I.J. Information Technology and Computer Science, **2026, 3, 109-129**
Published Online on June 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2026.03.08

Adaptive Swarm-Optimized Ensemble Learning for Generalizable Heart Disease Risk Prediction

Ankit Maithani
CSE Dept, School of Computing, DIT University, Dehradun, India
E-mail: ankit.maithani@dituniversity.edu.in
ORCID iD: https://orcid.org/0000-0003-2626-2490

Garima Verma*
School of Computing, DIT University, Dehradun, India
E-mail: garima.verma@dituniversity.edu.in
ORCID iD: https://orcid.org/0000-0002-6715-8596
*Corresponding author

Received: 08 January 2026; Revised: 27 March 2026; Accepted: 20 April 2026; Published: 08 June 2026

Abstract: Machine learning (ML) has made it much easier to find and estimate the risk of early-stage cardiovascular illnesses by making it possible to analyze massive, various clinical datasets quickly and easily. In these kinds of datasets, demographic information, lifestyle characteristics, medical history, and diagnostic measurements are all included. These are all things that may not be easy to see through standard clinical examination. This study examines heart disease prediction through a series of hybrid ML models that integrate neighborhood-based classifiers, swarm intelligence-driven optimization, and ensemble learning, motivated by existing obstacles. There are four hybrid models being proposed: MSMO-KE and MSMO-KM, which combine Modified Spider Monkey Optimization (MSMO) with K-Nearest Neighbour classifiers that use Euclidean and Minkowski distance measures, respectively. There are also two ensemble variants, MSMO-KECB and MSMO-KMCB, which add CatBoost as a final prediction layer. To make sure it is strong and can be used in other situations, the proposed framework is tested on three separate cardiovascular datasets using a cross-validation method. The experimental findings show that the performance is always better than the baseline and the best models that are already used. The MSMO-KMCB model performs the best overall out of all the approaches tested. It has a cross-validated accuracy of 98.2% on Dataset-3 while keeping a high sensitivity. The comparative research demonstrates that the proposed MSMO-based ensemble models surpass current methodologies in predictive accuracy and recall, underscoring their promise for dependable and efficient heart disease risk prediction in clinical decision-support systems.

Index Terms: Heart Disease, Modified Spider Monkey Optimization, KNN, Catboost, Cross-validation.

1. Introduction

Heart disease has emerged as a major public health concern and exerts immense economic and social impacts worldwide. It is now one of the leading causes of death in the world. This has led to more efforts in the management, prevention, and risk assessment of cardiovascular disease to emphasize the seriousness of this issue [1]. Cardiovascular disease (CVD) is the collective term for the wide group of diseases of the heart and blood vessels, primarily due to atherosclerosis. These are chronic conditions that develop over long periods without apparent symptoms, possibly appearing with different severities later in life [2]. It is also uncovered in research that 90% of all CVD cases are preventable, and through diagnostic tools like electrocardiograms or ECG, heart sound sensors, and hospital medical testing, early identification can be achieved. Among the risk factors named for heart disease are obesity, poor diet, lack of physical fitness and generally unhealthy lifestyle choices. Also, high blood pressure, smoking, uncontrolled diabetes, stress, family history of heart problems, hypertension, and high cholesterol add up to the risk of developing CVD [3-4]. According to the World Health Organization, more than 17 million people died from CVD in 2015, and about 25% of these had no warning symptoms before having a heart attack [4].

Angiography is the common diagnostic test for cardiovascular disease, but it is often time-consuming and costly, especially in resource-constrained settings. The rising mortality rates due to CVD have made early detection and timely intervention very important. From this angle, machine learning has become very important in medical diagnosis since it

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

assists in disease classification and fastens the diagnostic process for clinicians to save lives [5]. Data mining and ML are effective technologies for analyzing massive datasets in the healthcare business to find important information that can be used in many medical fields. ML algorithms analyze extensive medical datasets to uncover previously obscured patterns, hence enhancing the accuracy of disease diagnosis, detection, and prediction [6]. The techniques are basically important for enhancing healthcare outcomes and the quality of life, bringing down death rates, and improving the detection of diseases at an early stage [7].

Despite continuing research, the goal of consistent and reliable prediction in heart disease has not been met. However, the complexity of cardiovascular health, resulting from a diversity of contributing factors, requires the highest level of accuracy in predictive models if it is to provide an effective solution to this major world health problem [8-9].

This paper proposes four hybrid prediction models to improve heart disease risk assessment through the integration of neighborhood-based classification, swarm intelligence-driven optimization, and ensemble learning. The first two models, MSMO-KE and MSMO-KM, use kNN classifiers with MSMO and Euclidean and Minkowski distance methods, respectively. Two ensemble variations, MSMO-KECB and MSMO-KMCB, are created by adding CatBoost as a final prediction layer to these optimized classifiers. This helps capture higher-order feature interactions and makes the models more general. MSMO is based on the fission-fusion foraging behavior of spider monkeys. It is an adaptive optimization technique that strikes a good balance between exploration and exploitation, making it perfect for tuning hyper parameters in high-dimensional machine learning problems. A cross-validation methodology is used on three different cardiovascular datasets to test the proposed models. To deal with class distributions that are not balanced, we use imbalance-aware learning. Extensive experimental evaluations against baseline KNN classifiers and current state-of-the-art methodologies illustrate the enhanced predictive accuracy and resilience of the proposed MSMO-based ensemble framework. The key contributions by the authors in the work are as follows-

- This study introduces four novel hybrid models—MSMO-KE, MSMO-KM, MSMO-KECB, and MSMO-KMCB—that amalgamate Modified Spider Monkey Optimization (MSMO) with K-Nearest Neighbour classifiers, employing Euclidean and Minkowski distance metrics, and further enhance these optimized classifiers through CatBoost-based ensemble learning.
- The proposed MSMO framework uses dynamic leader guidance control and elite-preserving systems to optimize hyper parameters of the model in mixed search spaces which include both discrete and continuous values. This improves stability of convergence and predictive performance.
- The models are trained and tested on three separate datasets using imbalance-aware techniques and a cross-validation method. This makes them more robust and able to work with a wider range of data distributions.
- Comprehensive experimental assessments indicate that the proposed MSMO-based models routinely surpass baseline KNN classifiers and many existing state-of-the-art methodologies across multiple performance criteria, including accuracy and recall.

The remainder of the text is broken into the following sections: section 2 discusses several existing studies and their limitations. Section 3 discusses the description of the three datasets used for the study and the pre-processing steps applied to each. Section 4 addresses the proposed models along with experiment outcome analysis and compares with some previous studies. Finally, section 5 presents the paper's conclusion and future scope.

2. Related Works

Recent machine learning developments have given much help in predicting heart disease by fusing various techniques [11]. However, some limitations remain, such as the imbalance of datasets, suboptimal feature selection, and generalization issues.

In a study [12], Mohan et al. employed a machine learning-based predictive model by means of random forest (RF) in conjunction with a linear model to get an 88.7% accurate result in predicting heart diseases. Their method did not handle class imbalance and may give biased predictions. Besides, the size of the dataset was relatively small, which limits generalizability.

Pooja Rani et al. developed a cardiac disease prediction system using a hybrid feature selection algorithm with genetic algorithms and recursive feature elimination. The dataset was found to be imbalanced, so SMOTE data balancing was done to achieve a balanced dataset, and some other techniques were used to improve data quality. The study achieved good accuracy but was limited by the size of the dataset and genetic algorithm limitations, such as the tendency to converge to a suboptimal solution prematurely [13].

Alqahtani et al. proposed an ensemble-based ML and DL framework for predicting cardiovascular disease using six classification models trained on a publicly available dataset. Although the ensemble model showed an accuracy of 88.7%, the lack of data balancing techniques and limitations in the dataset size affected its performance [9].

As per the study [1], the authors have proposed a model called SOLSSA, which is a combination of improved Sparrow Search Algorithm, opposition-based learning combined with lateral mutation using the Salp Swarm Algorithm. The proposed model leads to superior values of F1 scores of 90 and 81.5 % on two CVD datasets. The robustness of this model still needs to be evaluated by comparing more diverse datasets with state-of-the-art models.

A dataset consisting of 2642 samples from 717 patients with 33 attributes was used by Zhang et al. to diagnose CHD using the CatBoost model. The data was collected through various surveys, lab reports, and demographics. The authors implemented various ML models such as logistic regression (LR), random forest (RF) and catboost. Its high performance was noted in the study by the CatBoost model. Although the study showed good improvement in accuracy, it will need to be validated in a broader variety of datasets to be more generalized [3]. The authors of study [14] proposed a catboost model with an improvisation done with SHAP (SHapley Additive exPlanations) explainability technique. An accuracy of 84.4% was achieved using the CatBoost classifier with SHAP. In addition to improving model interpretability, a limited dataset may raise generalization concerns.

To classify ECG signals based on features extracted from the time domain and amplitude domain, Dhananjay et al. applied Extremely Randomized Trees, CatBoosts, and Ridge Classifiers. Among these, CatBoost showed the highest performance in terms of sensitivity and precision. However, this study did not detail any pre-processing steps of ECG, which may affect the performance of models, especially noise reduction [15].

Another study [16], introduced an ML-based technique for the estimation of the risk for heart diseases, which was quite different in applying a mean-based splitting approach using the CART and derived an accuracy-based weighted ensemble with 93% and 91% accuracy for the Cleveland and Framingham datasets, respectively. While the results here were quite encouraging, they surely need much validation on various diversified datasets for the strengths of model checking. Shaikhan et al. implemented a stack ensemble learning approach for predicting heart disease risk by integrating a number of weak learners and using SMOTE to deal with class imbalance. Their model achieved an AUC of 0.922 in the ROC-AUC curve. However, it may not generalize well from the UCI repository dataset alone, as the dataset may not fully represent diversity in real clinical settings [17]. Majumder et al. proposed an ML-based predictive framework using Extra Trees Classifier (ETC) and a hybrid ensemble voting classification technique with an accuracy of 86.89% in demonstrating the effectiveness of combining multiple classifiers for early heart disease detection. The standardization and pre-processing of the datasets were analyzed to a lesser extent regarding potential effects on real-world applicability [18].

Table 1. Reviews of existing studies

Study	Technique / Model Used	Dataset Characteristics	Data Imbalance Handling	Key Limitations Identified	How the Proposed MSMO-Based Framework is Unique
[12]	Random Forest + Linear Model	Small dataset	Not addressed	Class imbalance ignored; limited generalizability	Proposed work explicitly addresses imbalance and validates performance on three heterogeneous datasets using optimized hybrid models
[13]	GA + RFE + ML classifiers	Small, imbalanced dataset	SMOTE	Premature convergence of GA; limited dataset	MSMO avoids premature convergence via adaptive leader control and elite preservation and demonstrates robustness across datasets
[9]	Ensemble ML & DL (6 models)	Single public dataset	Not addressed	No imbalance handling; dataset dependency	Proposed models integrate optimization + ensemble learning with imbalance-aware evaluation and cross-validation
[1]	Hybrid Sparrow–Salp Swarm optimization	Two CVD datasets	Not clearly discussed	Limited validation against diverse datasets	Proposed work uses three datasets, cross-validation, and ensemble learning to improve generalizability
[3]	CatBoost, LR, RF	2642 samples, 33 features	Not emphasized	Single dataset; limited generalization	Proposed work combines CatBoost with MSMO-optimized KNN, achieving higher robustness across datasets
[14]	CatBoost + SHAP	Small dataset	Not emphasized	Limited dataset size	Proposed framework focuses on predictive robustness and optimization, not just explainability, across multiple datasets
[15]	CatBoost, Extra Trees, Ridge	ECG signal features	Not discussed	Pre-processing not detailed	Proposed work includes systematic pre-processing and optimizati**on**, ensuring stable learning
[16]	CART-based weighted ensemble	Cleveland & Framingham	Not emphasized	Limited dataset diversity	Proposed framework extends evaluation to multiple heterogeneous datasets
[17]	Stacked ensemble + SMOTE	UCI dataset	SMOTE	Generalization concerns	Proposed work combines swarm optimization + ensemble learning with cross-validation for robustness
[18]	Extra Trees + voting ensemble	Single dataset	Not detailed	Limited pre-processing discussion	Proposed work explicitly integrates optimization, ensemble learning, and imbalance-aware validation
Proposed Work	**MSMO-KE, MSMO-KM, MSMO-KECB, MSMO-KMCB**	**Three heterogeneous cardiovascular datasets**	**Imbalance-aware learning + CV**	—	**Unified hybrid framework combining adaptive swarm optimization, neighbourhood-based learning, ensemble boosting, and multi-dataset cross-validated evaluation**

The existing studies point out the requirement for more adaptive hybrid models in addressing dataset imbalance, optimization of feature selection, and generalizability. There are still some more problems with the methods used. First, a lot of research uses only one dataset, which makes it hard to apply their models to other clinical groups. Second, not

much attention is paid to computational complexity and optimization efficiency, especially when swarm intelligence or ensemble learning methods are used. Metaheuristic algorithms sometimes necessitate a meticulous equilibrium between exploration and computing expense; nevertheless, several existing studies fail to examine convergence behavior or optimization overhead. Third, it's hard to compare studies since they employ various datasets, feature sets, and ways to evaluate them. Consequently, stated accuracy values cannot consistently be directly compared between investigations.

The proposed MSMO-based framework tries to solve these problems by (i) testing models on a variety of different cardiovascular datasets, (ii) using optimization methods that keep the search space small to keep the computations efficient, and (iii) using the same experimental protocol on all datasets to make it possible to compare models fairly. Table-1 shows the summarization of most of the studies discussed in this section to identify the gaps in the studies.

3. Methods and Materials

3.1. Dataset Description

This work exploits the use of three different datasets, a strategy designed to increase precision and widen the area of analysis. This study makes an attempt to capture the different insights and unique features of each by using integrated datasets.

A. Dataset 1

Dataset-1 is taken from an ongoing cardiovascular study on the residents of Framingham, Massachusetts, which is freely available. The dataset contains more than 4000 patient records with 15 attributes. Each attribute in the dataset represents a plausible risk factor, including demographic, behavioural, and clinical symptoms. Proper pre-processing has been applied to the data (outlier detection, missing value, etc.) to ensure accuracy and relevance for predictive modelling. The classification goal is to predict the 10-year risk of CHD for each patient within the following decade. The target variable, "TenYearCHD" is binary: "1" corresponds to a high risk for CHD in the next 10 years, while "0" corresponds to no significant risk. This dataset provides a firm ground for developing and testing machine learning models that can improve strategies in CHD prediction and prevention [19].

B. Dataset 2

The dataset-2 used in this work is a comprehensive set that includes four separate databases from Cleveland, Hungary, Switzerland, and Long Beach V. Each dataset is detailed, with 1025 entries and 14 attributes. This dataset is widely available on Kaggle and is very important to any researcher or practitioner in the field. It allows the study to be conducted across a broader range of settings, enabling comparison analyses across diverse geographical regions and increasing the results' robustness and generalizability. The "target" field in the second dataset denotes the presence of heart disease in patients; "0" means no disease, and "1" means a patient has the disease. This binary classification scheme helps to reduce the complexity of assessing the existence of heart disease for practical analysis and forecasting of patients' outcomes [6].

C. Dataset 3

The Stroke Prediction Dataset that was made public during the McKinsey & Company Healthcare Analytics Hackathon and is available on the Kaggle platform is the same as Dataset-3 utilized in this work. You may get to the dataset at: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset. There are 5,110 anonymised patient records in the collection, each with 12 clinical and demographic characteristics. These include age, gender, hypertension, heart disease, marital status, work type, home type, average glucose level, body mass index (BMI), and smoking status. The dataset was first generated to forecast when a stroke would happen based on health indicators of the patients. All records are completely anonymous and made available to the public for research purposes, which meets ethical criteria for secondary data analysis. Because the dataset does not have any personally identifiable information (PII), there was no need for any more institutional ethical approval. The dataset was initially created to predict strokes, but it also features a binary characteristic called "heart_disease" that tells you if the patient has been diagnosed with a heart issue (0 = no heart disease, 1 = heart illness). In this study, this feature serves as the target variable for the analysis of cardiovascular risk patterns. The stroke variable is kept as an explanatory factor since earlier medical research has shown strong links between stroke risk factors and heart disease. Before training the model, a number of pre-processing procedures were taken: deleting records that don't have BMI values, putting nominal characteristics (such as gender, occupational type, and smoking status) into categories• utilizing Min-Max scaling to normalize numerical features, dealing with class imbalance by randomly oversampling only the training set [19].

3.2. Data Normalization

Min-Max scaling, a feature scaling technique, transforms the features with numerical values in the dataset into a predefined range. The features are normalized using Min-Max scaling to have a value from 0 to 1 [20]. This step has to be performed to prevent features with large magnitudes from dominating the learning algorithm. In addition, this scaling ensures that the features are within a range that makes comparing and interpreting them easy. Eq-1 shows the normalization process:

$$X_{scaled} = \frac{(X - X_{minimum})}{(X_{maximum} - X_{minimum})} \quad (1)$$

Where the actual value of feature is represented by X, scaled value is X_{scaled} , $X_{minimum}$ is the least value, and $X_{maximum}$ is the maximum value of the feature.

3.3. Data Balancing using Oversampling

One of the widely used data balancing techniques is oversampling, used to handle imbalance in the categorical feature of the dataset. An imbalance arises when one class, referred to as a majority class, has substantially more samples than another class referred to as the minority class. In our first experiment, we apply an oversampling method using a non-heuristic algorithm namely, random oversampling. The main intention of using random oversampling is to handle class imbalance by adding randomness while duplicating instances from the minority class, so approximately a balanced distribution of classes is obtained in the dataset [21-23]. Eq-2 mathematically depicts the oversampling process.

$$N_{diff} = N_{maj} - N_{min} \quad (2)$$

Where:

N_{maj} - Amount of instances in the majority class, N_{min} - Number of instances in the minority class

From the minority class, choose N_{diff} instances at random. The chosen instances should be duplicated and added to the minority class.

Class imbalance during model training will be lessened because there will be an equal number of examples in the final dataset for both the majority and minority classes.

3.4. K-Nearest Neighbour (KNN)

KNN is an essential approach of ML which is used for solving the problem of regression as well as classification. KNN is a flexible method that may be useful to a varied range of tasks, such as anomaly detection, picture identification, and recommendation systems [17]. KNN is called a non-parametric algorithm, which is also called an instance-based algorithm. The KNN makes predictions as per the majority class or average of the data points considered as 'k' nearest in the feature space.

Similarity is calculated by KNN using items being compared. The majority votes cast by a KNN's immediate neighbours determine its classification. Normally, in the feature space, the data points which are of a similar type are nearer to each other. Each data point is labelled with the class that has the most neighbours. With more nearest neighbours, k may be better chosen and the accuracy may be improved [18], [24].

The two types of KNN considered in this study are KNN with Euclidean distance (KNN-E) and KNN with Minkowski distance (KNN-M), which is used as the measure of the similarity between data points. In a multi-dimensional feature space, Eq-3 is used to calculate the Euclidean distance between two data points, X and Y .

$$(X, Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \quad (3)$$

Where X_i & Y_i are the feature values of data points X and Y for the i^{th} dimension and n represents the total number of dimensions, called features.

KNN with Minkowski Distance

Integral to KNN, the choice of the distance metric defines how similarity between the data points is measured. Within this choice of distance measures is Minkowski Distance, a generalized metric with Euclidean and Manhattan distances as special cases. The Minkowski Distance between two points $P = (x_1, x_2, ..., x_n)$ and $Q = (y_1, y_2, ..., y_n)$ in an n-dimensional space is given by Eq-4.

$$D(P, Q) = \left(\sum_{i=1}^{n} | x_i - y_i |^p \right)^{1/p} \quad (4)$$

Where x_i and y_i are the coordinates of the two points in n-dimensional space and n is the number of features.

3.5. Traditional Spider Monkey Optimization (SMO)

Spider Monkey Optimization (SMO) is a swarm intelligence-based optimization algorithm that imitates the foraging behaviour of spider monkeys. These monkeys are considered social animals, and they search for food in groups. The algorithm imitates their decisions and communication methods to reach the best solution for any problem [10], [25]. SMO proves helpful in feature selection, parameter optimization, and improvement of machine learning models. The SMO algorithm runs in six significant steps:

Initialization Phase:

A population of spider monkeys (solutions) is randomly initialized within the predefined bounds. Each monkey represents a candidate solution for a problem. Eq-5 represents the position calculation of a spider monkey

$$X_i = LB + \text{rand}() \times (UB - LB) \tag{5}$$

Where X_i is the position of i^{th} spider monkey, rand() generates a random number between 0 and 1, and LB, UB are the lower and upper bounds, respectively.

Local Leader Phase:

Each monkey updates its position on the best solution, called the local leader () in its group. Updating is depicted in Eq-6.

$$X_i^{'} = X_i + r_1 \times (Loc_{Lead} - X_i) \tag{6}$$

Where Loc_{Lead} is a local leader and r_1 is the random number in the range $[-1,1]$.

Global leader phase:

Each monkey also updates its position based on the best solution among all groups depicted in Eq-7, called the global leader ().

$$X_i^{'} = X_i + r_2 \times (Glo_{Lead} - X_i) \tag{7}$$

Where and r_2 is the random number in the range $[-1,1]$.

Local leader learning phase:

If a local leader does not improve for a certain number of iterations, it is replaced with a new candidate solution depicted in Eq-8.

$$X_i^{'} = X_i + \alpha \times (R_{neighbor} - X_i) \tag{8}$$

Where α is a learning parameter.

Stage Global Leader Learning

If the global leader does not improve over a predefined number of iterations, then a fraction of the worst performing monkeys is reinitialized to explore new solutions.

Local Leader Decision Phase:

If a group does not improve, its members dissolve the group and switch to other groups to gain diversity.

3.6. CatBoost

CatBoost is an implementation of the gradient boosting method by researchers and engineers at Yandex. CatBoost is a machine learning algorithm designed to handle category information efficiently; it stands for "Category" and "Boosting," hence the name. Combining ordered boosting with random permutations, CatBoost offers a solid way to handle categorical features, even when many categories are involved. This flexible technique is helpful for a wide range of problems in the disciplines of regression and classification [26]. The basis predictors in CatBoost are binary decision trees. Since it only needs to change the learning rate along with the number of iterations, CatBoost is robust and generally requires minimal hyper-parameter tuning. This is because the method uses ordered boosting, a random permutation-driven approach. Furthermore, CatBoost applies, in particular, to datasets with many categorical features; it is very efficient in handling categorical data using vector representation [27-28]. Without special tuning of parameters, CatBoost shows better results than XGBoost. One of the most important advantages of CatBoost is flexibility, as it has CPU and GPU

implementations that are faster than other gradient-boosting algorithms [29]. In CatBoost, the Mean Squared Error is used in creating the objective function for regression tasks, as shown in Eq-9.

$$Lo\left(A_i - \bar{A}_i\right) = \frac{1}{2n}\sum_{i=1}^{n}\left(A_i - \bar{A}_i\right)^2 \tag{9}$$

Where, loss of the i^{th} observation is represented by $Lo\left(A_i - \bar{A}_i\right)$, N is the total number of data points, A_i represents true target value, and $\bar{A}_i$ predicted target value. CatBoost frequently uses the log loss (cross-entropy) as the goal function for classification tasks, which is shown in Eq-10.

$$Lo\left(A_i - \bar{A}_i\right) = -\frac{1}{N}\sum_{i=1}^{n}\left[A_i \log\left(\bar{A}_i\right) + \left(1 - A_i\right)\log\left(1 - \bar{A}\right)\right] \tag{10}$$

Where, A_i is the true class label (0 or 1), and $\bar{A}_i$ is expected likelihood of being in class 1.

4. Proposed Model

This section of the work talks about the overall framework that was made for applying machine learning and adaptive optimization techniques to forecast cardiac disease early and accurately. To comprehensively assess the contributions of optimization and ensemble learning, six models are constructed, executed, and evaluated. Model-1 utilizes the K-Nearest Neighbor algorithm with Euclidean distance (KNN-E), and Model-2 uses K-Nearest Neighbour with Minkowski distance (KNN-M). These two models are the baseline classifiers. Model-3 (MSMO-KE) and Model-4 (MSMO-KM) build on these baselines by combining Modified Spider Monkey Optimization (MSMO) with KNN-E and KNN-M, respectively. This makes it possible to optimize hyper parameters and increase classification performance. Lastly, to improve prediction and generalization even more, two ensemble models are suggested: Model-5 (MSMO-KECB), which combines MSMO-KE with CatBoost, and Model-6 (MSMO-KMCB), which combines MSMO-KM with CatBoost. This progressive modeling technique enables a thorough evaluation of the effects of optimization and ensemble learning on the prediction of cardiac disease.

4.1. Modified SMO

Spider Monkey Optimization (SMO) is a metaheuristic based on swarm intelligence that is based on the way spider monkeys hunt for food in groups. The program simulates collaborative decision-making via local leaders, global leaders, and adaptive group reconfiguration to achieve a balance between exploration and exploitation. Classical SMO has been shown to be effective for optimization problems, but it may not work as well when used on complex, nonlinear, and mixed discrete-continuous search spaces, like hyper parameter optimization in medical prediction models, because it can get stuck too soon, have a fixed leader influence, and lose elite solutions. To address these constraints, this paper presents a Modified Spider Monkey Optimization (MSMO) algorithm that incorporates two fundamental additions to the conventional SMO framework: (i) Adaptive Leader Influence Control, and (ii) Elite-Preserving Global Leader Reinforcement. These changes make the algorithm more stable, better at finding solutions, and more robust without making it more complicated. In the suggested MSMO-KE and MSMO-KM models, MSMO fine-tunes the distance and neighborhood size settings for KNN classifiers. For the ensemble models MSMO-KECB and MSMO-KMCB, MSMO also adjusts hyper parameters in CatBoost, including the learning rate. The suggested changes let MSMO work with mixed discrete and continuous characteristics, reduce overfitting in datasets that have too many samples, and have better predictive accuracy across many cardiovascular datasets.

Let the population consist of N spider monkeys, where the position of the i^{th} monkey at iteration t is represented as $X_i^{(t)} = [x_{i,1}^{(t)}, x_{i,2}^{(t)}, ..., x_{i,D}^{(t)}]$, Where with D denoting the number of decision variables. In this study, D=2, corresponding to (k,p) for KNN-based models or (k,η) for CatBoost-based models. Each dimension is bounded as $x_{i,j} \in [LB_j, UB_j]$. The initial population is generated uniformly within these bounds- as per Eq-11, where, $r_{i,j} \sim U(0,1)$. The population is divided into multiple subgroups, each assigned a local leader (LL), while the best solution across all groups is designated as the global leader (GL).

$$x_{i,j}^{(0)} = LB_j + r_{i,j}(UB_j - LB_j) \tag{11}$$

Adaptive Leader Influence Control (ALIC)- In conventional SMO, the effect of local and global leaders stays the same from one iteration to the next. This might cause too much exploration in later stages or convergence too soon in early iterations. To fix this, MSMO adds adaptive leader influence coefficients that change how exploration and exploitation happen. This adaptive mechanism makes it easy to go from global exploration to local exploitation, which

speeds up convergence and improves solution refinement. Eq-12 shows how the local leader update has changed.

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} + \alpha_t(LL_{g,j}^{(t)} - x_{i,j}^{(t)}) + \beta_t(x_{r,j}^{(t)} - x_{i,j}^{(t)}) \tag{12}$$

Where, LL_g denotes the local leader of group g, r is a randomly selected monkey from the same group, and the adaptive coefficients are defined as Eq-13.

$$\alpha_t = \alpha_{max}\left(1 - \frac{t}{T}\right),\ \beta_t = \beta_{min} + (\beta_{max} - \beta_{min})\frac{t}{T} \tag{13}$$

Here, t is the current iteration and T is the maximum number of iterations. The decreasing α_t promotes broad exploration in early stages, while the increasing β_t strengthens exploitation as the algorithm approaches convergence. Now Eq-14 defines the modified global leader update of the proposed algorithm.

$$x_{i,j}^{(t+1)} = x_{i,j}^{(t)} + \gamma_t(GL_j^{(t)} - x_{i,j}^{(t)}) + \delta(x_{r,j}^{(t)} - x_{i,j}^{(t)}) \tag{14}$$

where γ_t gradually reduces the exploratory pull of the global leader over time.

Elite-Preserving Global Leader Reinforcement (EP-GLR)- In conventional SMO, if the global leader stops moving forward, it could lead to population reorganization that throws away good solutions. MSMO has an elite preservation strategy to stop this from happening. At each iteration, the best answers are kept in an elite subset. Elite preservation protects high-quality solutions, makes convergence more stable, and stops promising areas in the search space from being disturbed for no reason. Only the top people in this group, as described by Eq-15, can be chosen as the global leader.

$$GL^{(t+1)} = \underset{X_i \in \varepsilon}{\arg\max}\, f(X_i) \tag{15}$$

During global leader learning or fission–fusion phases, only non-elite solutions are subject to re-initialization as per Eq-16.

$$x_{i,j}^{new} = LB_j + r_{i,j}(UB_j - LB_j)\ ,\ \ X_i \notin \varepsilon \tag{16}$$

The optimization process continues until the maximum number of iterations is reached or convergence criteria are satisfied. The final global leader represents the optimal parameter configuration. After each update, feasibility is enforced as per Eq-17

$$x_{i,j}^{(t+1)} = \min(\max(x_{i,j}^{(t+1)}, LB_j), UB_j) \tag{17}$$

Fitness Function Definition - In the proposed MSMO-based optimization framework, the fitness function is designed to guide the search toward parameter configurations that maximize predictive performance while remaining robust to class imbalance. Since MSMO is formulated as a minimization problem, performance metrics are transformed into their negative counterparts. The fitness function (eq-18) is defined using the cross-validated Area Under the ROC Curve (AUC) to ensure stability under oversampled datasets.

$$\text{Fitness} = -AUC_{CV} \tag{18}$$

The AUC was chosen as the optimization goal since it tests how well the model can tell the difference between classes without using a predefined classification threshold. This feature is very significant for medical datasets because class imbalance is widespread and accuracy alone can give false results. The algorithm optimizes AUC during the MSMO search phase by trying to make the positive and negative classes as diverse as possible at different threshold settings. This makes sure that the optimization step puts more weight on strong ranking performance than on a certain decision boundary.

4.2. Proposed Model-1 (MSMO-KE) and Model-2 (MSMO-KM)

These two new models, MSMO-KE and MSMO-KM, combine the optimization capability of MSMO with ML classifiers, KNN-E and KNN-M, respectively. Aiming to use KNN's local search capabilities by exploiting the information-sharing mechanism of spider monkeys, they thereby combine the best of MSMO with KNN-E and KNN-M, respectively. The KNN component introduces similarity-based solution proximity into the exploitation of the monkeys,

to efficiently explore and exploit neighboring solutions. The whole process happens in two algorithms: Algorithm-1 , and 2. Algorithm-1 pinpoints the crucial steps of the model. The work starts with the ingestion of a dataset followed by partitioning (80:20 train-test ratio). Further, it includes a sequence of algorithm executions, starting from KNN to the proposed MSMO-KE and MSMO-KM models and, finally, evaluation of their respective performance metrics. The flow diagram of the model is shown in Fig-1.

Algorithm-1 Main Model
Input: heartdata-1 , 2 and 3, population=10, max-iter=10, n_dim=2, lb=[1,1], ub=[20,10]
Output: accuracy_MSMO-KE, accuracy-MSMO-KM, accuracy_MSMO-KECB, accuracy_MSMO-KMCB
Begin

1. Read heartdata-1, 2 and 3.
2. The dataset is divided into train and test sets: X_tr, y_tr, X_ts, y_ts using a 80% training and 20% test ratio.
3. Perform data normalization on X_tr, X_ts, y_tr, y_ts.
4. bestk=Call SMO(fitness_fun, n_dim, population, max_iter,lb,ub)
5. Extract best k and p values from global_best_position
6. accuracy_MSMO-KE =Call KNN-E(X_tr, y_tr, X_ts,bestk)
7. bestk, bestp=Call MSMO(fitness-function, n_dim, population, max_iter, lb, ub)
8. accuracy-MSMO-KM =call KNN-M(X_tr, y_tr, X_ts, bestk, bestp)
9. bestk, best_learning_rate=Call MSMO(fitness_fun, n_dim, population, max_iter,lb,ub)
10. accuracy_MSMO-KECB, accuracy_MSMO-KMCB =CALL train_catboost(X_tr, y_tr, X_ts, bestk, best_learning_rate)
11. print accuracy_MSMO-KE, accuracy-MSMO-KM, accuracy_MSMO-KECB, accuracy_MSMO-KMCB.

End

Algorithm-2 shows the steps that need to be taken for the proposed MSMO and how to evaluate fitness. The first stage is to create a population of spider monkeys in the predefined search area. Each possible solution contains the model parameters. The parameter k indicates the count of nearest neighbors in the KNN classifier, whereas the parameter p signifies the distance metric—Euclidean distance for KNN-E and Minkowski distance for KNN-M. The system then checks how good each solution is and finds the local leaders in each subgroup and the global leader for the whole group. MSMO then changes the placements of the solutions over and over again using learning methods guided by local and global leaders. This lets the search space be explored and used effectively. Adaptive control strategies are used to improve candidate solutions, and elite solutions are kept to make sure that convergence stays stable. When better fitness values are found, the local and global leader positions are changed. This process of optimizing over and over again goes on until the maximum number of iterations is achieved or the convergence requirements are met.

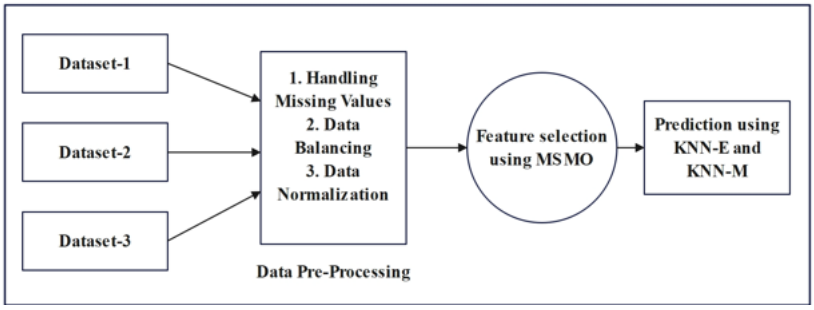


Fig. 1. Flow diagram of proposed model-1 and 2

Algorithm-2: Modified Spider Monkey Optimization (MSMO)
Input: Fitness function f(·), population size N ; maximum iterations T , bounds $[LB, UB]$, number of groups G ; local and global stagnation limits; elite rate E
Output: Optimal parameter vector GL
Begin

1. Initialize the population of N spider monkeys using Eq-(11).
2. Divide the population into G subgroups; assign a local leader (LL) to each group and identify the global leader (GL) from the elite set.
3. Evaluate fitness of all monkeys using the fitness function (Eq-18).
4. For iteration $t=1 \text{to} T$:

- Compute adaptive coefficients α_t and β_t using Eq-(13).
- Local Leader Phase:
  - Update monkey positions using the modified local leader rule in Eq-(12).

- Enforce boundary constraints using Eq-(17).
- Update local leaders if improved; otherwise update stagnation counters.

- Global Leader Phase:
  - Update positions using the modified global leader rule in Eq-(14).
  - Enforce feasibility using Eq-(17).
- Elite-Preserving Global Leader Reinforcement:
  - Select the global leader exclusively from the elite subset using Eq-(15).
- Leader Learning and Decision:
  - If local or global stagnation limits are exceeded, reinitialize non-elite solutions using Eq-(16) and reorganize groups.

End For

5. Return the final global leader GL as the optimal solution.

End

4.3. Proposed Model-3 MSMO-KECB and Model-4 MSMO-KMCB

A stacked ensemble architecture is presented to improve predictive accuracy and capture higher-order correlations among features by combining optimized KNN models with the CatBoost algorithm. MSMO-KECB and MSMO-KMCB are two new ensemble models that build on the MSMO-KE and MSMO-KM models that were formerly described. They include a second learning step based on gradient boosting. The suggested ensemble uses a two-stage stacking method. The original feature space is used to train the optimized base classifiers MSMO-KE and MSMO-KM in the first stage. These models make predictions based on neighborhood-based similarity measures that have been improved using the MSMO algorithm. To stop prediction leakage during training, out-of-fold predictions are made by utilizing the cross-validation method on the training data. During each fold of cross-validation, the optimized KNN model is trained on the training part of the fold and then used to make predictions for the validation part. Then, these predictions are combined across all folds to generate a full collection of out-of-fold prediction features for the whole training dataset. These prediction characteristics are the results of the basic learners and are used as input for the second learning stage.

In the second stage, the CatBoost classifier is used as a meta-learner to piece together the results of the KNN models that have been improved. CatBoost was chosen because it does a great job of classifying tabular data, it can find complex nonlinear correlations, and it has built-in ways to deal with categorical characteristics and cut down on overfitting through ordered boosting and regularization techniques. The base classifiers make predictions that the CatBoost model uses to train itself. This lets it learn higher-level decision boundaries that make its predictions more accurate overall.

In the last step of making predictions, the optimized KNN model is trained again on the whole training dataset and then used to make predictions on the test data that hasn't been seen yet. After that, the trained CatBoost meta-learner uses these outputs to make the final ensemble predictions. This process makes sure that the meta-learner only learns from predictions made from training data, which stops predictions from leaking from the training phase to the testing phase.

Algorithm-3 shows the step-by-step training process for the CatBoost-based stacking model. Figure 2 shows how the ensemble models MSMO-KECB and MSMO-KMCB function together. The main experimental pipeline in Algorithm-1 controls the whole training process by running the MSMO optimization, KNN-based base learners, and the CatBoost stacking stage in that order to make the final predictions.

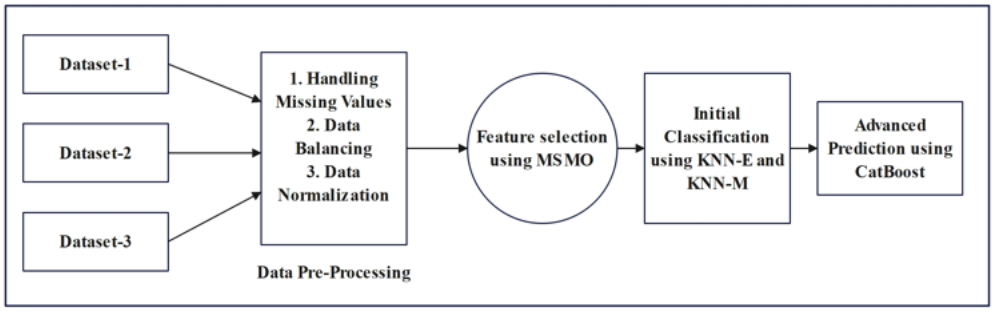


Fig. 2. Flow diagram of proposed model-3 and 4

Algorithm-3 Train_CatBoost

Input: *X_tr, y_tr, X_ts, k, learning_rate, iteration=100, depth=6*

Output*:* final_predictions

Begin

1. Perform K-fold cross-validation on training data
2. For each fold:
   Train optimized KNN model on training portion
   Generate predictions for validation portion
3. Collect out-of-fold predictions for all training samples

4. Train CatBoost classifier using out-of-fold predictions as input features
5. Train final KNN model on full training data
6. Generate predictions for test data using trained KNN model
7. Apply trained CatBoost model on these predictions
8. Produce final ensemble predictions

End

5. Performance Evaluation and Discussion

5.1. Experiment Setup

The suggested framework was executed in Python 3.x utilizing the PyCharm integrated programming environment on a Windows 10 operating system. Three diverse cardiovascular datasets were utilized to assess the robustness and generalizability of the proposed models across various clinical data distributions [30-31]. To make sure the evaluation process was strict and that no information got out, each dataset was first split into two parts: 80% for training and 20% for testing. During the creation of the model, the testing subset was never observed, and it was only utilized to evaluate the model's ultimate performance. All preprocessing and model optimization steps were done just on the training data. In the initial analysis, the class distribution showed that there was a clear class imbalance in all of the datasets. To fix this, random oversampling was only done on the training subset to make the minority and majority classes equal. This method makes sure that the testing data stays the same and keeps the evaluation process honest. Using Min-Max normalization, we scaled the features so that the numbers were in the range [0,1]. The normalization parameters were calculated solely from the training data and then utilized on the test data to prevent data leaking.

A 5-fold cross-validation process was used on the training subset to help improve the model and figure out how stable its performance was. The training data were separated into five parts, with four parts utilized for model training and one part used for validation, in that order. We used the MSMO technique to find the best hyper-parameter settings by combining this cross-validation approach with it. Once the best parameters were found, the final model was retrained using the whole training set and tested on the independent test set.

The KNN technique with two distance metrics, Euclidean distance (KNN-E) and Minkowski distance (KNN-M) is used, to build baseline classifiers. These baseline models served as benchmarks for future comparisons. Two new hybrid optimization-based models, MSMO-KE and MSMO-KM, were created by combining the Modified Spider Monkey Optimization algorithm with KNN-E and KNN-M, respectively. MSMO was utilized to find the best values for important model parameters, especially the neighborhood size and distance metric parameters. This made the classification work better. The MSMO-optimized KNN models were better at making predictions, but they couldn't capture complicated feature interactions. To fix this problem, two ensemble models—MSMO-KECB and MSMO-KMCB—were developed by adding CatBoost as a final prediction layer. We chose CatBoost because it can handle categorical features well, ordered boosting helps prevent overfitting, and it can find correlations between higher-order features.

The suggested framework was used to analyze the three datasets one after the other, and common classification measures including accuracy, precision, recall, and F1-score were used to measure how well the model worked. For classification-based evaluation metrics the predicted probabilities were converted into binary class labels using a decision threshold of 0.5. This threshold represents the standard default in binary classification and allows consistent comparison across baseline and optimized models [32]. The ROC–AUC analysis, however, evaluates model performance across all possible thresholds, which is why AUC was chosen as the objective function during the optimization stage. For the KNN-based models, the optimization concentrated on two principal parameters that directly affect the similarity calculation and the classifier's decision boundary: the neighborhood size k and the Minkowski distance parameter p. The neighborhood size k was explored within the range [1, 20], while the parameter p was searched within [1, 10], allowing the model to adaptively select distance metrics spanning from Manhattan distance (p=1) to Euclidean distance (p=2) and other generalized forms. The KNN classifiers used uniform distance weighting, where each neighbor contributes equally to the class decision. In cases of equal class votes, the class associated with the minimum average distance among neighboring instances was selected to resolve ties and ensure deterministic predictions. The MSMO algorithm was used to identify optimal parameter configurations. The MSMO framework was set up with a maximum of 10 iterations and a group of 10 spider monkeys. This setup might look modest compared to big metaheuristic optimization jobs, but it is right for this problem because the parameter search space is low-dimensional. In low-dimensional optimization contexts, reduced population sizes frequently attain swift convergence while preserving computational efficiency. Initial pilot tests utilized bigger search budgets, with population sizes varying from 10 to 30 and iteration counts from 10 to 50. These tests yielded minimal enhancements in predictive performance while significantly escalating computing expenses. So, the chosen MSMO architecture strikes a good mix between optimization efficacy and computing economy, letting the algorithm converge reliably without putting too much strain on the computer.

The CatBoost classifier was used as the last layer for the ensemble models' predictions. MSMO was used to find the best learning rate in the range of [0.01, 0.3], because it has a big effect on how quickly gradient boosting models converge. We kept certain other settings the same as they were in prior CatBoost-based research and preliminary experiments. For example, we set the tree depth to 6 and the number of boosting iterations to 100.

All datasets used in this study are publicly available research datasets, ensuring reproducibility and enabling

independent validation of the proposed framework. All three datasets utilized in this study exemplify distinct clinical environments and cardiovascular outcomes. Dataset-1 and Dataset-2 are mostly about diagnosing coronary heart disease. Dataset-3, on the other hand, comes from a stroke prediction dataset that includes cardiovascular risk indicators and a heart_disease attribute. Stroke and coronary heart disease are clinically separate illnesses; yet, they share numerous underlying risk factors associated with vascular pathology and metabolic dysfunction. By testing the proposed models on these different datasets, we can see how well they work with different types of cardiovascular risk patterns. However, the results from each dataset are examined separately, and the study does not assert clinical equivalency between stroke and coronary heart disease prediction tasks.

5.2. Evaluation of Models in Accordance with Various Measures for Dataset-1

The cross-validation method is used to test all of the models on Dataset-1, and the mean and standard deviation of the performance measures are shown in Table-2. This evaluation technique was developed to ensure robustness and to offset the impacts of sample bias introduced by class imbalance. First, the KNN-E and KNN-M classifiers were tested using cross-validated performance measures, with the neighborhood size chosen from the predetermined search range. Table 1 shows that KNN-E and KNN-M had average accuracies of about 0.80 and 0.81, respectively, with not much variation between folds. This set a solid baseline for comparison. We then tested the hybrid models MSMO-KE and MSMO-KM by using Modified Spider Monkey Optimization to adaptively tune the KNN parameters within the cross-validation framework. The MSMO-based optimization always made the predictive performance better than that of the KNN models that were used as a baseline. In particular, MSMO-KE and MSMO-KM had higher mean accuracies of about 0.83 and 0.85, respectively, as shown in Table 1. This shows that MSMO is good at finding strong parameter configurations across folds. The ensemble models MSMO-KECB and MSMO-KMCB, which use CatBoost as a final prediction layer, showed even better performance. These models had the best cross-validated accuracies on Dataset-1. MSMO-KECB had a mean accuracy of around 0.90, and MSMO-KMCB had a mean accuracy of about 0.92. Along with accuracy, recall (sensitivity) levels stayed high across all folds. MSMO-KMCB had an average recall of about 0.92, which shows that it was very good at correctly recognizing positive cardiac instances. Figure 3 shows a side-by-side view of the cross-validated performance metrics for all models on Dataset-1. Figure 4(a–f) shows the ROC–AUC curves that go with them. The ensemble MSMO-based models show better ROC characteristics and less variance, which further proves that they generalize better than both baseline and non-ensemble hybrid models.

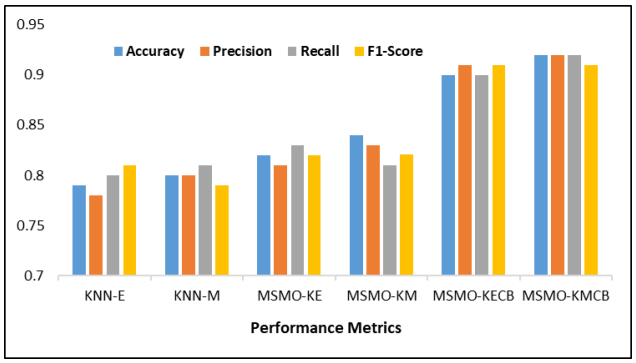
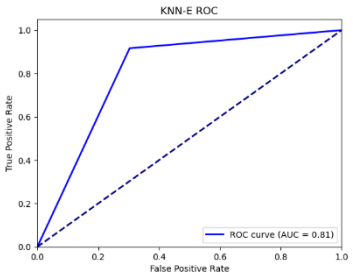
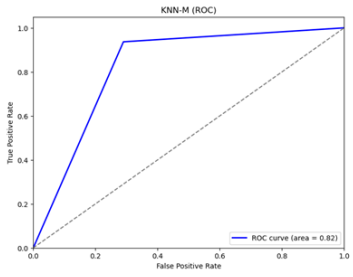


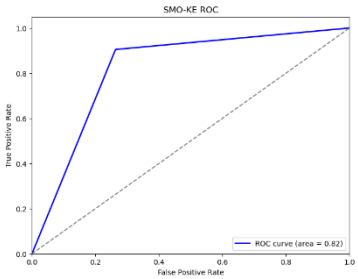
Fig. 3. Performance evaluation graph for dataset 1



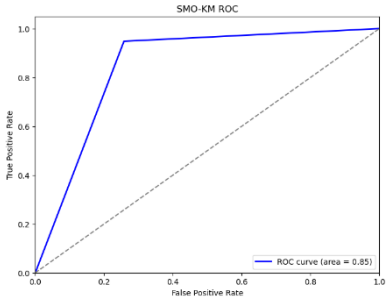
(a) AUC ROC curve KNN-E



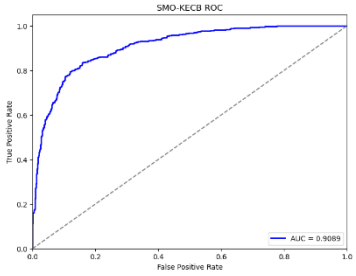
(b) AUC ROC curve KNN-M



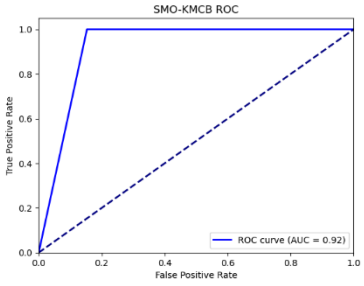
(c) AUC ROC curve MSMO-KE



(d) AUC ROC curve MSMO-KM



(e) AUC ROC curve MSMO-KECB



(f) AUC ROC curve MSMO-KMCB

Fig. 4. (a to f) AUC ROC curve of all the models using Dataset-1

5.3. Evaluation of Models in Accordance with Various Measures for Dataset-2

A cross-validation strategy is used to test all of the models on Dataset-2, and Table-2 shows the mean performance metrics for each model. Dataset-2 is more diverse than Dataset-1, which is shown by a widespread drop in baseline performance across models.

The KNN-E and KNN-M classifiers have average accuracies of about 0.77 and 0.79, respectively, over all validation folds. These results serve as a benchmark for evaluating the efficacy of MSMO-based optimization in the context of more complex data attributes. The hybrid models MSMO-KE and MSMO-KM were later tested by adding Modified Spider Monkey Optimization for adaptive parameter adjustment to the cross-validation framework. MSMO-KE showed just a small improvement over the baseline KNN-E, but MSMO-KM showed a big improvement, with a mean accuracy of about 0.88. This shows that MSMO-based optimization works best when combined with the Minkowski distance formulation on Dataset-2. The ensemble models MSMO-KECB and MSMO-KMCB, which use CatBoost as a final prediction layer, made even more progress. MSMO-KMCB had the best cross-validated accuracy on Dataset-2 of all the models tested, with an average value of about 0.89. Along with accuracy, recall (sensitivity) stayed high across all folds, with MSMO-KMCB showing a mean recall of about 0.92. This shows that it is reliable for finding positive cardiovascular cases. Figure 5 shows a side-by-side view of the cross-validated performance metrics for Dataset-2, and Figure 6(a–f) shows the ROC–AUC curves for the same data. The ROC characteristics further validate the enhanced discriminative ability and superior generalization performance of the ensemble MSMO-based models compared to both baseline and non-ensemble hybrid methodologies.

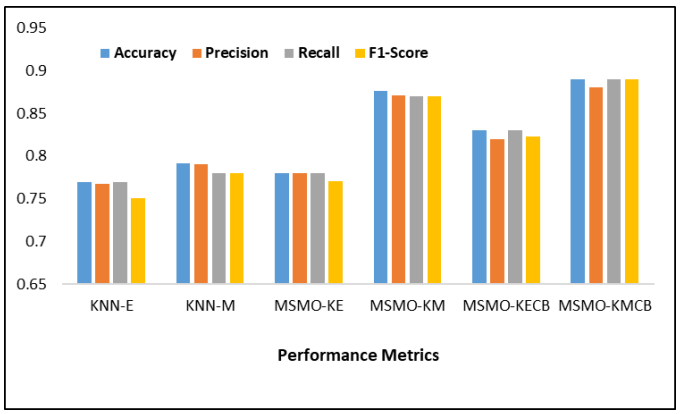
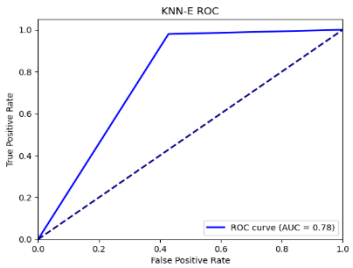
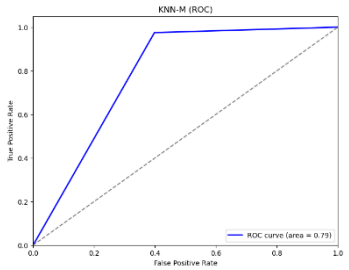


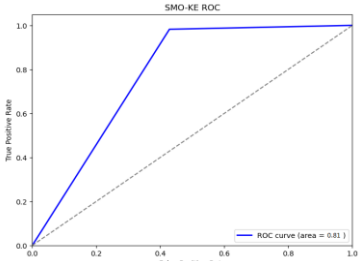
Fig. 5. Performance evaluation graph for dataset 2



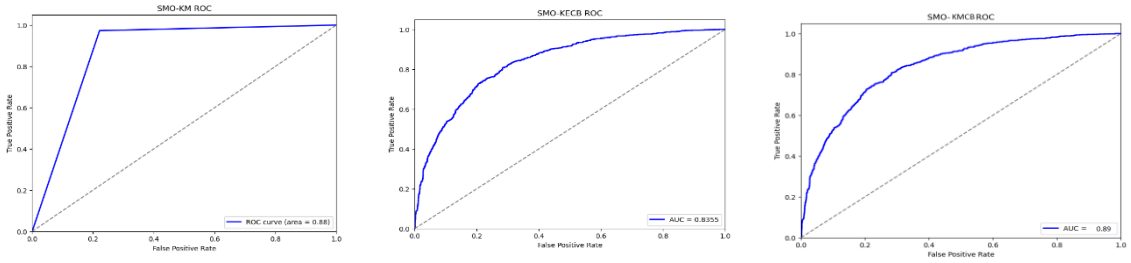
(a) AUC ROC curve KNN-E



(b) AUC ROC curve KNN-M



(c) AUC ROC curve SMO-KE



(d) AUC ROC curve MSMO-KM (e) AUC ROC curve MSMO-KECB (f) AUC ROC curve MSMO-KMCB

Fig. 6. (a to f) AUC ROC curve of all the models using dataset-2

5.4. Evaluation of Models in Accordance with Various Measures for Dataset-3

The identical experimental approach from Section 5.1 was used to test Dataset-3. At first, each dataset was split into two parts: a training set (80%) and a testing set (20%). The model was never trained on the testing set. We only used 5-fold cross-validation on the training data for all model optimization techniques. To fix the class imbalance without adding bias, random oversampling was only done in the training part of each cross-validation fold. The validation fold was not changed. The performance of the final model shown in Table-2 is based on the held-out test set. Cross-validation was performed to help find the best parameters and measure how stable the performance is.

Dataset-3 has more stable feature patterns and stronger links between predictors and the target variable than Dataset-1 and Dataset-2. This feature makes both the baseline and optimized models better at making predictions. The KNN-E and KNN-M baseline classifiers have mean accuracies of about 0.922 and 0.926, respectively. These results show that the dataset has a rather robust separability structure that works well with neighborhood-based learning methods.

The hybrid models MSMO-KE and MSMO-KM used the Modified Spider Monkey Optimization method to adaptively tune the KNN parameters, which made the predictions even better. The MSMO optimization technique worked well to find strong parameter settings and improve classification performance. The improved models had mean accuracies of about 0.963 and 0.975, respectively. The ensemble models MSMO-KECB and MSMO-KMCB, which include CatBoost as a final prediction layer, did the best overall on Dataset-3. MSMO-KMCB had the highest accuracy of 0.983 ± 0.008 of all the models tested, which shows that it is very good at making predictions. The model not only had great accuracy, but it also regularly had high recall values, which showed that it could reliably find positive cardiovascular cases. The relatively high performance seen on Dataset-3 can be explained by the fact that it includes multiple clinically important risk factors, such as age, hypertension, glucose level, and BMI. These traits are well-known signs of heart disease and give machine learning models significant signals that help them tell the difference between different types of data. Also, the low standard deviation across folds shows that the model is stable and consistent, not unstable and overfitting.

Figure-7 shows a side-by-side view of the performance metrics for all models on Dataset-3, and Figure 8(a–f) shows the ROC–AUC curves that go with them. The ensemble MSMO-based models exhibit superior ROC characteristics and diminished variance relative to baseline classifiers, validating their enhanced discriminative capability and superior generalization performance on this dataset.

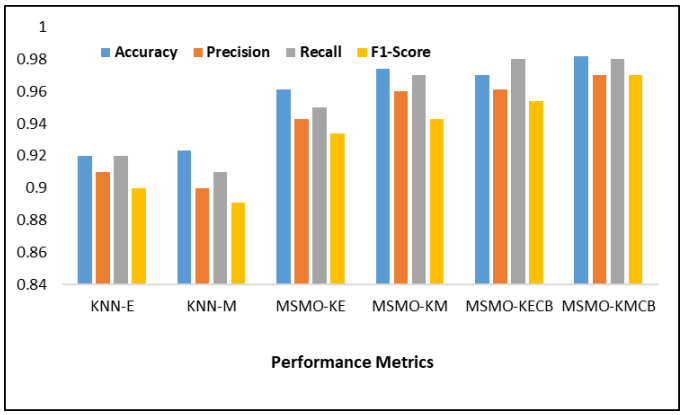
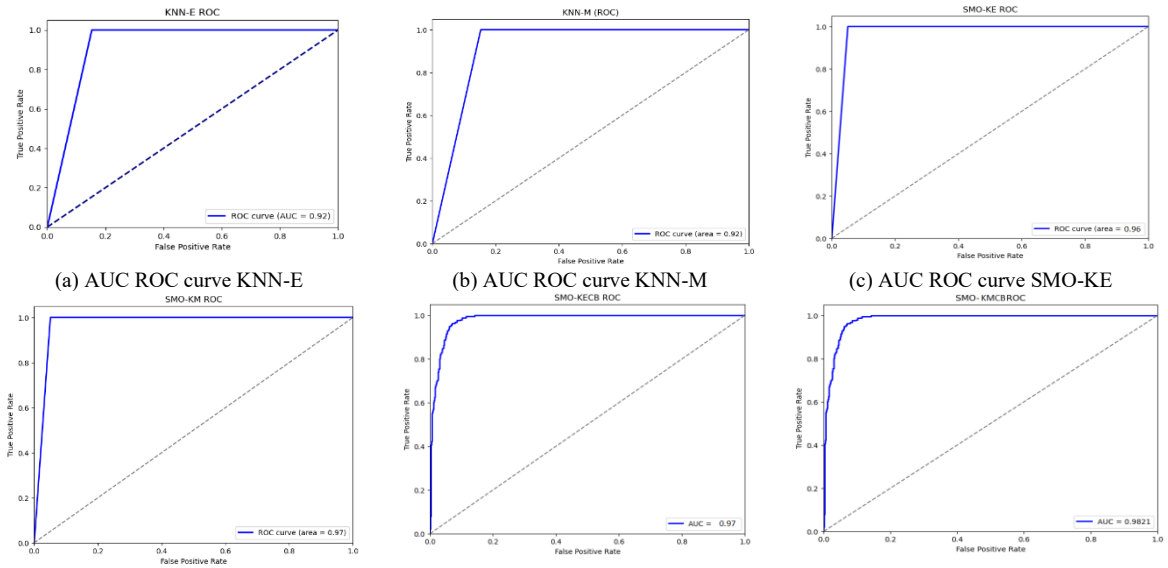


Fig. 7. Performance evaluation graph for dataset-3



(a) AUC ROC curve KNN-E (b) AUC ROC curve KNN-M (c) AUC ROC curve SMO-KE

(d) AUC ROC curve MSMO-KM (e) AUC ROC curve MSMO-KECB (f) AUC ROC curve MSMO-KMCB

Fig. 8. (a to f) AUC ROC curve of all the models using Dataset-3

Table 2. Summary of performance metrics for all the proposed models

Dataset-1 (5-fold CV; mean ± std)				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
KNN-E	0.802 ± 0.018	0.783 ± 0.021	0.799 ± 0.019	0.806 ± 0.020
KNN-M	0.812 ± 0.017	0.801 ± 0.020	0.810 ± 0.018	0.795 ± 0.022
MSMO-KE	0.829 ± 0.015	0.816 ± 0.017	0.835 ± 0.016	0.827 ± 0.017
MSMO-KM	0.846 ± 0.014	0.835 ± 0.016	0.819 ± 0.017	0.831 ± 0.016
MSMO-KECB	0.904 ± 0.011	0.914 ± 0.012	0.903 ± 0.011	0.911 ± 0.011
MSMO-KMCB	0.923 ± 0.010	0.923 ± 0.011	0.922 ± 0.010	0.914 ± 0.011
Dataset-2 (5-fold CV; mean ± std)				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
KNN-E	0.772 ± 0.022	0.768 ± 0.023	0.771 ± 0.022	0.754 ± 0.024
KNN-M	0.793 ± 0.021	0.791 ± 0.022	0.782 ± 0.022	0.781 ± 0.023
MSMO-KE	0.784 ± 0.021	0.782 ± 0.022	0.781 ± 0.022	0.773 ± 0.023
MSMO-KM	0.879 ± 0.014	0.873 ± 0.015	0.872 ± 0.015	0.872 ± 0.015
MSMO-KECB	0.834 ± 0.018	0.824 ± 0.019	0.833 ± 0.018	0.826 ± 0.019
MSMO-KMCB	0.894 ± 0.016	0.883 ± 0.017	0.892 ± 0.016	0.891 ± 0.016
Dataset-3 (5-fold CV; mean ± std)				
Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
KNN-E	0.922 ± 0.012	0.912 ± 0.013	0.921 ± 0.012	0.904 ± 0.014
KNN-M	0.926 ± 0.012	0.903 ± 0.014	0.913 ± 0.013	0.894 ± 0.015
MSMO-KE	0.963 ± 0.010	0.946 ± 0.011	0.952 ± 0.010	0.938 ± 0.012
MSMO-KM	0.975 ± 0.009	0.962 ± 0.010	0.971 ± 0.009	0.946 ± 0.011
MSMO-KECB	0.971 ± 0.009	0.963 ± 0.010	0.981 ± 0.008	0.956 ± 0.011
MSMO-KMCB	0.983 ± 0.008	0.972 ± 0.009	0.982 ± 0.008	0.972 ± 0.009

5.5. Ablation Study

An ablation study was performed to ascertain the distinct contribution of each component within the proposed framework by systematically eliminating or deactivating certain modules from the comprehensive MSMO-KMCB model. The analysis concentrates on assessing the effects of (i) MSMO-based optimization, (ii) adaptive leader influence, (iii) elite preservation, and (iv) CatBoost-based ensemble learning. The ablation study shows that adding more parts to the framework makes the predictive performance better and better. The change from the baseline KNN model (A1) to the standard SMO-optimized KNN (A2) shows how swarm-based hyper parameter optimization can help. Adding adaptive leader influence (A3) makes things even better by making the move from exploration to exploitation easier. Adding elite-preserving global leader reinforcement (A4) leads to even more benefits, showing that convergence stability has increased

and that the system is protected from getting stuck too soon. Finally, adding CatBoost as an ensemble learner to the whole MSMO-KMCB model (A5) gives the best results on all measures. This shows how well MSMO-based optimization and CatBoost work together. The ablation trends (Dataset-3 Best) listed in Table-3 show that each suggested change improves the final performance. The whole MSMO-KMCB architecture gives the strongest and most discriminative model.

Table 3. Ablation results using dataset-3 (5-fold CV, illustrative mean ± std)

Ablation Model	Accuracy	Precision	Recall	F1-Score
A1: KNN Baseline	0.921 ± 0.013	0.912 ± 0.014	0.919 ± 0.013	0.904 ± 0.015
A2: KNN + SMO	0.942 ± 0.011	0.941 ± 0.012	0.948 ± 0.011	0.934 ± 0.012
A3: KNN + MSMO (ALIC)	0.965 ± 0.010	0.954 ± 0.011	0.960 ± 0.010	0.946 ± 0.011
A4: KNN + MSMO (ALIC + EP-GLR)	0.973 ± 0.009	0.962 ± 0.010	0.970 ± 0.009	0.955 ± 0.010
A5: MSMO-KMCB (Full Model)	**0.982 ± 0.008**	**0.972 ± 0.009**	**0.982 ± 0.008**	**0.972 ± 0.009**

5.6. Convergence Analysis of MSMO Optimization

A convergence analysis of the MSMO method was done to see how stable and reliable the optimization process is. We looked at the convergence behaviour by keeping track of the best fitness value (measured by AUC) that was found throughout each iteration of the optimization procedure. Figure 9 shows how the optimum fitness value changes over time. Figure 9 shows that the optimization process improves quickly in the early iterations. This means that the algorithm does a good job of exploring the search space in the beginning. As good parameter settings are found, the best fitness value goes up a lot between the first few rounds. After about the fifth iteration, the pace of improvement starts to slow down, and the algorithm starts to settle. This means that the search process is moving from exploration to exploitation.

The convergence curve ultimately gets close to a steady fitness value of 0.983, which means that the algorithm finds the best or almost best parameter configuration in a minimal number of iterations. The MSMO algorithm only optimizes a few parameters, like the neighbourhood size k, the Minkowski distance parameter p, and the CatBoost learning rate. This makes the search space low-dimensional, which speeds up convergence. Also, the smooth convergence pattern shows that the algorithm doesn't have unstable oscillations or early divergence while it's searching. The stability of the convergence curve also shows that the chosen MSMO configuration is good. It has a population size of 10 and a maximum of 10 iterations, which is a good compromise between optimization effectiveness and computing cost. In general, the convergence study shows that the MSMO algorithm can optimize the suggested hybrid learning framework in a stable and reliable way while still being efficient in terms of computing.

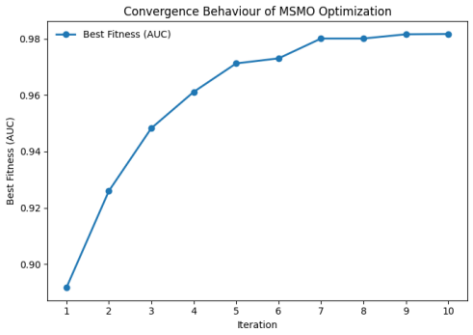


Fig. 9. Plot for convergence behaviour of the MSMO with the best fitness value (AUC) across iterations

5.7. Statistical Significance Analysis

A statistical significance study was done to make sure that the performance gains seen in the proposed models were not just due to random changes across folds. A 5-fold cross-validation framework was utilized to test the models; therefore, the performance ratings from each fold were used to evaluate the hypotheses. To see if the differences in performance across competing models were statistically significant, a Wilcoxon signed-rank test was performed. This is a common non-parametric statistical test for comparing machine learning models. The Wilcoxon test does not presume normality in the performance distribution, making it suitable for cross-validation comparisons with small sample sizes.

The null hypothesis $H0$ posits the absence of a statistically significant difference between the studied models, whereas the alternative hypothesis $H1$ asserts that the suggested MSMO-based ensemble models deliver enhanced predictive performance. The statistical tests were performed utilizing the fold-wise performance scores acquired during cross-validation. We used a significance level of α=0.05. Results showed that the suggested ensemble model MSMO-KMCB always made statistically significant improvements over the baseline KNN classifiers and the intermediate MSMO-optimized models throughout the datasets that were tested. Along with hypothesis testing, the mean and standard deviation values in Table-2 give confidence interval estimates for model stability. This shows that the observed gains are not due to huge differences between folds. The small standard deviations show that the proposed models are even more

consistent and strong. These statistical findings substantiate the conclusion that the performance enhancements realized by the MSMO-based hybrid and ensemble models are not only quantitatively superior but also statistically valid.

5.8. Computational Complexity and Runtime Considerations

When optimization-driven machine learning models are being explored for use in clinical decision-support systems, computational efficiency is just as crucial as predictive performance. The suggested framework combines neighbourhood-based categorization, swarm intelligence optimization, and ensemble learning. Because of this, it is important to know how much extra work each part adds to the computation.

The basic KNN classifier's computational cost is mostly based on the number of training samples N and the number of input features d. This is because classification has to find the distances between the query instance and all training samples in the feature space. So, the prediction complexity of KNN is about O(N×d). The baseline KNN classifiers are still easy to employ because the datasets used in this study have a moderate number of samples and characteristics. The MSMO approach requires more computing power because it has to check candidate parameter combinations over and over again. During optimization, several rounds are run on a number of potential solutions (spider monkeys) to find the best hyper-parameter values. But the suggested framework's optimization search field is limited to a low-dimensional parameter space with only three important parameters: the neighbourhood size k, the Minkowski distance parameter p, and the CatBoost learning rate. So, the optimization process quickly comes to a conclusion with a small setup (population size = 10 and maximum iterations = 10).

The suggested MSMO setup greatly decreases the amount of processing needed while still allowing for effective parameter exploration. This is in contrast to large-scale metaheuristic optimization frameworks that often need massive populations and hundreds of iterations. The convergence study in Section 5.6 backs up the idea that the optimization process stabilizes after only a few iterations, which shows that the search is working well.

The ensemble stage based on CatBoost adds an extra training step by using gradient boosting. But in the proposed architecture, the CatBoost model works using the prediction outputs of the improved KNN base learners instead of the original high-dimensional feature space. This stacking method cuts down on the number of inputs that the meta-learner has to deal with, which in turn cuts down on the extra computing costs. Also, the CatBoost settings utilized in this study (100 boosting iterations and a tree depth of 6) result in a moderate model size that strikes a balance between training speed and predictive power. In general, the suggested MSMO-based ensemble architecture strikes a good balance between the quality of the optimization and the cost of the computation. The framework can make better predictions without adding too much computational overhead because it has a low-dimensional search space, a limited optimization budget, and small meta-learning inputs. This makes it useful for real-world machine learning workflows and possible use in clinical decision-support systems.

5.9. Performance Comparison

In addition, the performance of the proposed models, MSMO-KE, MSMO-KM, MSMO-KECB and MSMO-KMCB has been systematically evaluated and compared with existing studies conducted on heart datasets. Table-4 provides a comprehensive comparison of the KNN-E, KNN-M models and proposed models-1, 2 ,3 and 4 with some existing models. The first study [1] utilized the SOLSSA-CatBoost model and achieved an accuracy of 90%. The study [3], employed the CatBoost model and achieved an accuracy of 82.5%. The study [9], used an Ensemble model and achieved an accuracy of 88.7%, and study [14], used the CatBoost model with SHAP and achieved an accuracy of 84.4%.

In the present research study, three distinct datasets have been utilized to check the generalizability of the proposed models. The accuracies of KNN-E, KNN-M, MSMO-KE, MSMO-KM, MSMO-KECB and MSMO-KMCB are 0.80, 0.81, 0.82, 0.84, 0.90, and 0.92 respectively using dataset-1. Using dataset-2, the accuracy values achieved were 0.77, 0.791, 0.78, 0.876, 0.83, and 0.89, respectively, and at last, using dataset-3 values of accuracy achieved were -0.92, 0.923, 0.961, 0.974, 0.97, and 0.982. All the models exhibited their best performance with dataset 3.

Further, a comparison of the proposed models with existing studies has been conducted based on techniques, dataset accuracy, and dataset size to highlight the significance of the proposed models in the heart disease prediction problem, establishing it as a noteworthy advancement in the field. A detailed examination of Table-4 reveals that the MSMO-KMCB model not only outperforms existing approaches but also demonstrates its potential for real-world generalization, achieving an impressive accuracy rate of 98.2%. The performance reported in Table 4 was taken from the research that used varied datasets, sample sizes, feature sets, and experimental techniques. So, the accuracy values in the table should not be seen as straight comparisons of one thing to another. The table is meant to give an overview of modern methods used to forecast cardiovascular disease in context, not to show the best ones. The main goal of the table is to show where the suggested MSMO-based models fit into the larger research landscape, not to say that they are better than all other models across different datasets.

Table 4. Comparison with existing studies

Study	Number of Dataset	Name of Dataset	Size of Dataset (No. of Patients)	Best Accuracy (%)
[1]	02	CVD Dataset 1, Kaggle	Not Revealed	90
		CVD Dataset 2, Kaggle	Not Revealed	81.51
[3]	01	CHD Dataset (Self-prepared)	2642	82.5
[9]	01	The Kaggle repository's cardiovascular disease dataset	70000	88.70
[14]	01	Cleveland dataset.	303	84.4
MSMO-KE	03	Dataset-1	4000	0.82
MSMO-KM				0.84
MSMO-KECB				0.90
MSMO-KMCB				0.92
MSMO-KE		Dataset-2	1025	0.78
MSMO-KM				0.876
MSMO-KECB				0.83
MSMO-KMCB				0.89
MSMO-KE		Dataset-3	5110	0.961
MSMO-KM				0.974
MSMO-KECB				0.97
MSMO-KMCB				0.982

6. Conclusions

Cardiovascular disease continues to be a significant worldwide health issue, with prompt and precise risk assessment being essential for enhancing patient outcomes and decreasing healthcare expenses. This study developed a collection of hybrid machine learning frameworks that merge neighborhood-based categorization, adaptive swarm optimization, and ensemble learning for early heart disease prediction, motivated by the need for reliable and generalizable prediction models. Four models—MSMO-KE, MSMO-KM, MSMO-KECB, and MSMO-KMCB—were created by using Modified Spider Monkey Optimization (MSMO) and K-Nearest Neighbor classifiers with Euclidean and Minkowski distance measures. These optimized classifiers were then improved even more through CatBoost-based ensemble learning. We used a cross-validation method on three different cardiovascular datasets to test the suggested models and make sure they were strong and could be used in more situations. Experimental results show that all MSMO-based models regularly do better than baseline classifiers and a number of other state-of-the-art techniques on a number of performance parameters. MSMO-KMCB had the best predictive performance of all the models tested, especially on Dataset-3. This shows how well adaptive optimization and ensemble learning work together.

In general, the suggested framework helps improve the prediction of cardiovascular illness by offering strong, scalable, and high-performing models that may be used in clinical decision-support systems. Even though the results are encouraging, the study is limited because it used old data sets and did not have real-time physiological signs. Future endeavors will concentrate on the integration of data from wearable IoT-based sensors, including ECG signals, to facilitate real-time monitoring and prediction, thereby broadening the application of the presented models to continuous and distant healthcare environments.

All the Declarations and Statements

Author Contributions Statement

Ankit Maithani: Data Curation and Software Implementation: Handled data acquisition, dataset preprocessing, and implementation of the research model, Formal Analysis, Visualization, Writing – Drafted the initial manuscript.
Garima Verma: Statistical Analysis: Performed in-depth analysis of experimental results, prepared performance charts, and ensured the statistical robustness of the evaluation, writing – Review and Editing, and Project Management: Reviewed and edited the manuscript, ensured clarity and coherence, and helped coordinate project milestones and deadlines.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflicts of interest.

Funding Declaration
None

Data Availability Statement
There are three datasets utilized for the study. Dataset-1 (https://www.kaggle.com/datasets/aasheesh200/framingham-heart-study-dataset), Dataset-2 (https://www.kaggle.com/code/mustafaalkhazraji93/end-to-end-heart-disease-classification) and dataset-3 (https://www.kaggle.com/fedesoriano/stroke-prediction-dataset) are publicly available datasets which can be accessed using the given links.

Ethical Declarations
None

Acknowledgments
We sincerely thank the experts for their professional evaluation and valuable recommendations, which have contributed to improving the quality of the experiment and the reliability of its results.

Declaration of Generative AI in Scholarly Writing
The author utilized Grammarly (v2024) and Quillbot (2024 release) solely to refine the wording and make the content more straightforward. These tools were not utilized to analyze data, write code, create figures, or obtain results. The author meticulously scrutinized and validated all AI-assisted modifications, retaining ultimate accountability for the manuscript's contents.

Abbreviations
The following abbreviations are used in this manuscript:

ML - Machine Learning
MSMO - Modified Spider Monkey Optimization
CVD - Cardiovascular disease
BMI - Body mass index
PII - Personally identifiable information
SMO - Spider Monkey Optimization
LL - Local leader
GL- Global leader
ALIC - Adaptive Leader Influence Control
EP-GLR - Elite-Preserving Global Leader Reinforcement
AUC- Area under the curve

References

[1] X. Wei, C. Rao, X. Xiao, L. Chen, and M. Goh, “Risk assessment of cardiovascular disease based on SOLSSA-CatBoost model,” Expert Systems with Applications, vol. 219, p. 119648, 2023. https://doi.org/10.1016/j.eswa.2023.119648
[2] M. U. Rehman, S. Naseem, A. U. R. Butt, T. Mahmood, A. R. Khan, I. Khan, J. Khan, and Y. Jung, “Predicting coronary heart disease with advanced machine learning classifiers for improved cardiovascular risk assessment,” Scientific Reports, vol. 15, no. 1, p. 13361, 2025. https://www.nature.com/articles/s41598-025-96437-1
[3] A. Kumar, P. R. U, and M. Maben, "Heart disease prediction using machine learning algorithms," in Proc. 3rd Int. Conf. Networks, Multimedia and Information Technology (NMITCON), Aug. 2025, pp. 1-6. doi: 10.1109/NMITCON65824.2025.11187908.
[4] D. Asif, M. Bibi, M. S. Arif, and A. Mukheimer, “Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization,” Algorithms, vol. 16, no. 6, p. 308, 2023. https://doi.org/10.3390/a16060308
[5] A. Maithani, R. Koli, R. Pal, D. S. Rai, A. Rohilla, and R. Kumar, “Agile approach of predicting cardiac disease using ANN based on feature selection,” in Proc. Int. Conf. Innovative Practices in Technology and Management (ICIPTM), Feb. 2021, pp. 243–248. https://doi.org/10.1109/ICIPTM52218.2021.9388325
[6] A. Maithani and G. Verma, "Hybrid model with improved score level fusion for heart disease classification," Multimedia Tools and Applications, vol. 83, no. 18, pp. 54951-54987, 2024. doi: 10.1007/s11042-023-17599-6.
[7] D. Rohan, G. P. Reddy, Y. P. Kumar, K. P. Prakash, and C. P. Reddy, “An extensive experimental analysis for heart disease prediction using artificial intelligence techniques,” Scientific Reports, vol. 15, no. 1, p. 6132, Feb. 2025. https://www.nature.com/articles/s41598-025-90530-1
[8] C. J. Harrison and C. J. Sidey-Gibbons, “Machine learning in medicine: A practical introduction to natural language processing,” BMC Medical Research Methodology, vol. 21, no. 1, pp. 1–11, 2021. https://doi.org/10.1186/s12874-021-01347-1
[9] A. Alqahtani, S. Alsubai, M. Sha, L. Vilcekova, and T. Javed, “Cardiovascular disease detection using ensemble learning,” Computational Intelligence and Neuroscience, vol. 2022, Art. no. 5267498, 2022. https://doi.org/10.1155/2022/5267498
[10] G. Verma and S. Kanrar, “Load balancing model for cloud environment using swarm intelligence technique,” Multiagent and Grid Systems, vol. 19, no. 3, pp. 211–229, 2023. https://content.iospress.com/articles/multiagent-and-grid-systems/mgs230021

[11] J. Kaur, B. S. Khehra, and A. Singh, "A state of the art survey of soft computing based approaches for heart disease detection and development of a novel PSO-fuzzy approach," Archives of Computational Methods in Engineering, vol. 33, no. 2, pp. 1841-1864, 2026. doi: 10.1007/s11831-025-10360-2.
[12] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," IEEE Access, vol. 7, pp. 81542–81554, 2019. https://doi.org/10.1109/ACCESS.2019.2923707
[13] P. Rani, R. Kumar, N. M. S. Ahmed, and A. Jain, "A decision support system for heart disease prediction based upon machine learning," Journal of Reliable Intelligent Environments, vol. 7, no. 3, pp. 263–275, 2021. https://doi.org/10.1007/s40860-021-00133-6
[14] F. Ahmed, M. Saleem, Z. Rajpoot, and A. Noor, "Intelligent heart disease prediction using CatBoost empowered with XAI," International Journal of Computational and Innovative Sciences, vol. 2, no. 4, pp. 8–13, 2023.
[15] B. Dhananjay and J. Sivaraman, "Analysis and classification of heart rate using CatBoost feature ranking model," Biomedical Signal Processing and Control, vol. 68, p. 102610, 2021. https://doi.org/10.1016/j.bspc.2021.102610
[16] I. D. Mienye, Y. Sun, and Z. Wang, "An improved ensemble learning approach for the prediction of heart disease risk," Informatics in Medicine Unlocked, vol. 20, p. 100402, 2020. https://doi.org/10.1016/j.imu.2020.100402
[17] Y. Shaikh, V. K. Parvati, and S. R. Biradar, "Heart disease prediction using ensemble learning," in Proc. Int. Conf. Applied Intelligence and Sustainable Computing (ICAISC), Jun. 2023, pp. 1–5. https://doi.org/10.1109/ICAISC58445.2023.10200375
[18] A. B. Majumder, S. Gupta, D. Singh, B. Acharya, V. C. Gerogiannis, A. Kanavos, and P. Pintelas, "Heart disease prediction using concatenated hybrid ensemble classifiers," Algorithms, vol. 16, no. 12, p. 538, 2023. https://doi.org/10.3390/a16120538
[19] M. S. Pathan, A. Nag, M. M. Pathan, and S. Dev, "Analyzing the impact of feature selection on the accuracy of heart disease prediction," Healthcare Analytics, vol. 2, p. 100060, 2022. https://www.sciencedirect.com/science/article/pii/S2772442522000235
[20] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: Overview study and experimental results," in Proc. 11th Int. Conf. Information and Communication Systems (ICICS), Apr. 2020, pp. 243–248. https://doi.org/10.1109/ICICS49469.2020.239556
[21] S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (KNN) approach for predicting economic events: Theoretical background," Int. J. Engineering Research and Applications, vol. 3, no. 5, pp. 605–610, 2013. https://api.semanticscholar.org/CorpusID:15532755
[22] A. Garg, B. Sharma, and R. Khan, "Heart disease prediction using machine learning techniques," in IOP Conf. Series: Materials Science and Engineering, vol. 1022, no. 1, p. 012046, 2021. https://doi.org/10.1088/1757-899X/1022/1/012046
[23] A. Kooshari, M. Fartash, P. Mihannezhad, M. Chahardoli, J. Akbari Torkestani, and S. Nazari, "An optimization method in wireless sensor network routing and IoT with water strider algorithm and ant colony optimization algorithm," Evolutionary Intelligence, 2023. https://doi.org/10.1007/s12065-023-00847-x
[24] L. Wu, X. Huang, J. Cui, C. Liu, and W. Xiao, "Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot," Expert Systems with Applications, vol. 215, p. 119410, 2023. https://doi.org/10.1016/j.eswa.2022.119410
[25] G. Verma, "Load balancing in cloud environment using opposition-based spider monkey optimization," Wireless Personal Communications, vol. 137, no. 2, pp. 977–996, 2024. https://link.springer.com/article/10.1007/s11277-024-11445-0
[26] R. Anudeep and S. J. J. Thangaraj, "Accurate prediction of myocardial infarction by comparing logistic regression algorithm with CatBoost classifier," in E3S Web of Conferences, vol. 399, p. 04019, 2023. https://doi.org/10.1051/e3sconf/202339904019
[27] P. Anuradha and V. K. David, "Feature selection and prediction of heart diseases using gradient boosting algorithms," in Proc. Int. Conf. Artificial Intelligence and Smart Systems (ICAIS), Mar. 2021, pp. 711–717. https://doi.org/10.1109/ICAIS50930.2021.9395819
[28] A. März, "CatBoostLSS—An extension of CatBoost to probabilistic forecasting," arXiv preprint, arXiv:2001.02121, 2020. https://doi.org/10.48550/arXiv.2001.02121
[29] S. G. Taj and K. Kalaivani, "Hybrid prediction model with improved score level fusion for heart disease diagnosis," Computational Biology and Chemistry, p. 108278, 2024. https://www.sciencedirect.com/science/article/abs/pii/S1476927124002664
[30] M. A. Bouqentar, O. Terrada, S. Hamida, S. Saleh, D. Lamrani, B. Cherradi, and A. Raihani, "Early heart disease prediction using feature engineering and machine learning algorithms," Heliyon, vol. 10, no. 19, 2024. https://www.cell.com/heliyon/fulltext/S2405-8440(24)14762-7
[31] S. Arif, S. H. Son, H. Y. Kim, S.-C. Kim, and J. Y. Lee, "A diagnosis tool for early detection and classification of heart disease in individuals using transformer mechanisms," Computer Methods and Programs in Biomedicine, p. 109248, 2026. https://www.sciencedirect.com/science/article/abs/pii/S0169260726000167
[32] O. Rainio, J. Tamminen, M. S. Venäläinen, J. Liedes, J. Knuuti, J. Kemppainen, and R. Klén, "Comparison of thresholds for a convolutional neural network classifying medical images," International Journal of Data Science and Analytics, vol. 20, no. 3, pp. 2093–2099, 2025. https://link.springer.com/article/10.1007/s41060-024-00584-z

Authors' Profiles



Ankit Maithani working as Assistant Professor in School of Computing, DIT University, Dehradun, INDIA. Ankit is pursuing his Ph.D in Computer Science Engineering from DIT University & completed his M.Tech in Computer Science Engineering from DIT University in 2017. Ankit's research interest includes Machine Learning & Artificial Intelligence.



Dr. Garima Verma working as Assistant Professor in the School of Computing, DIT University, INDIA. Garima Verma received her Ph.D in Computer Science Engineering from DIT University in 2018 and M.Tech in Computer Science Engineering from Uttarakhand Technical University in 2011. Garima has Qualified UGC Net in Computer Science and received a Gold Medal (First Merit) in M.Tech. Garima's research interest includes cloud computing, Blockchain, and machine learning.

How to cite this paper

Ankit Maithani, Garima Verma, "Adaptive Swarm-Optimized Ensemble Learning for Generalizable Heart Disease Risk Prediction", International Journal of Information Technology and Computer Science(IJITCS), Vol.18, No.3, pp.109-129, 2026. DOI:10.5815/ijitcs.2026.03.08