Scientific Journal of Informatics
Vol. 13, No. 4, November 2026
p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Hyperparameter Optimization of Machine Learning Models Using Differential Evolution for Heart Failure Mortality Prediction

Royan Farid Fathurrahman[1*], Abdul Syukur[2], Nova Rijati[3]
[1,2,3]Master's Program in Informatics Engineering, Faculty of Computer Science,
Universitas Dian Nuswantoro, Indonesia

Abstract.
Purpose: Heart failure is a cardiovascular disease associated with high morbidity and mortality rates. Various machine learning models have been increasingly applied to predict mortality among patients with heart failure, but their predictive performance can be influenced by hyperparameter configuration. This study aims to optimize hyperparameters using the Differential Evolution (DE) algorithm to improve predictive performance.
Methods: This research used the Heart Failure Clinical Records Dataset, which consists of 299 patient data with 13 clinical features. The dataset is first split into a training set and a testing set, with the testing set kept independent throughout the modeling process. Within the training process, feature selection is performed using Chi-square-based SelectKBest, class imbalance is addressed using the Synthetic Minority Oversampling Technique (SMOTE), and feature scaling is performed using StandardScaler. These preprocessing steps are incorporated into the cross-validation pipeline to prevent information leakage. Next, nine machine learning algorithms were optimized using Differential Evolution with an accuracy based objective function and 10-fold cross-validation.
Result: Experimental results show that Differential Evolution produced model-specific effects, with accuracy improvements observed in six of the nine evaluated machine learning models compared with their baseline configurations. Random Forest (RF), Support Vector Machine (SVM), and Extra Trees Classifier (ETC) achieved the highest accuracy of 0.8500, while RF obtained the highest F1-score and ROC-AUC of 0.7692 and 0.8922, respectively. The exact McNemar test showed that none of the baseline-versus-DE differences reached statistical significance at the 0.05 level.
Novelty: This research proposes a mortality classification framework for heart failure patients that integrates SelectKBest, SMOTE, and Differential Evolution for hyperparameter optimization across nine machine learning models. The framework enables a systematic evaluation of the model specific effects of Differential Evolution on heart failure mortality classification performance.

Keywords: Heart failure, Machine learning, Hyperparameter optimization, Differential evolution, Mortality classification
Received July 2026 / **Revised** September 2026 / **Accepted** October 2026

This work is licensed under a Creative Commons Attribution 4.0 International License.

INTRODUCTION

The heart is a vital muscle organ located below and slightly to the left of the breastbone. Along with its blood vessels, it is susceptible to a variety of disorders known as cardiovascular disease (CVD) [1]. CVD includes a variety of disorders such as peripheral artery disease, coronary heart disease, stroke, and heart failure [2]. WHO reports that cardiovascular diseases (CVDs) are the leading cause of death globally, with an estimated 19.8 million deaths in 2022, representing approximately 32% of all global deaths [3]. Furthermore, the World Heart Federation projects that the number of CVD deaths will increase from an estimated 18.9 million in 2020 to more than 22.2 million in 2030 and 32.3 million in 2050 [4].

Among various cardiovascular disorders, heart failure is a significant cause of rising mortality, morbidity, and economic burden [5]. Globally, more than 23 million people have heart failure [6], and the ESC report puts the number at around 26 million cases with 3.6 million new diagnoses each year [7], [8]. In the UK, hospital admissions for heart failure increased from 65,025 in 2013/14 to 86,474 in 2018/19, a 33% increase according to the British Heart Foundation [9]. Meanwhile, China's 2022 cardiovascular health report showed 8.9 million heart failure patients with a mortality rate due to acute heart failure approaching 12.7% [10].

*Corresponding author.
Email addresses: p31202402648@mhs.dinus.ac.id (Fathurrahman)
DOI: 10.15294/sji.v13i4.59449

Heart failure is an advanced clinical syndrome of various heart diseases, characterized by ventricular dysfunction or a reduced ability of the ventricle to pump blood effectively so that the blood supply for metabolic needs is reduced [11]. With complex symptoms, an accurate diagnosis is crucial for assessing severity and determining appropriate therapy. Furthermore, the use of mortality predictive models has been shown to help identify key risk factors, allowing physicians to identify patients requiring intensive monitoring, specialized therapy, or palliative care in an intensive care unit [12].

Heart failure diagnosis is generally made using two approaches. Conventional methods rely on clinical patterns, medical tests, symptoms, and patient history, but have limitations such as the risk of misinterpretation and nonspecific symptoms, which can delay diagnosis and reduce predictive accuracy [11], [13], [14]. In contrast, machine learning-based approaches leverage relevant data to improve accuracy and support clinical decision-making [13]. Various studies have shown that machine learning models are able to provide better predictive performance than traditional methods in various medical cases [15], [16], including in cases of heart failure [17], [18]. However, its performance depends on and is heavily influenced by the hyperparameter configuration used. Hyperparameters are external parameters that control the model's learning process and must be determined before training begins. If not specified, the model will use the default configuration which may not necessarily produce optimal performance because the best configuration may vary, depending on the dataset used and the problems domain [15], [19], [20]. Therefore, a hyperparameter optimization process is required to obtain a configuration that is capable of producing better prediction performance.

Various studies have applied machine learning methods to predict survival outcomes among patients with heart failure. Early studies have primarily focused on feature selection and class imbalance handling to improve classification performance. For instance, Chicco and Jurman [21] demonstrated that machine learning models could achieve a ROC-AUC of approximately 0.80 using only two key features, namely ejection fraction and serum creatinine. Meanwhile, Ishaq et al [22] integrated feature selection and data imbalance handling using SMOTE, so that the Extra Trees Classifier model was able to achieve 92.62% accuracy and improve minority class detection. These studies demonstrate the importance of relevant feature identification and appropriate data preparation in heart failure survival prediction.

Recent studies have subsequently investigated hyperparameter optimization to improve the configuration of machine learning models for heart failure prediction. These approaches include metaheuristic optimization, reinforcement learning, and conventional search strategies. Ahmed et al [23] applied Adaptive Inertia Weight Particle Swarm Optimization (AIW-PSO) to optimize machine learning models, while Ribino et al [24] proposed a Reinforcement Learning Hyperparameter Optimization (RLHPO) for heart failure survival prediction. Conventional approaches, including Grid Search, Random Search, and Bayesian Optimization, have also been investigated [25]. These studies demonstrate the potential of hyperparameter optimization to improve machine learning model configuration, but they differ in optimization mechanisms, model settings, and experimental scope.

Differential Evolution (DE) is another population-based optimization approach that has been applied to machine learning classification problems. DE iteratively generates and evaluates candidate solutions through evolutionary operations to explore the optimization space [26]. Its relatively simple mathematical structure and limited number of control parameters also facilitate its implementation in optimization problems [27], [28]. Previous research has demonstrated the applicability of DE in binary heart disease classification [26], indicating its potential as an optimization approach for machine learning-based classification tasks.

Despite the progress demonstrated by previous studies, several aspects remain open for further investigation. First, studies focusing on feature selection and class imbalance handling [21], [22] do not primarily investigate systematic hyperparameter optimization as a central component of the modeling framework. Second, existing hyperparameter optimization studies [23], [24], [25] employ different optimization strategies and experimental settings, making it necessary to further investigate an optimization approach that can be systematically applied across multiple machine learning classifiers. Third, although Differential Evolution has been applied to binary heart disease classification [26], its systematic use for hyperparameter optimization across multiple machine learning classifiers in heart failure outcome prediction remains limited in the reviewed literature. These gaps motivate the investigation of a unified

Differential Evolution-based hyperparameter optimization framework for heart failure mortality classification.

Therefore, this study investigates a Differential Evolution-based hyperparameter optimization framework across nine machine learning models for heart failure mortality classification. The study aims to systematically evaluate the effect of Differential Evolution-based hyperparameter optimization on the predictive performance of different machine learning classifiers. By evaluating multiple classifiers within a unified optimization framework, this study provides a comparative analysis of the applicability of Differential Evolution for hyperparameter optimization in heart failure mortality classification.

METHODS

This study proposes the application of hyperparameter optimization using the Differential Evolution (DE) algorithm to machine learning models for classifying heart failure mortality. The research framework is systematically designed, encompassing data preparation, model development, hyperparameter optimization, and performance evaluation. All procedures were conducted with strict adherence to the principle of test data independence, ensuring that evaluation results objectively reflect model performance. The overall research workflow is visualized in Figure 1, illustrating the relationships between stages, from data input to final evaluation.

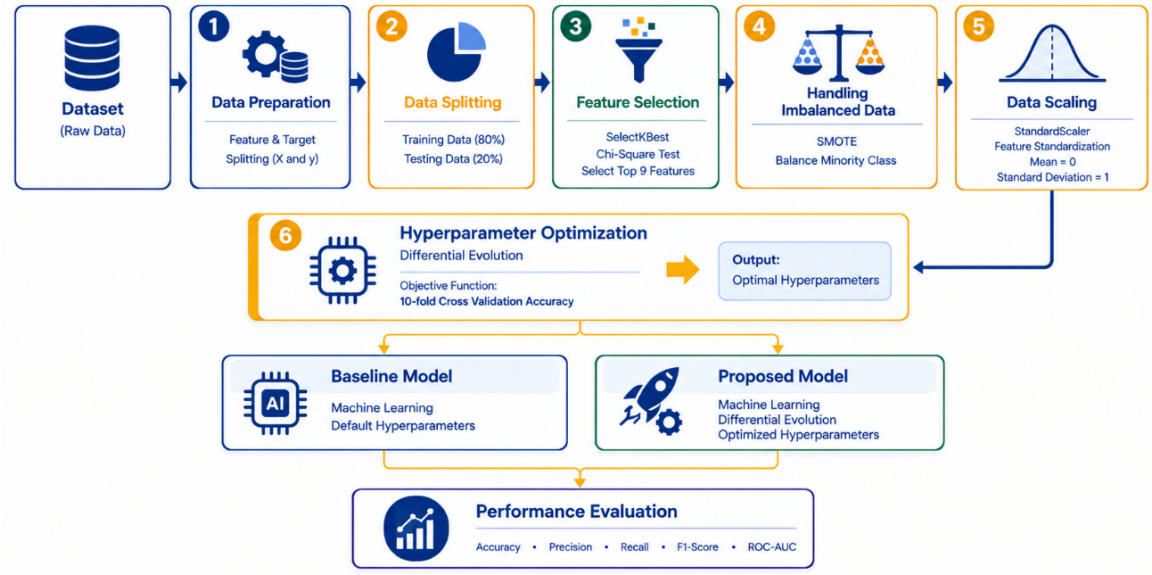


Figure 1. Proposed model

Figure 1 illustrates the research workflow, which comprises six main stages. The process begins with data preparation, involving the separation of features and target variables, as well as the splitting of data into training and testing sets. Subsequently, the training data undergoes feature selection using SelectKBest with the Chi-square test, class imbalance handling via SMOTE, and feature scaling using StandardScaler. The processed data is then utilized in a Differential Evolution (DE)-based hyperparameter optimization stage to determine the optimal configuration for each machine learning model. Following this, two modeling scenarios are evaluated: a baseline model using default hyperparameters and a proposed model using hyperparameters derived from DE optimization. The final stage involves evaluating the performance of both models using accuracy, precision, recall, F1-score, and ROC-AUC metrics to assess classification effectiveness.

Dataset

This research uses a dataset available on the Kaggle platform named Heart Failure Clinical Records, with 299 entries of heart failure patients, consisting of 203 living patients and 96 deceased patients, and includes 13 clinical variables used in the analysis. This study specifically focuses on efforts to predict patient survival status during the clinical follow-up period based on the target variable, namely DEATH_EVENT.

Table 1. Dataset features

No	Feature	Description
1	age	patient age
2	anaemia	decreased levels of erythrocytes or hemoglobin in the blood
3	creatinine_phosphokinase	CPK enzyme levels in the patient's bloodstream
4	diabetes	diabetes patient
5	ejection_fraction	the relative amount of blood ejected by the heart during the contraction phase of the heart muscle
6	high_blood_pressure	if the patient has a history of hypertension
7	platelets	platelet levels circulating in the circulatory system
8	serum_creatinine	the amount of creatinine detected in blood serum
9	serum_sodium	the amount of serum sodium detected in blood serum
10	sex	patient's gender, whether male or female
11	smoking	whether the patient is a smoker or not
12	time	observation period used to assess clinical progress or outcomes
13	DEATH_EVENT	if the patient dies during the clinical monitoring period

Table 1 lists 13 features from the dataset along with their descriptions. These features include patient demographic data such as age and gender, as well as health indicators such as anemia, diabetes, hypertension, and smoking status. Additionally, there are biochemical and physiological parameters, namely CPK enzyme levels, platelet count, serum sodium, ejection fraction, and serum creatinine. The last two features are related to time and clinical outcomes: time as the observation period and DEATH_EVENT as a binary variable representing the patient's death status during the follow-up period. The DEATH_EVENT feature serves as the target variable for binary mortality classification in patients with heart failure.

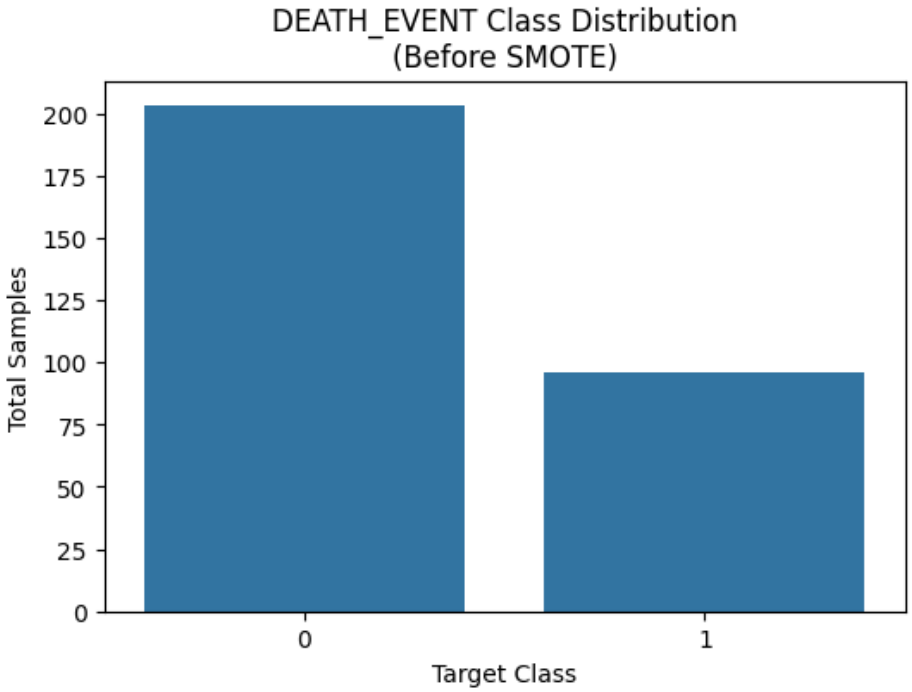


Figure 2. Data visualization before SMOTE

However, this dataset still shows class imbalance in the target data, DEATH_EVENT. The visualization in Figure 2 shows a significant imbalance in the class data within the dataset. Class 0, which represents surviving patients, has the largest number of data points and dominates the dataset compared to class 1, which represents deceased patients. Therefore, to address this imbalance, special handling is required.

Data Preparation

The data preparation phase begins by splitting the dataset into predictor variables (X) and a target variable (y). The predictor variables consist of clinical features used as input for the machine learning model, while DEATH_EVENT serves as the binary target variable indicating whether a patient died during the

monitoring period. Thus, X is formed by excluding DEATH_EVENT from the dataset, whereas y contains the DEATH_EVENT labels. This separation establishes the input features and classification target before the data are divided into training and testing sets.

Data Splitting

Next, the dataset will be split into two parts, consisting of a training set and a testing set. In this study, the research data were split into a training set with a proportion of 80% and a testing set with a proportion of 20% to evaluate the model's performance. The training set is used for feature selection, class imbalance handling, data scaling, model training, and hyperparameter optimization, while the testing set is kept completely independent throughout these processes. The data splitting is performed before any preprocessing step to prevent information from the testing set from being incorporated into the model development process. This separation ensures that the testing data remain unseen during training and can therefore provide an unbiased assessment of the model's generalization performance.

Feature Selection

Feature selection is performed using the SelectKBest method with the Chi-square statistical test. SelectKBest evaluates each predictor based on its statistical association with the DEATH_EVENT target variable and assigns a Chi-square score to each feature. The features are then ranked according to their scores, and the k features with the highest scores are retained. In this study, k is set to 9, meaning that nine features are selected from the available predictors in each training fold. The Chi-square test evaluates the association between each predictor and the target by comparing the observed and expected frequencies across the DEATH_EVENT classes. A higher Chi-square score indicates a stronger statistical association between a feature and the target variable, making the feature more likely to be selected by SelectKBest. The feature selection process is illustrated in Table 2.

Table 2. Feature selection procedure using SelectKBest and Chi-square test

Stage	Process	Description
1	Input predictors	All available clinical features are provided as candidate predictors, while DEATH_EVENT is used as the binary target variable.
2	Chi-square test	The association between each predictor and DEATH_EVENT is evaluated using the Chi-square statistical test.
3	Feature scoring	Each predictor receives a Chi-square score based on its association with the target variable.
4	Feature ranking	The candidate features are ranked according to their Chi-square scores from the highest to the lowest.
5	SelectKBest	SelectKBest retains the top $k = 9$ features with the highest Chi-square scores.
6	Subsequent processing	The selected features are passed to SMOTE, StandardScaler, and the classifier within the pipeline.
7	Cross-validation	The feature selection procedure is fitted independently within the training portion of each cross-validation fold to prevent data leakage.

To prevent data leakage, SelectKBest is incorporated into the machine learning pipeline and applied within the cross-validation process. For each fold, the Chi-square scores are calculated using only the training portion of that fold, and the nine highest-scoring features are selected. The selected features are then passed to the subsequent SMOTE, scaling, and classification steps. Thus, the validation portion of each fold and the independent testing set do not contribute to the feature selection process. The application of feature selection aims to reduce data complexity, eliminate less relevant features, and increase the efficiency of the model learning process [29], [30].

Handling Imbalance Data

After the feature selection process, class distribution imbalance is addressed using the Synthetic Minority Oversampling Technique (SMOTE). Additional samples are synthetically generated for the minority class based on the proximity between samples using the k-nearest neighbors approach, so that the new instances still resemble the original data, as they are formed from the same features [22]. The application of SMOTE aims to balance the class distribution so that the characteristics of both classes can be better learned by the model and reduce the tendency for bias towards the majority class [23].

Data Scaling

The next step involves data standardization using StandardScaler from the Sci-Kit Learn library because it is very effective and easy to use [31]. This process standardizes the values of each feature so that they are

distributed with a standard deviation of one and mean of zero [32]. Standardization is implemented because some machine learning models are sensitive to scale differences between features, so scale equalization can improve the stability of the learning process and computational efficiency during model training [32]. The scaling parameters are fitted using only the training data and subsequently applied to the corresponding data used for model evaluation. This ensures that information from the testing data is not used to determine the scaling parameters and helps maintain a consistent feature scale during model development and evaluation.

Hyperparameter Optimization

The hyperparameter optimization process is carried out using the Differential Evolution (DE) algorithm approach. In this research, each individual in the population is viewed as a candidate solution representing a hyperparameter combination of an optimized machine learning model. The optimization process begins with random initial population generation within a predetermined search space. Next, candidate solutions are iteratively updated through three main mechanisms, starting with mutation, then crossover, and finally selection. In the mutation stage, DE algorithms generally feature five mutation strategies, comprising rand/1 (DE/rand/1), rand/2 (DE/rand/2), best/1 (DE/best/1), best/2 (DE/best/2), and current-to-best/1 (DE/current-to-best/1) [27]. However, best/1 and rand/1 are commonly used strategies, with the following mathematical formula.

best/1

$$Y_i^t = X_{best}^t + F(X_{r_1}^t - X_{r_2}^t) \quad (1)$$

rand/1

$$Y_i^t = X_{r_1}^t + F(X_{r_2}^t - X_{r_3}^t) \quad (2)$$

Each candidate solution is evaluated based on an objective function that focuses on accuracy as the key performance indicator. The objective function used in hyperparameter optimization in each model has the following mathematical formula:

$$min\, f(\theta) = 1 - CVAccuracy(\theta) \quad (3)$$

where θ represents the combination of hyperparameters being tested and $CVAccuracy(\theta)$ is the average accuracy obtained through the 10-fold cross-validation process on the training data.

Table 3. Differential Evolution algorithm parameters

Parameter	Nilai	Keterangan
maxiter	15	Determine the maximum number of generations in the optimization process
popsize	20	Determine the number of individuals (candidate solutions) in each generation.
strategy	best/1	A mutation strategy that uses the best solution as a search basis.
seed	42	Used to ensure that experimental results can be reproduced.
polish	True	Perform refinement of the final solution using local optimization
Mutation Factor (*F*)	0.5 – 1 (default)	Controls the magnitude of changes in the mutation process, uses default values from SciPy.
Crossover Rate (*CR*)	0.7 (default)	Sets the probability of component exchange between solutions

The configuration of the Differential Evolution algorithm used can be seen in Table 3, which includes a maximum number of iterations (maxiter) of 15, a population size (popsize) of 20 individuals, a mutation strategy of best/1, and other parameters using the standard configuration from the SciPy library. The final result of the optimization process is the best combination of hyperparameters which is then used in the model training process. An optimization process is applied to various machine learning models to tune their respective hyperparameters. Table 4 summarizes the hyperparameters optimized using the DE algorithm.

Data Classification

The classification stage was carried out by applying nine machine learning models, namely Random Forest (RF), AdaBoost (AB), Gradient Boosting Machine (GBM), Support Vector Machine (SVM), Stochastic Gradient Desent (SGD), Logistic Regression (LR), Gaussian Naïve Bayes (GNB), Extra Trees Classifier (ETC), and Decision Tree (DT). Each model was trained using training data that had gone through preprocessing and hyperparameter optimization stages using Differential Evolution. The trained model is then evaluated using test data to measure its predictive ability for heart failure patient survival based on predetermined evaluation metrics.

Table 4. Hyperparameters optimized by DE for various models

Model	Hyperparameters Optimized by DE
RF	n_estimators, max_depth, min_samples_split, min_samples_leaf
AB	n_estimators, learning_rate
GBM	n_estimators, max_depth, min_samples_leaf
SVM	C, gamma, kernel
SGD	alpha, eta0, penalty
LR	C, penalty
GNB	var_smoothing
ETC	n_estimators, max_depth, min_samples_split, min_samples_leaf
DT	max_depth, min_samples_split, min_samples_leaf, criterion

Performance Evaluation

Model performance evaluation is conducted using test data that was not included in the training or optimization process. Several evaluation metrics were used to measure the quality of the model's predictions, namely accuracy, precision, recall, f1-score, and ROC (AUC).

Accuracy is an indicator metric used to measure the percentage of correct predictions against the entire test data [33]. In general, accuracy describes the proportion of correct predictions both True Positive (TP) and True Negative (TN), calculated against the total number of observations [34]. Precision is a metric used to assess a model's accuracy in predicting the positive class. High precision in heart disease prediction indicates more accurate positive predictions and fewer false positive errors [34]. Meanwhile, Recall is a metric that measures a model's ability to detect all actual positive cases. In the context of cardiovascular disease, this metric is important because of its ability to reflect the extent to which the model identifies positive cases [34]. The F1-Score is calculated as the harmonic mean of Recall and Precision, thereby yielding a more balanced evaluation. Furthermore, ROC-AUC indicates the model's effectiveness in distinguishing the difference between negative and positive data at various classification thresholds. Probabilistically, AUC indicates the probability of positive data scores being higher than negative data, where a high AUC value indicates a more efficient model, especially on imbalanced data [35], [36]. Mathematically, the performance evaluation metrics applied in this research can be formulated as follows.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

$$F1-Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

The model performance before and after hyperparameter optimization was then compared to analyze the effectiveness of the Differential Evolution algorithm in improving the survival prediction ability of heart failure patients.

RESULT AND DISCUSSION

In this research, the data preparation phase involved separating the target variable (y), namely DEATH_EVENT which represents the patient's survival status, from the clinical variables (X) present in the dataset. The next stage was data splitting, in which the dataset was divided into training and testing sets with proportions of 80% and 20%, respectively. The feature selection, handling of class imbalance, and data scaling processes were subsequently performed using the training data, while the testing data were kept independent for final model evaluation. The feature selection stage was performed using SelectKBest with the Chi-square statistical test to select nine relevant features from the training data. The nine features identified for reporting include anaemia, age, ejection fraction, creatinine phosphokinase, platelets, serum creatinine, high blood pressure, serum sodium, and time.

As seen in Figure 2, the dataset used in this research experienced an imbalanced data distribution, so the SMOTE method was applied to the training data to address this issue. Figure 3 shows the data visualization after handling the imbalanced data using SMOTE.

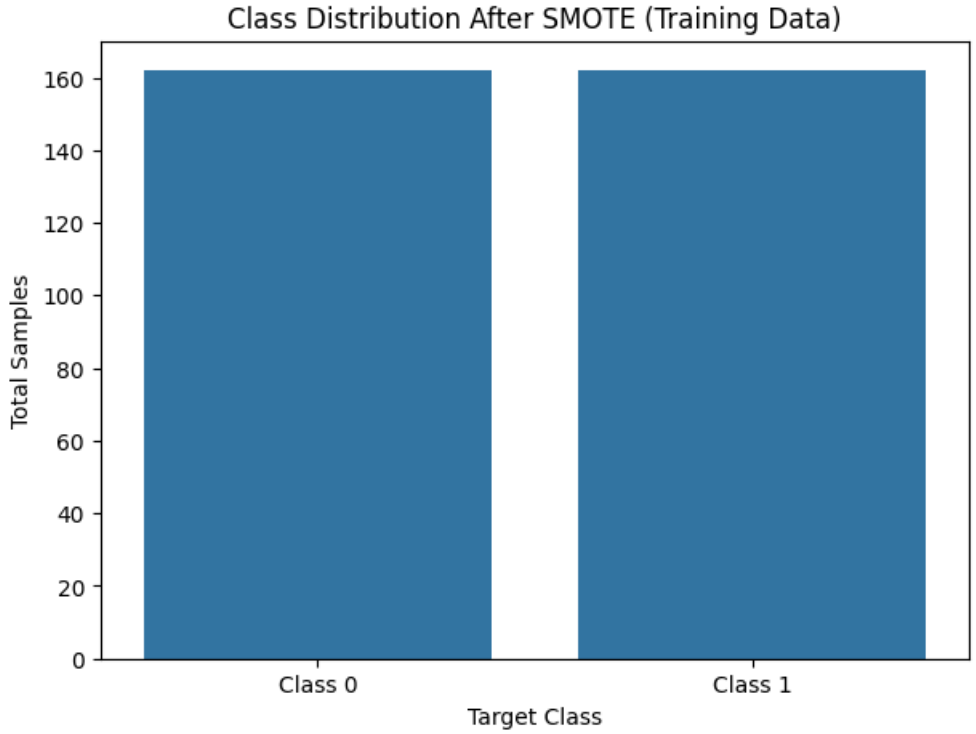


Figure 3. Data visualization after SMOTE

The visualization shows that the distribution of samples between classes in the training data is balanced after applying the SMOTE method. The subsequent step involves data scaling using the StandardScaler method to standardize the value of each feature to have a standard deviation of one and a mean of zero. The testing data were not subjected to SMOTE and remained in their original class distribution to preserve their independence for final model evaluation.

Table 5. Data distribution before and after data splitting and SMOTE

Dataset	Total	Survive	Die
Original Dataset	299	203	96
Training Set (80%)	239	162	77
Training Set after SMOTE	324	162	162
Testing Set (20%)	60	41	19

As seen in Table 5, the original dataset consists of 299 observations, with 203 surviving and 96 deceased patients. After data splitting, 239 observations were assigned to the training set and 60 observations to the testing set. The training set initially contained 162 surviving and 77 deceased patients and was subsequently balanced using SMOTE, resulting in 324 training samples with an equal distribution of 162 samples in each

class. In contrast, the testing set remained unchanged, consisting of 41 surviving and 19 deceased patients, and was kept independent from the preprocessing and model training processes.

After the data preprocessing process is complete, the next stage is hyperparameter optimization using the Differential Evolution (DE) algorithm. The optimization process is carried out to obtain the best hyperparameter combination for each machine learning model based on the objective function used, namely 1 − CV Accuracy. Each candidate solution on the training data is evaluated using 10-fold cross-validation, so that the obtained configuration is expected to optimally improve model performance.

Table 6. Optimal hyperparameter configuration

Model	Best Hyperparameter
RF	- n_estimators: 121 - max_depth: 15 - min_samples_split: 7 - min_samples_leaf: 20
AB	- n_estimators: 194 - learning_rate: 0.0226
GBM	- n_estimators: 382 - learning_rate: 0.0355 - max_depth: 13 - min_samples_leaf: 18
SVM	- C: 0.001407 - gamma: 0.000960 - kernel: linear
SGD	- alpha: 0,000003671 - eta0: 0.000104 - penalty: l1
LR	- C: 0.1125 - penalty: l1
GNB	- var_smoothing: 0.07694
ETC	- n_estimators: 220 - max_depth: 13 - min_samples_split: 11 - min_samples_leaf: 4
DT	- max_depth: 5 - min_samples_split: 18 - min_samples_leaf: 10 - criterion: gini

The best hyperparameter configuration resulting from the optimization process using the Differential Evolution algorithm for each model is shown in Table 6. The optimization results show that each algorithm has a different combination of hyperparameters, according to its characteristics and learning mechanisms. These differences in configuration indicate that hyperparameter combinations are not universal for all models. Next, the best hyperparameter configurations obtained from the optimization process are used in the final model training and classification process for heart failure mortality prediction. After optimization and classification, the next stage continues with prediction and evaluation, utilizing accuracy, precision, recall, F1-score, and ROC-AUC as performance evaluation metrics.

Table 7 and Table 8 show the performance evaluation metrics of nine machine learning models before and after hyperparameter optimization using the Differential Evolution algorithm. Overall, performance improved for most of the tested models following Differential Evolution, although this varied across models. This demonstrates that hyperparameter optimization can influence the model's ability to learn data patterns and affect the resulting mortality prediction performance for heart failure patients.

Table 7. Performance evaluation metric without differential evolution (baseline)

Model	Performance Evaluation Metric				
	Acc	Prec	Rec	F1-Score	ROC (AUC)
RF	0.8167	0.7500	0.6316	0.6857	0.8883
AB	0.8333	0.7647	0.6842	0.7222	0.8729
GBM	0.8167	0.7222	0.6842	0.7027	0.8511
SVM	0.8000	0.7059	0.6316	0.6667	0.8306
SGD	0.7833	0.7500	0.4737	0.5806	0.8716
LR	0.8167	0.7222	0.6842	0.7027	0.8652
GNB	0.7333	0.6000	0.4737	0.5294	0.8164
ETC	0.8333	0.8462	0.5789	0.6875	0.8787
DT	0.7833	0.6667	0.6316	0.6486	0.7426

Table 8. Performance evaluation metric with differential evolution (proposed)

Model	Performance Evaluation Metric				
	Acc	Prec	Rec	F1-Score	ROC (AUC)
RF	0.8500	0.7500	0.7895	0.7692	0.8922
AB	0.8167	0.7222	0.6842	0.7027	0.8691
GBM	0.8333	0.7647	0.6842	0.7222	0.8383
SVM	0.8500	0.9167	0.5789	0.7097	0.8909
SGD	0.8000	0.6842	0.6842	0.6842	0.8909
LR	0.8000	0.7059	0.6316	0.6667	0.8537
GNB	0.7333	0.6000	0.4737	0.5294	0.8203
ETC	0.8500	0.8571	0.6316	0.7273	0.8870
DT	0.8167	0.7500	0.6316	0.6857	0.8010

Referring to the evaluation conducted, Differential Evolution (DE) produced different effects across the nine machine learning models, indicating that the benefit of hyperparameter optimization depends on the learning characteristics of each model. Among the evaluated models, Random Forest (RF), Support Vector Machine (SVM), and Extra Trees Classifier (ETC) achieved the highest accuracy after optimization, each reaching 0.8500. RF also obtained the highest F1-score of 0.7692 and ROC-AUC of 0.8922 among the optimized models. Compared with the baseline configuration, RF improved its accuracy from 0.8167 to 0.8500, recall from 0.6316 to 0.7895, F1-score from 0.6857 to 0.7692, and ROC-AUC from 0.8883 to 0.8922. This improvement is consistent with the optimized RF configuration, which increased the number of estimators to 121 and set max_depth to 15, while introducing a larger min_samples_leaf value of 20. These changes indicate a configuration that provides stronger control over tree complexity and may contribute to better generalization on the test set.

The effect of DE was also observed in SVM and SGD, although the changes appeared in different evaluation metrics. SVM achieved an accuracy improvement from 0.8000 to 0.8500 and an increase in precision from 0.7059 to 0.9167. However, its recall decreased from 0.6316 to 0.5789, indicating that the optimized configuration increased the precision of positive predictions while reducing the proportion of positive cases correctly identified. The optimized SVM used a linear kernel with a relatively small C value of 0.0014 and gamma of approximately 0.0010, substantially different from the baseline RBF configuration. This change in kernel and regularization configuration provides a possible explanation for the shift in the precision recall trade off. For SGD, the improvement was more apparent in F1-score and ROC-AUC than in accuracy. Accuracy increased from 0.7833 to 0.8000, while F1-score increased from 0.5806 to 0.6842 and ROC-AUC increased from 0.8716 to 0.8909. The DE selected configuration used L1 regularization, an adaptive learning-rate strategy, a small alpha value of approximately 0,000003671, and eta0 of approximately 0.000104. These changes affect the strength of regularization and the learning-rate adaptation of SGD, which may explain why the optimization altered its classification behavior. Nevertheless, the improvement in SGD should not be interpreted as substantially greater than that of GBM based on accuracy alone, because both models improved by 0.0167 in accuracy. The larger increase in SGD F1-score indicates a different distribution of improvement across the evaluation metrics rather than a universally larger optimization effect.

The Decision Tree (DT) model also showed an improvement after optimization. Its accuracy increased from 0.7833 to 0.8167, F1-score from 0.6486 to 0.6857, and ROC-AUC from 0.7426 to 0.8010, while recall remained at 0.6316. The optimized configuration reduced max_depth to 5 and increased min_samples_split and min_samples_leaf to 18 and 10, respectively. These settings produce a more constrained tree than the

default configuration and may reduce unnecessary model complexity. The results therefore indicate that the optimized configuration was associated with improved accuracy and discrimination on the test set. Gaussian Naïve Bayes (GNB) showed a different pattern. Its accuracy remained at 0.7333, while precision, recall, and F1-score also remained unchanged at 0.6000, 0.4737, and 0.5294, respectively. Only a small change in ROC-AUC was observed, from 0.8164 to 0.8203. The DE process optimized only the var_smoothing parameter for GNB, unlike the tree-based and iterative models that expose several interacting hyperparameters. The limited change suggests that modifying var_smoothing within the investigated search space had little effect on the final classification decisions for this dataset. Therefore, the unchanged performance does not indicate that DE is ineffective in general, but rather that the available hyperparameter may have limited influence under the experimental conditions.

The comparison between GBM, RF, and ETC also provides an important observation. RF achieved an accuracy of 0.8500, F1-score of 0.7692, and ROC-AUC of 0.8922, whereas GBM achieved 0.8333, 0.7222, and 0.8383, respectively. GBM did, however, obtain slightly higher precision (0.7647) than RF (0.7500). ETC achieved the same accuracy as RF (0.8500), with an F1-score of 0.7273 and ROC-AUC of 0.8870. Thus, the similarity between ETC and GBM is primarily observed in their overall accuracy and discriminative performance, rather than in all evaluation metrics. The differences may be related to the distinct ensemble mechanisms and optimized parameter configurations used by the three models.

The results further indicate that no single hyperparameter can be identified as the most influential based solely on the final optimization results. DE simultaneously searches combinations of hyperparameters, and the observed performance reflects the interaction among these parameters rather than the isolated effect of one parameter. For example, RF was optimized using n_estimators, max_depth, min_samples_split, and min_samples_leaf, whereas GBM combined n_estimators, learning_rate, max_depth, and min_samples_leaf. Therefore, attributing a performance change to one parameter alone would require an additional sensitivity or ablation analysis. In this study, the hyperparameter results are consequently interpreted as model specific configurations rather than as evidence that one parameter universally dominates the optimization process.

Finally, the observed differences were evaluated using an exact McNemar test on predictions from the same untouched test set. The resulting p-values were 0.3750 for SVM, 0.6875 for RF, 0.7266 for DT, and 1.0000 for AdaBoost, SGD, Logistic Regression, GNB, GBM, and ETC. Thus, none of the baseline-versus-DE differences reached statistical significance at the 0.05 level. These results indicate that the observed performance changes should be interpreted as empirical differences within the current test set rather than as statistically significant improvements. The relatively small test set of 60 observations should also be considered when interpreting these findings.

The findings of this study provide additional evidence regarding the role of hyperparameter optimization in heart failure mortality classification. Ishaq et al [22] demonstrated the importance of feature selection and class imbalance handling in improving the predictive performance of machine learning models. The present study extends this line of research by incorporating Differential Evolution-based hyperparameter optimization after the data preparation stage. This finding is also related to Ahmed et al [23], who investigated hyperparameter optimization using Adaptive Inertia Weight Particle Swarm Optimization, and Ribino et al [24], who applied reinforcement learning for hyperparameter optimization in heart failure survival prediction. These studies indicate that hyperparameter configuration is an important consideration in the development of predictive models. In addition, Nicholas et al [26] demonstrated the application of Differential Evolution to binary heart disease classification. In the present study, DE produced observed accuracy improvements in six of the nine evaluated classifiers, while two models showed decreased accuracy and one model remained unchanged. However, the exact McNemar test showed that none of the baseline-versus-DE differences reached statistical significance at the 0.05 level. Therefore, the results indicate that DE can generate alternative hyperparameter configurations associated with different levels of predictive performance, but its effect is model specific and should not be interpreted as a statistically significant improvement across all classifiers.

CONCLUSION

This study proposes hyperparameter optimization of machine learning models using the Differential Evolution (DE) algorithm for heart failure mortality classification. The applied approach integrates feature

selection using SelectKBest, class imbalance handling using SMOTE, feature scaling, and DE-based hyperparameter optimization across nine machine learning models within a training only data preparation and cross-validation process to prevent information leakage. The results showed that DE produced model specific effects, with accuracy improvements observed in six of the nine evaluated models. Random Forest (RF), Support Vector Machine (SVM), and Extra Trees Classifier (ETC) achieved the highest accuracy of 0.8500, while RF obtained the highest F1-score and ROC-AUC of 0.7692 and 0.8922, respectively. Gaussian Naïve Bayes (GNB) showed unchanged accuracy, precision, recall, and F1-score, whereas AdaBoost (AB) and Logistic Regression (LR) experienced decreases in accuracy. The exact McNemar test indicated that none of the baseline-versus-DE differences reached statistical significance at the 0.05 level. Therefore, DE can generate alternative hyperparameter configurations with model specific effects, but the observed performance changes should not be interpreted as statistically significant improvements across the evaluated classifiers. Future research could compare Differential Evolution with other metaheuristic algorithms or apply it to more diverse datasets for a more comprehensive evaluation.

REFERENCES

[1] A. S. Osei-Nkwantabisa and R. Ntumy, "Classification and Prediction of Heart Diseases using Machine Learning Algorithms," pp. 1–10, 2024.

[2] A. Mahajan and B. Kaushik, "A Hybrid Feature Selection and Ensemble Stacked Learning Models on Multi-Variant CVD Datasets for Effective Classification," *IEEE Access*, vol. 12, no. June, pp. 87023–87038, 2024.

[3] World Health Organization, "Cardiovascular diseases (CVDs): Key Facts." [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[4] World Heart Federation, *WORLD HEART VISION 2030*.

[5] S. Rao *et al.*, "An Explainable Transformer-Based Deep Learning Model for the Prediction of Incident Heart Failure," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 7, pp. 3362–3372, 2022, doi: 10.1109/JBHI.2022.3148820.

[6] W. E. I. Zhou, Q. Chang, J. Chen, and X. Zhou, "Deep Ensemble Detection of Congestive Heart Failure Using Short-Term RR Intervals," *IEEE Access*, vol. 7, pp. 69559–69574, 2019, doi: 10.1109/ACCESS.2019.2912226.

[7] A. M. Qadri, A. Raza, K. Munir, and M. S. Almutairi, "Effective Feature Engineering Technique for Heart Disease Prediction With Machine Learning," *IEEE Access*, vol. 11, no. June, pp. 56214–56224, 2023, doi: 10.1109/ACCESS.2023.3281484.

[8] P. Ghosh, S. Azam, M. Jonkman, S. Shultana, and A. R. Beeravolu, "Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms With Relief and LASSO Feature Selection Techniques," *IEEE Access*, vol. 9, pp. 19304–19326, 2021, doi: 10.1109/ACCESS.2021.3053759.

[9] C. Chen, L. Li, M. Beetz, A. Banerjee, and R. Gupta, "Large Language Model-Informed ECG Dual Attention Network for Heart Failure Risk Prediction," *IEEE Transactions on Big Data*, vol. 11, no. 3, pp. 948–960, 2025, doi: 10.1109/TBDATA.2025.3536922.

[10] J. Huang, Y. Cai, X. Wu, X. Huang, J. Liu, and D. Hu, "Prediction of mortality events of patients with acute heart failure in intensive care unit based on deep neural network," *Computer Methods and Programs in Biomedicine*, vol. 256, no. May, p. 108403, 2024, doi: 10.1016/j.cmpb.2024.108403.

[11] K. Yongcharoenchaiyasit, S. Arwatchananukul, P. Temdee, and R. Prasad, "Gradient Boosting Based Model for Elderly Heart Failure, Aortic Stenosis, and Dementia Classification," *IEEE Access*, vol. 11, no. April, pp. 48677–48696, 2023, doi: 10.1109/ACCESS.2023.3276468.

[12] F. E. N. Miao, Y. Cai, Y. Zhang, X. Fan, and Y. E. Li, "Predictive Modeling of Hospital Mortality for Patients With Heart Failure by Using an Improved Random Survival Forest," *IEEE Access*, vol. 6, pp. 7244–7253, 2018, doi: 10.1109/ACCESS.2018.2789898.

[13] L. Ali *et al.*, "An Optimized Stacked Support Vector Machines Based Expert System for the Effective Prediction of Heart Failure," *IEEE Access*, vol. 7, pp. 54007–54014, 2019, doi: 10.1109/ACCESS.2019.2909969.

[14] S. Santhosh, K. Chadaga, R. Vijaya Arjunan, and S. D'Souza, "Cardiac Clarity: Harnessing Machine Learning for Accurate Heart-Disease Prediction," *IEEE Access*, vol. 13, pp. 97529–97544, 2025, doi: 10.1109/ACCESS.2025.3573760.
[15] S. Shafi, G. A. Ansari, and L. Alhazzaa, "Optimizing machine learning models for obesity risk prediction through hyperparameter tuning," *Systems and Soft Computing*, vol. 8, no. October 2025, p. 200472, 2026, doi: 10.1016/j.sasc.2026.200472.
[16] T. Fatima, K. Xia, W. Yang, Q. U. Ain, and P. L. Perera, "Diabetes Prediction Using ADASYN-Based Data Augmentation and CNN-BiGRU Deep Learning Model," *Computers, Materials & Continua*, vol. 84, no. 1, pp. 811–826, 2025, doi: 10.32604/cmc.2025.063686.
[17] S. F. Weng, J. Reps, J. Kai, J. M. Garibaldi, and N. Qureshi, "Can machine-learning improve cardiovascular risk prediction using routine clinical data ?," *PLOS ONE*, vol. 12, no. 4, pp. 1–14, 2017.
[18] S. I. Ansarullah and P. Kumar, "A Systematic Literature Review on Cardiovascular Disorder Identification using Knowledge Mining and Machine Learning Methods," *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, pp. 1009–1015, 2017.
[19] D. A. Anggoro, "Performance Comparison of Grid Search and Random Search Methods for Hyperparameter Tuning in Extreme Gradient Boosting Algorithm to Predict Chronic Kidney Failure," *International Journal of Intelligent Engineering & Systems*, vol. 14, no. 6, pp. 198–207, 2021, doi: 10.22266/ijies2021.1231.19.
[20] K. E. Hoque and H. Aljamaan, "Impact of Hyperparameter Tuning on Machine Learning Models in Stock Price Forecasting," *IEEE Access*, vol. 9, pp. 163815–163830, 2021, doi: 10.1109/ACCESS.2021.3134138.
[21] D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone," *BMC Medical Informatics and Decision Making*, vol. 20, p. 16, 2020.
[22] A. Ishaq *et al.*, "Improving the Prediction of Heart Failure Patients' Survival Using SMOTE and Effective Data Mining Techniques," *IEEE Access*, vol. 9, pp. 39707–39716, 2021, doi: 10.1109/ACCESS.2021.3064084.
[23] M. Ahmed, M. H. Sulaiman, M. M. Hassan, and T. Bhuiyan, "Predicting the Classification of Heart Failure Patients Using Optimized Machine Learning Algorithms," *IEEE Access*, vol. 13, no. January, pp. 30555–30569, 2025, doi: 10.1109/ACCESS.2025.3541069.
[24] P. Ribino, C. Di Napoli, G. Paragliola, and L. Serino, "Hyper-Parameter Optimization through Reinforcement Learning for Survival Prediction of Patients with Heart Failure," *Procedia Computer Science*, vol. 239, no. 2023, pp. 1754–1761, 2024, doi: 10.1016/j.procs.2024.06.354.
[25] Q. A. Hidayaturrohman, "A Comparative Analysis of Hyper-Parameter Optimization Methods for Predicting Heart Failure Outcomes," *Applied Sciences*, vol. 15, no. 6, p. 17, 2025.
[26] T. Nicholas, R. Egling, S. Mbuyu, and Z. Wang, "Binary Classification of Heart Disease Based on Differential Evolution-Optimised Machine Learning Approach," *Journal of Advances in Information Technology*, vol. 15, no. 4, pp. 467–479, 2024, doi: 10.12720/jait.15.4.467-479.
[27] J. Shang and G. Li, "A Hybrid Composite Differential Evolution and Multiobjective Particle Swarm Optimization Evolutionary Algorithm and Its Application," *IEEE Access*, vol. 12, no. June, pp. 74417–74431, 2024, doi: 10.1109/ACCESS.2024.3404407.
[28] I. Farda, A. Thammano, and J. Morris, "Differential Evolution with Self-adaptive Mutation and Population Improvement Strategy for Optimization Problems," *IEEE Access*, vol. 12, no. September, pp. 131809–131829, 2024, doi: 10.1109/ACCESS.2024.3460385.
[29] J. Leung, "Improving Educators' Search Engine Experience: A Quantitative Analysis of Search Terms," *IEEE Access*, vol. 12, no. April, pp. 69076–69086, 2024, doi: 10.1109/ACCESS.2024.3393423.
[30] K. Mohammad, M. Uddin, A. Al Mamun, A. Chakrabarti, and R. Mostafiz, "An ensemble machine learning-based approach to predict thyroid disease using hybrid feature selection," *Biomedical Analysis*, vol. 1, no. 3, pp. 229–239, 2024, doi: 10.1016/j.bioana.2024.08.001.
[31] N. Kedam, D. Kumar, V. Kumar, K. Mohamed, and M. Abdelaziz, "River stream flow prediction through advanced machine learning models for enhanced accuracy," *Results in Engineering*, vol. 22, no. May, p. 102215, 2024, doi: 10.1016/j.rineng.2024.102215.
[32] C. R. Rashmi and C. P. Shantala, "Evaluating Deep Learning with different feature scaling techniques for EEG-based Music Entrainment Brain Computer Interface," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 7, no. December 2023, p. 100448, 2024, doi: 10.1016/j.prime.2024.100448.

[33] G. Sunilkumar and P. Kumaresan, "Deep Learning and Transfer Learning in Cardiology: A Review of Cardiovascular Disease Prediction Models," *IEEE Access*, vol. 12, no. November, pp. 193365–193386, 2024, doi: 10.1109/ACCESS.2024.3514093.
[34] H. Khan, N. Javaid, T. Bashir, M. Akbar, N. Alrajeh, and S. Aslam, "Heart Disease Prediction Using Novel Ensemble and Blending Based Cardiovascular Disease Detection Networks: EnsCVDD-Net and BlCVDD-Net," *IEEE Access*, vol. 12, no. July, pp. 109230–109254, 2024, doi: 10.1109/ACCESS.2024.3421241.
[35] S. Mondal, R. Maity, and S. Member, "An Efficient Computational Risk Prediction Model of Heart Diseases Based on Dual-Stage Stacked Machine Learning Approaches," *IEEE Access*, vol. 12, no. December 2023, pp. 7255–7270, 2024.
[36] A. A. Asiri *et al.*, "Enhancing Brain Tumor Diagnosis : Transitioning From Convolutional Neural Network to Involutional Neural Network," *IEEE Access*, vol. 11, no. November, pp. 123080–123095, 2023, doi: 10.1109/ACCESS.2023.3326421.