| KNN | 0.85 | **0.866667** | 0.833333 | 0.85 |
| Gradient Boosting | 0.766667 | 0.783333 | **0.833333** | 0.816667 |

The Logistic Regression (LR) algorithm achieves the best results when using the Random Search hyperparameter optimization method, with an accuracy of 88.3333%. This shows that Random Search can explore hyperparameter spaces more effectively for simple models such as Logistic Regression than Grid Search and Bayesian Optimization. In the SVM model, Bayesian Optimization achieved the highest accuracy of 88.3333%. The probabilistic approach used in Bayesian Optimization proved effective for determining optimal hyperparameter combinations for margin-based models such as SVMs.

The Grid Search method provides the best performance on the KNN model with an accuracy of 86.6667%, slightly higher than Random Search and Bayesian Optimization. This suggests that systematic searches, such as Grid Search, may be more suitable for distance-based models, such as KNN, which have few key hyperparameters. Random Search achieves the highest accuracy of 83.3333% in the Gradient Boosting model. This suggests that random search allows for wider exploration of the hyperparameter space for ensemble models, compared to the more limited Grid Search or Bayesian Optimization, which prioritize efficiency.

The results of this study show that the hyperparameter tuning method significantly influences model performance. Bayesian Optimization and Random Search consistently show competitive results on various models, with efficiency advantages over Grid Search. However, the selection of tuning methods should be adjusted to the model's characteristics and computational needs, as shown by the results of the KNN model with Grid Search.

## IV. CONCLUSION

This study reveals key findings that the effectiveness of hyperparameter optimization techniques is highly dependent on the architecture of machine learning models. Comparative analysis shows that there is no universal optimal approach for all algorithms. Method selection must consider model characteristics and data complexity. Specifically, Random Search has proven effective for parametric models (Logistic Regression) and boosting-based models (Gradient Boosting), Bayesian Optimization excels for kernel-based models (SVM), while Grid Search is suitable for example-based models (KNN).

The main theoretical contribution of this research is the development of a conceptual framework that links the architectural characteristics of a model to the effectiveness of hyperparameter optimization methods. This framework provides a theoretical basis for understanding why certain optimization approaches are more suitable for specific model categories, a topic that has not been systematically explored in the literature. These findings enrich hyperparameter optimization theory by showing that optimization effectiveness depends not only on the parameter space's complexity but also on the model's inherent learning mechanisms.

These findings provide practical guidance for researchers and practitioners in selecting optimization strategies. Bayesian Optimization offers efficiency advantages for complex models, while Grid Search remains relevant for limited parameter spaces. Further research could develop hybrid approaches that leverage the strengths of each method or evaluate their effectiveness in more complex and diverse prediction scenarios.

This research has significant implications for advancing machine learning, providing a scientific basis for more accurate decision-making in hyperparameter optimization, ultimately improving model performance and computational efficiency across various machine learning applications.

## REFERENCES

[1] A. Supani, Alpian, Y. Andriani, and E. Oktavianti, "The tuning of machine learning models for the classification of cardiovascular disease," in *Proc. 6th Int. Conf. Comput. Informat. Eng. (IC2IE)*, 2023, pp. 381-384, doi: 10.1109/ic2ie60547.2023.10331436.

[2] A. R. Panda et al., "Heart disease prediction: A comparative analysis of machine learning algorithms," in *Proc. 2nd Int. Conf. Netw., Multimedia Inf. Technol. (NMITCON)*, 2024, pp. 1-6, doi:10.1109/nmitcon62075.2024.10698898.

[3] P. Lokhande, B. Sharma, and N. Ujjaliya, "An integrated approach in the management of cardiovascular diseases," *J. Ayurveda Integr. Med. Sci.*, vol. 9, no. 1, pp. 95-99, Feb. 2024, doi: 10.21760/jaims.9.1.12.

[4] A. Malim, "Heart disease prediction using machine learning," *Int. J. Sci. Res. Eng. Manage.*, vol. 08, no. 03, pp. 1-5, Mar. 2024, doi:10.55041/ijsrem29638.

[5] I. Ahmad et al., "New paradigms to break barriers in early cancer detection for improved prognosis and treatment outcomes," *J. Gene Med.*, vol. 26, no. 8, Aug. 2024, doi: 10.1002/jgm.3730.

[6] K. K. Baseer, S. B. A. Nas, S. Dharani, S. Sravani, P. Yashwanth, and P. Jyothirmai, "Medical diagnosis of human heart diseases with and without hyperparameter tuning through machine learning," in *Proc. 7th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, 2023, pp. 1-8, doi: 10.1109/iccmc56507.2023.10084156.

[7] M. K. Biddinika, A. Masitha, Herman, and V. A. N. Fatimah, "Machine learning techniques for heart disease prediction using a multi-algorithm approach," *JUITA: J. Inform.*, vol. 12, no. 2, p. 149, Nov. 2024, doi: 10.30595/juita.v12i2.24153.

[8] A. El-Ibrahimi et al., "Optimizing machine learning algorithms for heart disease classification and prediction," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 19, no. 15, pp. 61-76, Oct. 2023, doi:10.3991/ijoe.v19i15.42653.

[9] E. Maini, B. Venkateswarlu, B. Maini, and D. Marwaha, "Machine learning–based heart disease prediction system for Indian population: An exploratory study done in South India," *Med. J. Armed Forces India*, vol. 77, no. 3, pp. 302-311, Jul. 2021, doi:10.1016/j.mjafi.2020.10.013.

[10] Y. Boer, L. Valencia, M. R. Setiadi, K. E. Setiawan, and M. F. Hasani, "Classification of heart disease: Comparative analysis using KNN, random forest, Gaussian naive bayes, XGBoost, SVM, decision tree, and logistic regression," in *Proc. 5th Int. Conf. Cybern. Intell. Syst. (ICORIS)*, 2023, pp. 1-5, doi: 10.1109/icoris60118.2023.10352195.

[11] V. Sawan, K. Jugnu, D. P. Roy, and R. Pandey, "Comparative assessment of machine learning algorithms for heart disease prediction," *Int. J. Eng. Appl. Sci. Technol.*, vol. 8, no. 3, pp. 121-125, Jul. 2023, doi: 10.33564/ijeast.2023.v08i03.016.

[12] T. Rustagi and M. Vijarania, "Implementation of ML techniques for heart prediction in healthcare," in *Mach. Learn. Multimedia*. Boca Raton, FL, USA: CRC Press, 2024, pp. 129-138. doi:10.1201/9781003477280-11.

[13] H. Azis, "Assessing the performance of logistic regression in heart disease detection through 5-fold cross-validation," *Int. J. Artif. Intell. Med. Issues*, vol. 2, no. 1, pp. 1-11, May 2024, doi:10.56705/ijaimi.v2i1.137.

[14] A. S. Osei-Nkwantabisa and R. Ntumy, "Classification and Prediction of Heart Diseases using Machine Learning Algorithms," *arXiv Prepr. arXiv2409.03697*, 2024, doi: 10.48550/arXiv.2409.03697.

[15] C. J. C. Kusuma and Khairunnisa, "Optimizing bidirectional LSTM for energy consumption prediction using chaotic particle swarm optimization and hyperparameter tuning," *Int. J. Artif. Intell. Inform.*, vol. 2, no. 2, pp. 57-60, Jul. 2024, doi: 10.33292/ijarlit.v2i2.37.

[16] F. Al Anshori and S. Pidgeon, "Prediction of Euro to US dollar exchange rate using CNN method with grid optimization," *Int. J. Artif. Intell. Inform.*, vol. 3, no. 2, pp. 27-44, Jul. 2025, doi:10.33292/ijarlit.v3i2.45.

[17] E. Yilmaz and E. Tcrol, "Comparison of LSTM and LSTM with grid search optimization for stock price prediction of Saudi Arabian oil company (Aramco)," *Int. J. Artif. Intell. Inform.*, vol. 3, no. 2, pp. 52-59, Jul. 2025, doi: 10.33292/ijarlit.v3i2.47.

[18] S. Sanhatham, "Stock price prediction of Thai oil public company limited (TOP.BK) using LSTM model with grid search hyperparameter optimization," *Int. J. Artif. Intell. Inform.*, vol. 2, no. 1, pp. 33-40, Jan. 2024, doi: 10.33292/ijarlit.v2i1.34.

[19] A. Flavia and C. Mio, "Performance comparison of standard LSTM and LSTM with random search optimization for spark New Zealand limited stock price prediction," *Int. J. Artif. Intell. Inform.*, vol. 3, no. 2, pp. 60-66, Jul. 2025, doi: 10.33292/ijarlit.v3i2.48.

[20] F. Agustin and P. D. Melin, "Comparison of GRU and CNN methods for predicting the exchange rate of Argentine peso (ARS) against US dollar (USD)," *Int. J. Artif. Intell. Inform.*, vol. 2, no. 1, pp. 9-16, Jan. 2024, doi: 10.33292/ijarlit.v2i1.31.

[21] G. Tzoulis, "Harnessing convolutional neural networks for accurate stock price prediction: A case study of Hellenic telecommunications organization (HTO.AT)," *Int. J. Artif. Intell. Inform.*, vol. 2, no. 2, pp. 66-72, Jul. 2024, doi: 10.33292/ijarlit.v2i2.39.

[22] D. Bohovic, "Comparison of LSTM and GRU methods for predicting gold exchange rate against US dollar," *Int. J. Artif. Intell. Inform.*, vol. 3, no. 1, pp. 24-29, Jan. 2025, doi: 10.33292/ijarlit.v3i1.43.

[23] M. Gabrielzinho and G. Moraes, "Petrobras stock price prediction using deep learning approach: Performance comparison of CNN and CNN-GRU methods," *Int. J. Artif. Intell. Inform.*, vol. 3, no. 1, pp. 30-36, Jan. 2025, doi: 10.33292/ijarlit.v3i1.44.

[24] Dzhumashev and M. Aybek, "Prediction of the exchange rate of the Russian ruble (RUB) against the United States dollar (USD): Performance comparison of LSTM and CNN models," *Int. J. Artif. Intell. Inform.*, vol. 2, no. 1, pp. 25-32, Jan. 2024, doi:10.33292/ijarlit.v2i1.33.

[25] G. N. Ahamad et al., "Influence of optimal hyperparameters on the performance of machine learning algorithms for predicting heart disease," *Processes*, vol. 11, no. 3, p. 734, Mar. 2023, doi:10.3390/pr11030734.

[26] L. Franceschi *et al.*, "Hyperparameter Optimization in Machine Learning," *arXiv Prepr. arXiv2410.22854*, 2024, doi:10.48550/arXiv.2410.22854.

[27] O. P. Barus, N. Phan, A. E. Widjaja, J. J. Pangaribuan, and Romindo, "Heart disease classification using decision trees," in *Proc. 3rd Int. Conf. Creative Commun. Innov. Technol. (ICCIT)*, 2024, pp. 1-6, doi:10.1109/iccit62134.2024.10701238.

[28] A. A. Mohsen, N. Kharroubi, T. Alrashahy, and S. Noaman, "Classification and diagnosis of heart disease using machine learning," Res. Square Platform LLC, Preprint, Mar. 4, 2024. doi:10.21203/rs.3.rs-3985932/v2.

[29] M. Alfonse, "Classification of heart disease using multiple classifiers," *Rev. Comput. Eng. Stud.*, vol. 5, no. 3, pp. 45-49, Sep. 2019, doi: 10.18280/rces.050301.

[30] M. I. Habelalmateen, A. H. A. Hussein, S. Jagadish, K. Ramakrishna, and J. Velumani, "A heart disease classification using an improved particle swarm optimization algorithm with support vector machine classifier," in *Proc. Int. Conf. Integr. Intell. Commun. Syst. (ICIICS)*, 2023, pp. 1-5, doi: 10.1109/iciics59993.2023.10421318.

[31] R. Balamurugan, S. Ratheesh, and Y. M. Venila, "Classification of heart disease using adaptive Harris hawk optimization-based clustering algorithm and enhanced deep genetic algorithm," *Soft Comput.*, vol. 26, no. 5, pp. 2357-2373, Nov. 2021, doi:10.1007/s00500-021-06536-0.

[32] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Heart Disease [Dataset]." UCI Machine Learning Repository, 1989. doi:10.24432/C52P4X.

[33] L. O. Joel, W. Doorsamy, and B. S. Paul, "A comparative study of imputation techniques for missing values in healthcare diagnostic datasets," *Int. J. Data Sci. Anal.*, vol. 20, no. 7, pp. 6357-6373, Jun. 2025, doi: 10.1007/s41060-025-00825-9.

[34] A. V. Pindiyan and Pramila R.M, "Machine learning-based imputation techniques analysis and study," in *Proc. Int. Conf. Electr. Electron. Comput. Technol. (ICEECT)*, 2024, pp. 1-6, doi:10.1109/iceect61758.2024.10739330.

[35] S. Prakash, S. Singh, and A. Mankar, "Bridging data gaps: A comparative study of different imputation methods for numeric datasets," in *Proc. Int. Conf. Data Sci. Netw. Secur. (ICDSNS)*, 2024, pp. 1-7, doi: 10.1109/icdsns62112.2024.10691111.

[36] B. G. Chepino et al., "Effect of minmax normalization on ORB data for improved ANN accuracy," *J. Electr. Eng., Energy, Inf. Technol. (J3EIT)*, vol. 11, no. 2, p. 29, Aug. 2023, doi:10.26418/j3eit.v11i2.68689.

[37] M. Shantal, Z. Othman, and A. A. Bakar, "A novel approach for data feature weighting using correlation coefficients and min–max normalization," *Symmetry*, vol. 15, no. 12, p. 2185, Dec. 2023, doi:10.3390/sym15122185.

[38] S. Hadianti, "Optimization of the machine learning approach using Optuna in heart disease prediction," *J. Med. Informat. Technol.*, vol. 1, no. 3, pp. 59-64, Sep. 2023, doi: 10.37034/medinftech.v1i3.15.

[39] A. Zaidi and A. S. M. Al Luhayb, "Two statistical approaches to justify the use of the logistic function in binary logistic regression," *Math. Problems Eng.*, vol. 2023, no. 1, Jan. 2023, doi:10.1155/2023/5525675.

[40] G. Charizanos, H. Demirhan, and D. İçen, "Binary classification with fuzzy logistic regression under class imbalance and complete separation in clinical studies," *BMC Med. Res. Methodol.*, vol. 24, no. 1, Jul. 2024, doi: 10.1186/s12874-024-02270-x.

[41] A. Testas, "Support vector machine classification with Pandas, Scikit-Learn, and PySpark," in *Distrib. Mach. Learn. PySpark*. Berkeley, CA, USA: Apress, 2023, pp. 259-280. doi: 10.1007/978-1-4842-9751-3_10.

[42] T. Ke et al., "A general maximal margin hyper-sphere SVM for multi-class classification," *Expert Syst. Appl.*, vol. 237, p. 121647, Mar. 2024, doi: 10.1016/j.eswa.2023.121647.

[43] P. Chen and T. Wen, "Margin maximization model of text classification based on support vector machines," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2006, pp. 3514-3518. doi:10.1109/icmlc.2006.258543.

[44] Y. Wang, W. Liao, H. Shen, Z. Jiang, and J. Zhou, "Some notes on the basic concepts of support vector machines," *J. Comput. Sci.*, vol. 82, p. 102390, Oct. 2024, doi: 10.1016/j.jocs.2024.102390.

[45] S. K. Mohamed, N. A. Sakr, and N. A. Hikal, "A review of breast cancer classification and detection techniques," *Int. J. Adv. Sci. Comput. Eng.*, vol. 3, no. 3, pp. 128-139, Oct. 2021, doi:10.30630/ijasce.3.3.55.

[46] Amelia, Marsani Asfi, and Rifqi Fahrudin, "Implementation of K-nearest neighbor method for selection of new employee candidates (Case study: CV. Syntax Corporation Indonesia)," *Eduvest - J. Universal Stud.*, vol. 4, no. 7, pp. 5742-5754, Jul. 2024, doi:10.59188/eduvest.v4i7.1305.

[47] R. K. Halder et al., "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *J. Big Data*, vol. 11, no. 1, Aug. 2024, doi: 10.1186/s40537-024-00973-y.

[48] N. Islam, H. Noor, and D. M. Farid, "A novel ensemble K-nearest neighbours classifier with attribute bagging," in *Commun. Comput. Inf. Sci.*. Singapore: Springer Nature Singapore, 2023, pp. 262-276. doi:10.1007/978-981-99-7666-9_22.

[49] K. Y. C. Lin, "Optimizing variable selection and neighbourhood size in the K-nearest neighbour algorithm," *Comput. Ind. Eng.*, vol. 191, p. 110142, May 2024, doi: 10.1016/j.cie.2024.110142.

[50] H.-H. Chen, "Understanding gradient boosting classifier: Training, prediction, and the role of $\gamma_j$," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2410.05623.

[51] A. Ganiyu et al., "Classification algorithms using ensemble methods," in *Proc. 11th Int. Conf. Future Internet Things Cloud (FiCloud)*, 2024, pp. 168-175, doi: 10.1109/ficloud62933.2024.00033.

[52] H. Zhang, W. Wu, C. Yang, and B. Li, "A tree regression algorithm based on incremental gradient boosting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2024, pp. 1-7, doi:10.1109/ijcnn60899.2024.10650730.

[53] A. Y. Yıldız and A. Kalayci, "Gradient boosting decision trees on medical diagnosis over tabular data," in *Proc. IEEE Int. Conf. AI Data Anal. (ICAD)*, 2025, pp. 1-8, doi: 10.1109/icad65464.2025.11114069.

[54] E. A. U. Malahina et al., "A grid-search method approach for hyperparameter evaluation and optimization on teachable machine accuracy: A case study of sample size variation," *J. Appl. Data Sci.*, vol. 5, no. 3, pp. 1008-1025, Sep. 2024, doi: 10.47738/jads.v5i3.290.

[55] Q. Lv, "Hyperparameter tuning of GDBT models for prediction of heart disease," in *Proc. Int. Conf. Electron. Inf. Eng. Comput. Sci. (EIECS)*, 2023, p. 127. doi: 10.1117/12.2668449.

[56] A. Esmaeili, Z. Ghorrati, and E. T. Matson, "Agent-based collaborative random search for hyperparameter tuning and global function optimization," *Systems*, vol. 11, no. 5, p. 228, May 2023, doi:10.3390/systems11050228.

[57] M. Fajri and A. Primajaya, "Komparasi teknik hyperparameter optimization pada SVM untuk permasalahan klasifikasi dengan menggunakan grid search dan random search," *J. Appl. Informat. Comput.*, vol. 7, no. 1, pp. 14-19, Jan. 2023, doi:10.30871/jaic.v7i1.5004.

[58] G. Onorato, "Bayesian Optimization for Hyperparameters Tuning in Neural Networks," *arXiv Prepr. arXiv2410.21886*, 2024, doi:10.48550/arXiv.2410.21886.

[59] A. I. Alvi, J. Janssen, D. Khatamsaz, D. Perez, D. Allaire, and R. Arróyave, "Hierarchical Gaussian process-based Bayesian optimization for materials discovery in high entropy alloy spaces," *Acta Mater.*, vol. 289, p. 120908, May 2025, doi:10.1016/j.actamat.2025.120908.

[60] L. I. Berlina, U. S. Aesyi, and Kharisma, "Community perspective analysis of Yogyakarta special region using K-means algorithm," *Emerg. Inf. Sci. Technol.*, vol. 5, no. 2, pp. 66-73, Nov. 2024, doi: 10.18196/eist.v5i2.24729.

[61] F. Fitrianto, E. Rouza, and Basorudin, "Implementation of transfer learning in cat breed detection using web-based convolutional neural network (CNN)," *Emerg. Inf. Sci. Technol.*, vol. 6, no. 1, pp. 43-53, May 2025, doi: 10.18196/eist.v6i1.27104.

[62] I. Prabasari, A. Zuhri, S. Riyadi, T. K. Hariadi, and N. A. Utama, "Analysis of cross validation on classification of mangosteen maturity stages using support vector machine," *Emerg. Inf. Sci. Technol.*, vol. 5, no. 1, May 2024, doi: 10.18196/eist.v5i1.22359.

[63] N. A. Utama, W. I. Triyani, S. Riyadi, and C. Damarjati, "Discrete curvelet transform feature extraction for mangosteen fruit surface damage detection," *Emerg. Inf. Sci. Technol.*, vol. 5, no. 1, Jun. 2024, doi: 10.18196/eist.v5i1.22602.

[64] C. Damarjati, S. Riyadi, and R. Irawan, "Classification of student understanding on Covid-19 booster vaccine using machine learning," *Emerg. Inf. Sci. Technol.*, vol. 3, no. 2, pp. 78-84, Nov. 2022, doi: 10.18196/eist.v3i2.18680.

[65] D. Regine and A. Zabarnyi, "Comparison of CNN-LSTM hybrid and CNN methods for Ethereum (ETH) to US dollar (USD) exchange rate prediction," *Int. J. Artif. Intell. Inform.*, vol. 3, no. 1, pp. 16-23, Jan. 2025, doi: 10.33292/ijarlit.v3i1.42.

[66] S. Mostafa, "Comparison of CNN, CNN-GRU, and GRU models for prediction of Hryvnia (Ukraine) exchange rate against US dollar," *Int. J. Artif. Intell. Inform.*, vol. 3, no. 2, pp. 67-73, Jul. 2025, doi:10.33292/ijarlit.v3i2.49.

[67] T. L. Aníbal and R. Okanlawon, "Stock price prediction of ReconAfrica (RECAF) using gated recurrent unit (GRU): Analysis and implications for investment decisions," *Int. J. Artif. Intell. Inform.*, vol. 2, no. 2, pp. 41-46, Mar. 2025, doi: 10.33292/ijarlit.v2i2.35.

[68] J. Muñoz and K. Castaño, "Enhancing stock price predictions: Leveraging LSTM for accurate forecasting of Ecopetrol's stock performance," *Int. J. Artif. Intell. Inform.*, vol. 2, no. 2, pp. 47-55, Jul. 2024, doi: 10.33292/ijarlit.v2i2.36.

[69] A. Kurnianti, P. Pahlevi, and I. Mufidah, "Recommendation system for prospective bride and groom using cosine similarity algorithm," *Emerg. Inf. Sci. Technol.*, vol. 4, no. 1, pp. 8-15, May 2023, doi: 10.18196/eist.v4i1.18683.

[70] H. Setyawan, L. M. Azizah, and A. Y. Pradani, "Sentiment analysis of public responses on Indonesia government using Naïve Bayes and support vector machine," *Emerg. Inf. Sci. Technol.*, vol. 4, no. 1, pp. 1-7, May 2023, doi: 10.18196/eist.v4i1.18681.

[71] A. Rakhmadi and N. D. Rahmawati, "Implementation of simple additive weighting to decide a fund proposal," *Emerg. Inf. Sci. Technol.*, vol. 4, no. 2, Dec. 2023, doi: 10.18196/eist.v4i2.20741.

[72] R. Y. Hartanta, Asroni, S. Riyadi, and Jeckson, "Analysis and visualization of high school student achievement data using decision tree and cross-validation in Rapidminer," *Emerg. Inf. Sci. Technol.*, vol. 4, no. 2, Dec. 2023, doi: 10.18196/eist.v4i2.20731.

[73] A. N. Adriansyah and S. Riyadi, "Classification of duration in global terrorism using ResNet," *Emerg. Inf. Sci. Technol.*, vol. 5, no. 2, pp. 74-79, Nov. 2024, doi: 10.18196/eist.v5i2.24756.

[74] S. Riyadi, C. Damarjati, S. Khotimah, and A. J. Ishak, "Optimization of the VGG deep learning model performance for Covid-19 detection using CT-scan images," *Regist.: J. Ilm. Teknol. Sist. Inform.*, vol. 10, no. 1, pp. 91-101, Jun. 2024, doi: 10.26594/register.v10i1.3598.

[75] S. Riyadi, "Classification of political party conflicts and their mediation using modified recurrent convolutional neural network," *J. Appl. Data Sci.*, vol. 6, no. 1, pp. 416-425, Jan. 2024, doi:10.47738/jads.v6i1.513.

[76] S. Riyadi, A. D. Andriyani, and S. N. Sulaiman, "Improving hate speech detection using double-layers hybrid CNN-RNN model on imbalanced dataset," *IEEE Access*, vol. 12, pp. 159660-159668, 2024, doi: 10.1109/access.2024.3487433.

[77] J.-D. Wang and C. O. N. Susanto, "Traffic flow prediction with heterogenous data using a hybrid CNN-LSTM model," *Comput., Mater. Continua*, vol. 76, no. 3, pp. 3097-3112, 2023, doi:10.32604/cmc.2023.040914.