# Optimizing Heart Disease Prediction Using A Hybrid Dynamic Swarm Evolution Approach

Vanssh Parikh, Bhoomi Sharma, Arnav Byotra, Amarjit Malhotra*

Department of Information Technology, Netaji Subhas University of Technology, New Delhi, India.

*Corresponding author(s). E-mail(s): amarjit.malhotra@nsut.ac.in;
Contributing authors: parikharyan30@gmail.com;
bhoomisharma107@gmail.com; arnavbyotra19@gmail.com;

**Abstract**

Early identification of Cardiovascular issues is crucial due to the high mortality rate associated with heart disease.This research assesses machine learning algorithms for heart disease prediction using tabular datasets.Previous researchers suggested Tree-based models like Random Forest, XGBoost, and Decision Trees and various hyper-parameter optimization methods including Grid Search, Random Search, Swarm and Evolutionary Algorithms which excel in accuracy and robustness. However, they are computationally inefficient and less effective in dynamic settings. A novel Hybrid Swarm Evolution Optimization (HySEOpt) is introduced, which adjusts mutation rates based on performance curves and utilizes parallel processing for faster optimization achieving 98.01% accuracy. HySEOpt enhances model's quality and robustness, addressing limitations of existing methods and contributes to hyper-parameter optimization in predictive healthcare modeling.

**Keywords:** Hyper-parameter Optimisation; Evolutionary Algorithm; Swarm Algorithm; Heart Disease Prediction; Hybrid Swarm Evolution.

# 1 Introduction

Heart disease remains one of the major global health issues responsible for enormous human deaths across the world. Heart disease detection is of utmost importance if proper and timely treatment is going to be administered for better results. Recent innovative technologies in machine learning and data analytics bring new dimensions for the potential to improve heart disease prediction. Researchers and other medical practitioners are moving towards the latest technologies to build more accurate and reliable predictive models. This shows tremendous potential for revolutionizing the detection and management of heart diseases, resulting implications for patient care. Machine learning and artificial intelligence, as they are known, make huge impacts across many disciplines. Healthcare is no different. For heart disease, these technologies change the game. This way, building predictive models using machine learning, we are able to train such algorithms using vast amounts of data: patient histories, genetic information, lifestyle habits, and test results. Such algorithms may find relationships that would elude a doctor's mind, hence enabling doctors to identify heart disease at an early stage and with greater precision. The result is that doctors are able to catch heart disease earlier and treat it more effectively, potentially saving many lives.

The motivation to carry out this project came with the need for improving the predictive models of heart disease using various machine learning techniques. Accurate diagnosis of the condition is important for better treatment outcomes and alleviating the burden on the healthcare system. By employing state-of-the-art algorithms and methodologies in the project, better models will be developed to more precisely measure heart diseases and predict early for individuals .This will help various patients to know about their disease before it further worsens thereby saving lives.

The objective of this work is to significantly improve heart disease prediction accuracy by integrating the novel HySEOpt hyper-parameter optimization technique with tree-based classifiers like RandomForest. By employing a cutting-edge optimization method to enhance model performance beyond traditional approaches, leading to more accurate risk identification and early intervention. This innovative combination aims to advance predictive accuracy and has the potential to positively impact patient outcomes and the broader field of heart disease prediction.

# 2 Literature Survey

Advancement in the heart disease detection process has also evolved in recent years due to innovations in medical imaging, wearable devices, and genomic sequencing. Techniques of Artificial Intelligence (AI) revolutionized predictive modeling and brought about more precise and personalized diagnosis. Together with the data preprocessing and ensemble methods, the enhancements improved the outcomes for patients in cardiovascular healthcare.

## 2.1 Analysis of multiple algorithms based methods

Dhai et al. [1] researched heart disease prediction from a data analytics perspective. Three distinct algorithms were used, namely: neural networks, Support Vector

Machine (SVM), and K-Nearest Neighbors (KNN). The study included datasets of varying sizes to evaluate the performance, accuracy, and stability of each algorithm. As was to be expected, the neural networks had a better ease of configuration, and, more importantly, they achieved an accuracy of 93%, which was quite commendable.

Muhammad et al. [2] felt the urge to diagnose heart diseases early and accurately. As part of the study, an array of machine learning classification algorithms was explored, including KNN, Decision Tree (DT), Extra Trees (ET), Gradient Boosting (GB), Random Forest (RF), SVM, AdaBoost (AB), Naive Bayes (NB), and Logistic Regression (LR). Moreover, to refine the dataset, feature selection was applied besides the exploration of algorithms. Therefore, to gauge the effectiveness of the developed predictive model, performance metrics such as accuracy, sensitivity, specificity, and Area Under The Curve(AUC) were analyzed carefully.

Hossain et al. [3] examined the heart disease diagnosis complications, especially in resource-constrained settings. The study reviewed different clinical decision support systems. In addition to predicting heart diseases using algorithms such as Naïve Bayes, Decision Tree, and Artificial Neural Network, the research found that it was not possible to achieve significant success in predictive modeling; more enhanced models integrating diverse sources of data are needed.

Sharma et al. [4] focused on addressing the ever-growing burden of heart-related ailments. They developed a machine learning model using a benchmark dataset. Multiple algorithms were used, such as Random Forest, SVM, Naive Bayes, and Decision Tree. The study also explored the correlation between different attributes in the dataset for better accuracy in the prediction and finally presented Random Forest as the most accurate algorithm for the prediction of heart diseases.

Bhatt et al. [5] had the main aim of classifying heart diseases using different models and a real-world dataset. The k-modes clustering algorithm was applied to a dataset of patients with heart diseases for the prediction of the diseases. In this study, through rigorous preprocessing of the data and evaluation of the models, it was revealed that the Multi-Layer Perceptron (MLP) model performed the best with a high accuracy of 87.23%. The results put emphasis on the k-modes clustering algorithm for the accurate prediction of heart diseases, indicating its significance in developing specially targeted diagnostic and treatment strategies.

While exploring the multiple algorithms, in the next section, focused analysis on specific categories of algorithms comes into place. This includes a clear pattern that comes out regarding the superiority of some approaches in the prediction of heart diseases. While initial studies have delved deep into the multitude of machine learning algorithms, later studies suggest dominance by tree-based approaches.

### 2.2 Dominance of Tree based algorithms

Chandrashekhar et al. [6] researched the identification of early cardiovascular issues. In this paper, the application of Random Forest, K-nearest neighbor, and AdaBoost machine learning techniques were presented. Random Forest and AdaBoost were models with high accuracy. An ensemble classifying model was used, and the results were better than the individual results of the models. The tree-based models were proven to give better accuracy in heart disease prediction.

Patidar et al. [7] compared the performance of various supervised machine learning algorithms for heart disease prediction. In this paper, many experiments were carried out with algorithms such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Gaussian Naïve Bayes, and k-Nearest Neighbor. Random Forest was the best performer, with an accuracy of 98.53%. One of the main reasons for this was that one-hot encoding was used for pre-processing.

Nayeem, et al. [8] focused on the prediction of the existence of heart disease using supervised machine learning algorithms. This research identified the most accurate classifier: Random Forest, which achieved a classification accuracy of 95.63%. Using feature selection and handling null values, this paper highlighted that Random Forest had the ability to predict heart diseases with high accuracy.

Marelli et al. [9], proposed the use of machine learning algorithms in the identification of congenital heart defects. Their study illustrated models such as Random Forest and XGBoost. In the study, feature selection and cross-validation were used and XGBoost was the best performer. This showed that tree-based models were better at accurately predicting heart conditions. The research of Nissa et al. [10] supported the need for predictive models with correctness. By relying on state-of-the-art boosting techniques and machine learning methodologies, such as CatBoost, Random Forest, Gradient Boosting, Light GBM, and AdaBoost, the research aimed to design robust models which would reliably forecast cardiovascular health risks. The empirical findings identified AdaBoost as the top model, achieving significant accuracy in most metrics and adding valuable insights to the field of prediction of cardiovascular health.

Superiority of Tree-Based Algorithms: The discussion shall now transition toward the superiority of tree-based algorithms, where a model such as Random Forest becomes the front runner for accuracy and reliability. This transition, therefore, points out a shift toward a more subliminal understanding of the efficacy of algorithms where tree-based classifiers have been found to depict better performance in most studies. This continuity in findings provokes a general agreement among the research fraternity on the efficacy of particular algorithmic approaches and charts a course for further research in the domain of cardiovascular health prediction.

## 2.3 HPO techniques in Heart Disease Prediction

Valarmath et al. [11] proposed a prediction system for heart disease detection. Their work utilized Hyper-Parameter Optimization techniques such as Grid Search, Randomized Search, and Genetic programming. Through the use of hyper-parameter optimization (HPO) on random forests and XGBoost classifiers, the work achieved high accuracy that was superior to previous studies. The research brought out the importance of HPO in improving predictive performance for the detection of heart disease.

Asif et al. [12] designed a machine learning model that incorporated several preprocessing techniques and hyper-parameter optimization methods along with ensemble learning algorithms in order to improve heart disease prediction. They merged three Kaggle datasets of similar features for validation of model performance. In this study, the Extra Tree Classifier was used, with data normalization and Grid Search Cross Validation (CV) for hyper-tuning the parameters, and an 80:20 split of the test dataset for testing and training, respectively. Such an approach seemed to be promising in providing predictions on the presence or absence of heart disease.

All these combined studies show the varied approaches and methodologies in the area of heart disease prediction, demonstrating the efficacy of machine learning techniques in the use of tree-based algorithms, k-modes clustering, and HPO to improve diagnostic accuracy and predictive performance.A brief of the literature survey on recent heart disease prediction studies is given in table 1.

An overview of heart disease prediction methodologies unravels the fast-changing landscape of machine learning applications in healthcare. While the studies above provide a rich view of the efficacy of algorithms and approaches, they also underscore the important task of model parameter optimization for further enhancement of predictive performance. Moving from the analysis of algorithm effectiveness, the thesis now proceeds to the area of hyper-parameter tuning techniques. By fine-tuning model parameters, and optimizing the algorithmic configurations, HPO techniques aim at overcoming the limitations and challenges reflected in the previous studies and thereby take forward the field of heart disease prediction towards more accuracy and reliability.

## 2.4 Early Methods and Their Limitations

The earlier machine learning techniques used Grid Search for hyper=parameter optimization. Later, this method was found to be poor in computational complexity, it lacked the exploration of search space, was not efficient in resources, and was dependent on user expertise. Bergstra et al. [13] conducted a study to show that random trials proved more efficient compared to grid search for Hyper-parameter optimization. They found that random search discovered models of similar or better quality in less time. Hence, it becomes a baseline for progress in Hyper-parameter optimization techniques. However, random search is unreliable to train deep belief network) as seen in Maclaurin et al study. [14] because of the difficulty in adjusting Hyper-parameters without gradients. Gradients derived via stochastic gradient descent with momentum optimize tens of thousands of Hyper-parameters, such as steps size, momentum schedules, weight initialization distributions, and neural network architectures.

## 2.5 Bayesian Optimization and Gradient-Based Optimization

With the progress of the field, the invention of Bayesian Optimization later came into being. In addressing the challenge of tuning learning parameters and model Hyper-parameters in machine learning algorithms, Snoek et al. [15] went for Bayesian optimization where a Gaussian process models generalization performance. They introduced the concept of modeling the objective function as a probabilistic surrogate, giving guidance on concentrating the search on promising areas of the Hyper-parameter space. This proved to be most effective for black-box optimization problems. In the development of a fast Hyper-parameter optimization algorithm, Hazan et al. [16] drew inspiration from Boolean function analysis. They presented a spectral approach to Hyper-parameter optimization, showing the advantages of gradient methods when it comes to the optimization of Hyper-parameters. Their approach used uniform sampling over the Hyper-parameters and was massively parallelizable, using compressed

**Table 1**: Comprehensive Overview of Literature Survey on Heart Disease Prediction Techniques.

| Researcher | Algorithm/Approach Used | Advantages | Disadvantages |
|---|---|---|---|
| Dhai et al. [1] | Neural Networks, SVM, KNN | Ease of configuration, 93% accuracy | May require significant computational resources |
| Muhammad et al. [3] | KNN, DT, ET, GB, RF, SVM, AB, NB, LR | Comprehensive evaluation, feature selection | Complexity in model interpretation |
| Hossain et al. [2] | Random Forest, Decision Tree, Artificial Neural Network | Clinical decision support systems review | Limited success, need for more complex models |
| Sharma et al. [4] | Random Forest, SVM, Naive Bayes, Decision Tree | Benchmark dataset usage, correlation analysis | Potential overfitting, dataset bias |
| Bhatt et al. [5] | k-modes clustering | High accuracy, MLP model performance | Limited scalability, dependence on initial parameters |
| Chandrashekhar et al. [6] | Random Forest, K-nearest neighbor, AdaBoost | High accuracy, ensemble classifier effectiveness | Potential overfitting, parameter sensitivity |
| Patidar et al. [7] | Logistic Regression, Decision Tree, Random Forest, SVM, Gaussian Naïve Bayes, k-Nearest Neighbor | Top performance of Random Forest | Computational complexity, Hyperparameter tuning requirements |
| Nayeem et al. [8] | Random Forest | High classification accuracy | Sensitivity to imbalanced datasets, computational overhead |
| Marelli et al. [9] | Random Forest, XGBoost | Effective congenital heart defect identification | Model complexity, computational resources |
| Nissa et al. [10] | CatBoost, Random Forest, Gradient Boosting, Light GBM, AdaBoost | Robust predictive models, AdaBoost performance | Model interpretability, computational intensity |
| Valarmath et al. [11] | Random Forest, XGBoost, Grid Search CV | High accuracy with HPO techniques | Computational cost, tuning complexity |
| Asif et al. [12] | Extra Tree Classifier, Grid Search CV | Enhanced predictive performance | Potential for overfitting, dataset bias, computational cost |

sensing techniques for orthogonal polynomials. Its performance is reported on deep neural network training with significant improvements over state-of-the-art tools, achieving faster overall runtimes and outperforming Random Search by eightfold.

## 2.6 Evolutionary Algorithms and Population Based Training

Soon enough, Evolutionary Algorithms joined. Freitas et al. [17] investigated the application of evolutionary algorithms in data mining tasks. EAs, inspired by neo-Darwinian evolution, are robust and adaptive global search techniques. They overviewed the application of Genetic Algorithms and Genetic Programming to classification rule discovery, clustering, attribute selection, and construction. They also explored the application of Multi-Objective EAs in various data mining tasks. Later, Jaderberg et al. [18] introduced Population Based Training (PBT), an asynchronous optimization algorithm for neural networks. PBT could efficiently exploit a fixed computational budget in order to concurrently optimize models and their Hyperparameters, which results in faster convergence and higher final performance across many domains, including deep reinforcement learning, supervised learning for machine translation, and training of Generative Adversarial Networks. Ho et al. [19] proposed the Population-Based Algorithm(PBA), a novel population-based data augmentation algorithm that generates non stationary augmentation policy schedules. PBA achieved comparable performance to Auto Augment on all datasets: CIFAR-10, CIFAR-100, and SVHN, with orders of magnitude less overall compute. On the CIFAR-10 dataset, they achieved a mean test error of 1.46%, a slight improvement over the current state-of-the-art.

## 2.7 Automated Machine Learning Frameworks

In the era of AutoML frameworks, Real et al. [20] introduced AutoML-Zero—a framework that automated the discovery of complete machine learning algorithms using basic mathematical operations as building blocks. Their approach reduced human bias considerably with the use of a generic search space. Despite its vastness, evolutionary search discovered two-layer neural networks trained by back propagation, outperforming them by directly evolving on tasks such as CIFAR-10 variants, to reveal modern techniques like bi-linear interactions and Normalized gradients. Furthermore, evolution adapted algorithms to different task types, hence promising developments in AutoML. Zoph et al. [21] proposed Neural Architecture Search with Reinforcement Learning, an approach for the automatic design of neural network architectures. Their approach used reinforcement learning to train a recurrent neural network (RNN) that generated model descriptions, achieving very impressive results. They achieved a test error rate of 3.65% on CIFAR-10, which is competitive with the best human-invented architectures and improved on the previous state-of-the-art by 0.09 percent. On the Penn Tree bank dataset, their novel recurrent cell outperformed widely-used Long Short-Term Memory cells and obtained a state-of-the-art perplexity of 62.4.

## 2.8 Ensemble Methods and Multi-Fidelity Optimization

Ensemble Methods introduced the concept of model aggregation. Shahhosseini et al. [22] proposed an optimization-based approach, Generalized Weighted Ensemble with Internally Tuned Hyper-parameters, for optimizing ensemble weights and hyper-parameters of machine learning models in regression problems. Their method combined ensembles through weighted averages and tuned hyper-parameters of each base learner within the optimization process. They utilized Bayesian search to accelerate optimization and implemented a heuristic for generating diverse and effective base learners. The algorithm demonstrated generalizability to real datasets through analyses with ten publicly available datasets. Multi-Fidelity Optimization aimed to optimize the trade-off between accuracy and computational cost. Folch et al. [23] proposed a novel Bayesian Optimization algorithm that combined multi-fidelity modeling and asynchronous batch methods. Their approach addressed challenges related to laboratory experiments—for example, designing battery design measurements—where measurements may come from different sources, and evaluations may require significant waiting times. Empirical studies demonstrated that the algorithm outperformed single-fidelity batch methods and multi-fidelity sequential methods. As an application, they considered designing electrode materials for optimum performance in pouch cells using experiments with coin cells to approximately compute performance.

## 2.9 Novel Approaches in Optimizing Hyper-parameters

Diaz et al. [24] formulated the optimization of parameters for a neural network as a box-constrained mathematical optimization problem. They used a derivative-free optimization tool that guided this search by means of a radial basis function model of the accuracy in prediction of the neural network. All candidate configurations were trained for a small number of epochs, and only the most promising ones were fully trained. They showed the effectiveness of their methodology with benchmark data sets and with the prediction of drug-drug interactions. Another contribution was that the optimization tool used was open-source, allowing access and replication by third parties. Going into another innovative approach, Liu et al.[25] proposed DARTS, Differentiable Architecture Search. The scalability challenge of architecture search was due to a non-differentiable nature of the search problem, which DARTS resolves. This formulation allows the search for architectures by using gradient descent on a continuous relaxation of the architecture representation. DARTS discovered state-of-the-art convolutional and recurrent architectures on image classification and language modeling tasks, with considerably fewer computations than non-differentiable state-of-the-art methods. Their implementation was publicly available to support further research. Afterward, Li et al. [26] introduced Hyperband, a new approach to hyper-parameter optimization. They concentrate on the acceleration of random search through adaptive resource allocation and early stopping. Hyper-parameter optimization was formulated as a pure-exploration, non-stochastic, infinite-armed bandit problem. Their algorithm, Hyperband, provides several desirable guarantees and is more effective than several state-of-the-art Bayesian optimization methods, achieving over an order-of-magnitude speedup on a variety of learning problems.

Table 2 gives a brief of the recent advancements in hyper parameter optimization techniques and their advantages and disadvantages .

### 2.10 Dynamic and Adaptive Hyper-parameter Optimization

Luo et al. [27] presented AdaBound and AMSBound, novel variants of Adam and AMSGrad, respectively. These variants solved an extreme learning rate of adaptive optimization, leading to poor generalization, through dynamic bounds on learning rates. They also provided the theoretical proofs for convergence and showed that the new variants eliminate the generalization gap between adaptive methods while keeping up higher learning speed early in training, particularly on complex deep networks.Ali et al. [28] addressed the challenge of dynamically fine-tuning hyper-parameters of a machine learning algorithm efficiently. They reviewed the various optimization strategies, including Random Search and Grid Search, and proposed to employ Support Vector Machine models optimized by Ant Bee Colony Algorithm, Genetic Algorithm, Whale Optimization, and Particle Swarm Optimization. Their study compared the computational complexity of these dynamic algorithms, finding that the Genetic Algorithm demonstrated lower temporal complexity compared to others. Moreover, Ibrahim et al. [29] developed the AD-PSO-Guided WOA algorithm to dynamically optimize Long Short-Term Memory hyper-parameters for wind speed ensemble forecasting. They used data from the Kaggle Global Energy Forecasting Competition 2012 to accomplish the accurate hourly power generation prediction of up to forty-eight hours ahead at seven wind farms. Their algorithm outperformed the others in accuracy, confirmed by statistical analyses such as Wilcoxon's rank-sum.

## 3 Methodology

The methodology employed a systematic approach in evaluating the predictive performance of diverse machine learning algorithms on the heart disease dataset. A comprehensive set of twelve algorithms was to be considered for evaluation in this regard, where the choice of algorithms was based on literature relevance and suitability for classification tasks: Random Forest, Decision Tree,Gradient Boosting, AdaBoost, Extra Trees,Logistic Regression. Support Vector Classifier, K-Nearest Neighbors, Gaussian Naive Bayes, Multi-Layer Perceptron,. Linear Discriminant Analysis. Quadratic Discriminant Analysis.To ensure the robustness of the evaluation, every algorithm was trained and tested following standard procedures, inclusive of techniques such as cross-validation. This helped in mitigating the over fitting problem and provided a good estimate of the performance of algorithms. It was among the tree-based methods that Random Forest demonstrated the best predictive accuracy. We used the Personal Key Indicators of Heart Disease Kaggle dataset, with 320,000 rows and 18 columns of data on different health indicators derived from 2020 CDC survey data. We explored Grid Search, Evolutionary Algorithms, and Swarm-Based Algorithms for HPO. The suggested Hybrid Swarm Evolution Optimization algorithm involves swarm intelligence and evolutionary computation, which started from a swarm of candidate solutions; it evaluated them with validation datasets and uses adaptive

**Table 2**: Advantages & disadvantages of recent Hyper-parameter Optimization Techniques

| Researcher | Method/Approach | Advantage | Disadvantage |
|---|---|---|---|
| Bergstra et al. [13], Maclaurin et al. [14] | Random Search | More efficient than grid search for Hyper-parameter optimization | Inefficiency due to random exploration of Hyper-parameter space |
| Snoek et al. [15] | Bayesian Optimization | Efficient for black-box optimization problems | Requires modelling the objective function as a probabilistic surrogate |
| Hazan et al. [16] | Gradient-based Optimization | Utilizes compressed sensing techniques for orthogonal polynomials | May require significant computational resources |
| Freitas et al. [17] | Evolutionary Algorithms (EAs) | Offer robust and adaptable global search techniques | May converge to suboptimal solutions |
| Jaderberg et al. [18] | Population Based Training (PBT) | Utilizes fixed computational budget to optimize models and Hyper-parameters concurrently | Requires careful tuning of parameters to avoid premature convergence |
| Ho et al. [19] | Population Based Augmentation (PBA) | Achieves performance comparable to AutoAugment with significantly lower computational requirements | Performance may vary depending on dataset characteristics |
| Real et al. [20] | AutoML-Zero | Automates the discovery of complete machine learning algorithms | Evolutionary search may require extensive computational resources |
| Zoph et al. [21] | Neural Architecture Search with Reinforcement Learning | Achieves impressive results comparable to human-invented architectures | Training a recurrent neural network (RNN) for architecture generation may be computationally intensive |
| Shahhosseini et al. [22] | Ensemble Methods Optimization | Combines ensembles through weighted averages and tunes Hyper-parameters of each base learner | Bayesian search may require significant computational resources |
| Folch et al. [23] | Multi-Fidelity Optimization | Balances the trade-off between accuracy and computational cost | Evaluations may require significant waiting times |
| Diaz et al. [24] | Derivative-free Optimization | Accelerates the search process by employing a radial basis function model of the NN's prediction accuracy | Limited training epochs for candidate configurations may result in suboptimal solutions |
| Liu et al. [25] | Differentiable Architecture Search (DARTS) | Efficiently searches architectures using gradient descent on a continuous relaxation of the architecture | Scalability challenge in handling large architecture search spaces |
| Li et al. [26] | Hyperband | Speeds up random search through adaptive resource allocation and early-stopping | Pure-exploration non-stochastic infinite-armed bandit problem may not suit all optimization scenarios |
| Luo et al. [27] | Adaptive Optimization Methods (AdaBound, AMSBound) | Addresses the issue of poor generalization caused by extreme learning rates in adaptive optimization methods | Dynamic bounds on learning rates may require careful tuning to prevent instability |
| Ali et al. [28] | Dynamic Optimization Algorithms | Explores various optimization strategies for dynamically fine-tuning Hyper-parameters | Computational complexity may increase significantly depending on the algorithm used, potentially limiting scalability |
| Ibrahim et al. [29] | AD-PSO-Guided WOA | Dynamically optimizes LSTM Hyper-parameters for ensemble forecasting | Performance may vary depending on the specific characteristics of the data, requiring careful parameter tuning |

mutation rates to enhance its performance. This approach allows for efficient exploration of the hyper-parameter space, leading to optimized model configurations. The pseudo code and detailed working of the model are provided in Algorithm 1 and Figure 1.

### 3.1 About the dataset

The Personal Key Indicators of Heart Disease dataset [30] is a rich dataset that contains detailed information on the health status of people. The dataset is based on the 2020 annual CDC (Centers for Disease Control and Prevention) survey data. It contains 320,000 rows and 18 columns. It is a refined, smaller version of the original survey, with data of 400,000 adults across the United States.Data for this dataset was collected from telephone surveys. Each row in the dataset represents a patient and contains their different indicators. Most of the questions asked during the survey were binary; that means a 'yes' or 'no' answer was required. All these 'yes' or 'no' answers were then collected into the dataset for analysis and research purposes. HeartDisease: Shows whether the respondents have ever reported coronary heart disease (CHD) or myocardial infarction (MI).
BMI: Body Mass Index of each individual; it is a floating-point value. Smoking: Indicates individuals who have ever smoked at least 100 cigarettes in their lifetime.
AlcoholDrinking: Identifies the heavy drinkers using the threshold of weekly alcohol consumption.
Stroke: Whether the individual was told they have had a stroke.
PhysicalHealth: How many days an individual experienced physical illness or injury in the last 30 days.
MentalHealth: How many days in the last 30 days the individual reported their mental health was not good.
DiffWalking: If the individual had serious difficulty walking or climbing stairs.
Sex: Whether the individual is male or female.
AgeCategory: Categorical age in fourteen age categories.
Race: Imputed race/ethnicity values
Diabetic: If the individual has ever had diabetes.
PhysicalActivity: Whether the individual did physical activity or exercise within the past 30 days.
GenHealth: The individual's general health as perceived by themselves.
SleepTime: The average number of hours of sleep in a 24-hour period.
Asthma: Demonstrates the presence of asthma among individuals
KidneyDisease: Denotes individuals diagnosed with kidney disease
SkinCancer: Denotes individuals who had been diagnosed with skin cancer.

To present the statistical details of the dataset, the following tables provide insights into the numerical and categorical attributes, as well as the results from statistical tests and confidence intervals.
Table 3 summarizes the results of various statistical tests performed on the dataset, including T-tests and Chi-square tests. These tests provide insights into the relationships and differences between different attributes in the dataset.Table 4 presents the

confidence intervals for various attributes, providing a range within which the true population parameters are expected to lie.Table 5 displays the descriptive statistics for the continuous numerical attributes in the dataset. This includes the mean, minimum, and maximum values for BMI, Age, Blood Pressure, and GenHealth Score. Table 6 provides the population percentages for various categorical attributes in the dataset. This table helps in understanding the distribution of different categories such as age groups, race, smoking status, sex, physical activity, and more.

**Table 3**: T-Tests

| Comparison | t-statistic | p-value |
|---|---|---|
| Age between Smokers and Non-Smokers | 73.7812 | 0.0000 |
| Age between Diabetic and Non-Diabetic | 113.5581 | 0.0000 |

**Table 4**: Confidence Intervals

| Statistic | Confidence Interval |
|---|---|
| Mean Age | (54.4446, 54.5690) |
| Mean BMI | (28.3034, 28.3474) |
| Proportion of Smokers | 0.4125 (0.4108, 0.4142) |
| Proportion of Physical Activity | 0.7754 (0.7739, 0.7768) |

**Table 5**: Numerical Attributes

| Attribute | Mean | Min | Max | Attribute Type |
|---|---|---|---|---|
| BMI | 28.325399 | 12.02 | 94.85 | Continuous |
| Age | 3.37171 | 0 | 30 | Continuous |
| Blood Pressure | 3.898366 | 0 | 30 | Continuous |
| GenHealth Score | 54.506812 | 21 | 82 | Continuous |

The statistical analyses conducted on the dataset reveal significant insights into the health status and behaviors of the population. The T-tests indicate a notable difference in age between smokers and non-smokers, as well as between diabetic and non-diabetic individuals, both with highly significant p-values. The Chi-square tests show a strong association between sex and physical activity, as well as between stroke and physical activity, again with highly significant p-values. The confidence intervals provide a reliable range for the mean age and BMI of the population, as well as for the proportions of smokers and those engaging in physical activity. These results underscore the importance of continuous monitoring and targeted interventions to address health disparities and promote healthier lifestyles within the population.

**Table 6**: Categorical Attributes with Percentages

| Attribute | Category | Percentage |
|---|---|---|
| AgeCategory | 65-69 | 10.68% |
| AgeCategory | 60-64 | 10.53% |
| AgeCategory | 70-74 | 9.71% |
| AgeCategory | 55-59 | 9.31% |
| AgeCategory | 50-54 | 7.94% |
| AgeCategory | 80 or older | 7.55% |
| AgeCategory | 45-49 | 6.81% |
| AgeCategory | 75-79 | 6.72% |
| AgeCategory | 18-24 | 6.59% |
| AgeCategory | 40-44 | 6.57% |
| AgeCategory | 35-39 | 6.43% |
| AgeCategory | 30-34 | 5.86% |
| AgeCategory | 25-29 | 5.30% |
| Race | White | 76.68% |
| Race | Hispanic | 8.58% |
| Race | Black | 7.17% |
| Race | Other | 3.42% |
| Race | Asian | 2.52% |
| Race | American Indian/Alaskan Native | 1.63% |
| Smoking | 0 (Non-smoker) | 58.75% |
| Smoking | 1 (Smoker) | 41.25% |
| Sex | 0 (Male) | 52.47% |
| Sex | 1 (Female) | 47.53% |
| PhysicalActivity | 1 (Yes) | 77.54% |
| PhysicalActivity | 0 (No) | 22.46% |
| AlcoholDrinking | 0 (No) | 93.19% |
| AlcoholDrinking | 1 (Yes) | 6.81% |
| Stroke | 0 (No) | 96.23% |
| Stroke | 1 (Yes) | 3.77% |
| DiffWalking | 0 (No) | 86.11% |
| DiffWalking | 1 (Yes) | 13.89% |
| Asthma | 0 (No) | 86.59% |
| Asthma | 1 (Yes) | 13.41% |
| KidneyDisease | 0 (No) | 96.32% |
| KidneyDisease | 1 (Yes) | 3.68% |
| General Health | 4 (Excellent) | 35.60% |
| General Health | 3 (Good) | 29.12% |
| General Health | 5 (Poor) | 20.90% |
| General Health | 2 (Fair) | 10.84% |
| General Health | 1 (Very Poor) | 3.53% |
| Diabetic | 0 (Non-diabetic) | 86.44% |
| Diabetic | 1 (Diabetic) | 13.56% |

## 3.2 Hyper Parameter Optimization

The Hyper-parameter Optimization (HPO) [29] is the process of finding the most optimal values for those parameters in a model that are external to the model, known as hyper-parameters. The process of HPO aims at improving machine learning models by fine-tuning these hyper-parameters. Three common strategies for hyper-parameter optimization include:

### 3.2.1 Grid Search

Grid search [13],[28] involves an exhaustive search over a set of hyper-parameter values defined by a grid. Grid search involves training the model with all combinations of hyper-parameters from the defined grid. This way, the combination yielding the best performance is chosen. Grid search is very simple and easy to implement; however, it tends to be computationally expensive, especially when dealing with a large number of hyper-parameters or wide ranges of values.

### 3.2.2 Evolutionary Algorithms

Evolutionary algorithms [20],[17] are algorithms that imitate the processes of natural selection. An evolutionary algorithm maintains a population of candidate solutions represented as individuals. It evolves the pool of candidate solutions by applying selection, crossover, and mutation operations, with the aim of generating better solutions. Evolutionary algorithms are effective at searching a large space and will be proper for nonlinear and multi-modal optimization problems. However, they may be very slow to arrive at a certain optimal solution and may require a large number of iterations.

### 3.2.3 Swarm-Based Algorithms

Swarm-based algorithms [27] , such as particle swarm optimization, are a family of techniques that mimic the behavior of swarms or groups of individuals that coordinate their activities towards optimal solutions. In Particle Swarm Optimisation, each individual, termed as a particle, changes its position within the search space based on its own experience and the experience of its neighboring particles. Swarm algorithms are great for exploring complex search spaces and often can converge to good solutions fairly quickly. However, they may become stuck in local optima and may require careful tuning of their parameters to achieve optimal performance.

## 3.3 Proposed Hybrid Swarm Evolution Optimisation

The Proposed Hybrid Swarm Evolution Optimization (HySEOpt) algorithm is a sophisticated strategy in hyper-parameter optimization (HPO), trying to combine the power of swarm-based exploration and evolutionary mechanisms while trying to mitigate drawbacks of existing approaches.The detailed working of the model can be seen in figure 1.

Core to HySEOpt is the integration of principles from swarm intelligence and evolutionary computation toward a novel solution for the complex problem of hyper-parameter tuning.First, this method initializes a swarm of candidate solutions representing a set of hyper-parameters for the Random Forest classifier. The hyper-parameters include aspects like the number of estimators, maximum depth, minimum samples split, and many others. The candidates are generated randomly, but in defined ranges, to cover a variety of possible configurations.

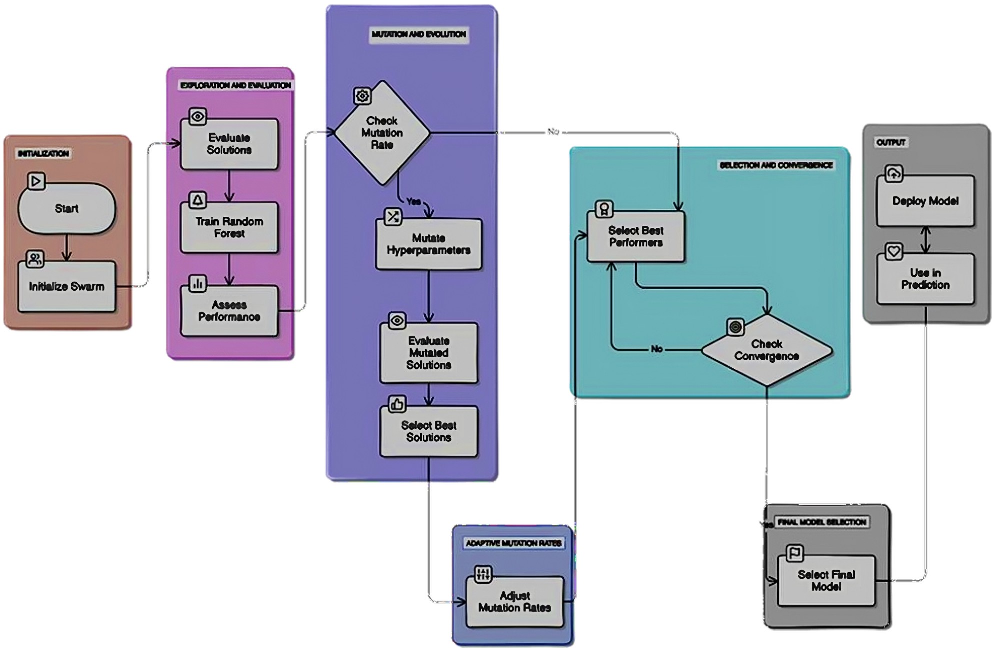

**Fig. 1**: Hybrid Swarm Evolution Optimisation Working

After initialization, the swarm moves on to an evaluation phase in which each solution is put through a detailed evaluation using a specifically set validation dataset. This evaluation involves training a Random Forest model with the given hyper-parameters and rating its performance according to well-known metrics, including the AUC-ROC score. Notice that the swarm does this in parallel, allowing for exploration of the hyper-parameter space and the effective discovery of promising configurations.

One of the outstanding features of HySEOpt is its adaptive mutation mechanism, which plays a fundamental role in exploration. Each solution's hyper-parameters are further subject to mutation, which is implemented based on dynamically adjusted mutation rates after the evaluation. In so doing, this mutation process maintains diversity in the swarm, keeping it equipped to explore further, unexplored regions of the hyper-parameter space. Importantly, mutation rates change in real-time as a function of swarm performance, with higher rates allocated for unexplored regions and lower rates allocated for promising ones.

Throughout the process of optimization, the algorithm carefully picks the best-performing solutions in the swarm and uses their insights to guide the steps in iterations. The solutions thus act as a beacon, guiding mutation and exploration, driving convergence in optimization. Convergence is the milestone within HySEOpt, reached at the moment the algorithm meets some predefined criterion, such as a number of maximum iterations, or no more improvement noticed in the performance.

Eventually, after the convergence of HySEOpt, the result of the algorithm is the optimal solution selection; in other words, the best model represented by the random forest model with the highest score on the validation set. Here is the step by step process

Initialization: Start with a swarm of randomly generated candidate hyper-parameters for the Random Forest classifier.

Evaluation: Train and evaluate each candidate using a validation dataset. Assess performance with metrics like AUC-ROC.

Adaptive Mutation: Apply mutations to hyper-parameters with dynamically adjusted rates. Higher mutation rates for less explored areas, lower for promising areas.

Iteration and Convergence: Iterate through multiple cycles, updating the best-performing solutions.Stop when performance stabilizes or after a set number of iterations.

Final Selection: Identify and select the best hyper-parameters based on highest performance scores.

The Pseudo code of the Proposed HySEOpt is given in Algorithm 1.

**Algorithm 1** Hybrid Swarm Evolution Optimization (HySEOpt) Algorithm

```
Require: $max\_iter$,$swarm\_size$, $mutation\_rate$, $initial\_mutation\_rate$,
        $mutation\_rate\_decay$
Ensure: $best\_parameters$
Initialize the HySEOpt algorithm with given parameters
for $iteration \leftarrow 1$ to $max\_iter$ do
    Initialize a swarm of candidate solutions
    for each solution in the swarm do
        Evaluate the solution using training and validation datasets
        Generate mutated solutions based on current solutions
        Evaluate mutated solutions
    end for
    Update mutation rate based on evaluation results
    Select the best-performing solution from the swarm
    if best score improves then
        Update best parameters
    end if
    Append best score to list of validation scores
    Print current iteration number and best validation score
end for
Print final model AUC score representing the optimized Random Forest classifier
    with the best set of hyper-parameters
```

The initialization of the algorithm is $O(1)$. The main loop, which executes for $max\_iter$ iterations, has a complexity of $O(max\_iter)$. Within each iteration, initializing the swarm of candidate solutions takes $O(swarm\_size)$. Evaluating each solution contributes $O(swarm\_size \times E)$, where $E$ represents the time complexity of evaluating a single solution. Generating and evaluating mutated solutions adds

$O(swarm\_size \times M \times E)$, with $M$ being the number of mutations per solution. Updating the mutation rate, selecting the best solution, and updating scores are $O(1)$ operations per iteration.

Thus, the overall time complexity of the HySEOpt algorithm is $O(max\_iter \times (swarm\_size \times (E + M \times E)))$.

With mutation rates and parallel exploration capabilities, HySEOpt makes a new approach to deal with the challenging search space in an effective manner. By smoothly combining these with tree-based classifiers, HySEOpt attempted to maximize predictive performance and generalization ability in the models for insightful and accurate prediction of heart disease. Equipped with such a comprehensive methodology, our attention next turns to the results of our experiment in Section 4.

# 4 Results and Discussion

Figures 2 to 5 show that the results of the experiment on the ML algorithms were really a showcase of the great performance of tree-based methods on most of the evaluation metrics. Tree-based classifiers are shown to perform consistently well in classification problems, which include Decision Tree, Random Forest, Gradient Boosting, AdaBoost, Extra Trees, and their variants.

Firstly, tree-based methods show high accuracy, as it is seen that Random Forest stands out to be the best among all, with an accuracy score of 93%. It means that these classifiers are effective in correctly predicting the target variable. Secondly, tree-based classifiers outperform in the class distinction and are indicated by their high area under the curve score. The AUC score is a measure which encapsulates the overall ability of the classifier to distinguish between the positive and negative instances. Therefore, the higher score obtained by tree-based methods further substantiates their efficacy in the classification task.

Thirdly, tree-based classifiers generalize robustly, shown by their high cross-validation score. In this case, Random Forest stands out as an excellent example with a score of 97% in cross-validation, therefore underpinning its ability to perform well across different data splits. This is a clear indication that tree-based methods are reliable and consistent across various datasets.Finally, the tree-based methods are balanced, which can be seen by their impressive F1 score that summarizes both precision and recall. With F1 scores of most tree-based methods consistently above 0.9, such classifiers demonstrate being effective in achieving both high precision and recall, which are important in correct classification in case of imbalanced datasets.

HySEOpt's effectiveness in the process of optimizing hyper-parameters. Starting from the initialization of HySEOpt, the algorithm iteratively mutated the hyper-parameters, evaluated the model performance based on validation datasets, and performed an adaptation of the mutation rate based on the results. Such a dynamic and systematic approach enables steady improvement in the validation scores, starting from the initial 97.23% and rising to an impressive 99.06% in the 10th iteration as seem in figure 6. The final model achieves a high AUC of 0.9906, which underscores its excellent ability to distinguish positive and negative instances with high accuracy. Such an iterative improvement in the validation score visualizes the capability of the

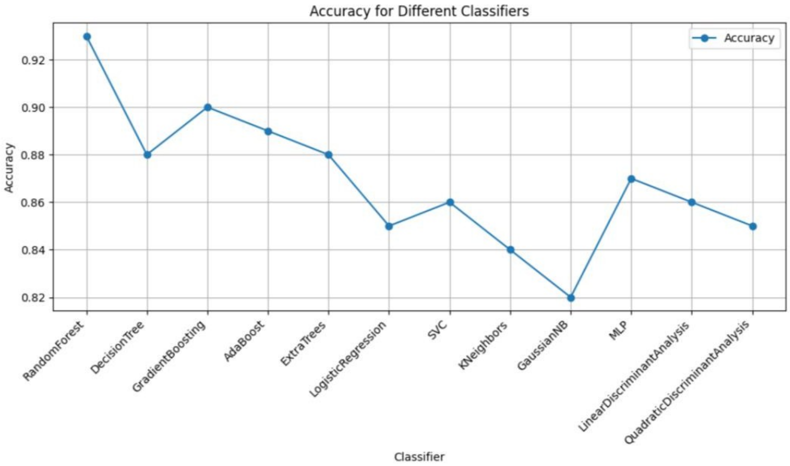

**Fig. 2**: Accuracy of Classifiers

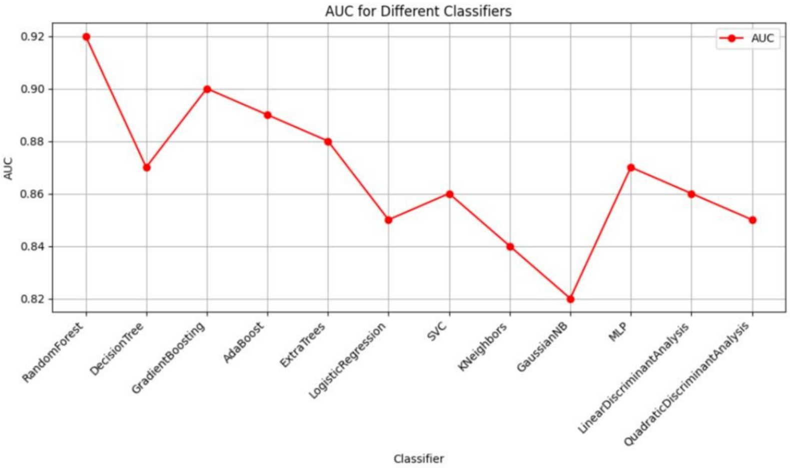

**Fig. 3**: AUC for Classifiers

algorithm to fine-tune the hyper-parameters, enabling a huge rise in enhancements of model accuracy and generalization. This experiment shows the proficiency of HySEOpt in optimizing hyper-parameters for the best possible performance of machine learning models, thus improving their accuracy and robustness. Furthermore Sensitivity and Specificity were also measured for HySEOpt and are 98.12% and 99.09% respectively Figure 7 is a bar chart that provides a comprehensive comparison of the performance metrics: Accuracy, Cross-Validation AUC, ROC AUC, and F1 Score between Grid Search and HySEOpt. HySEOpt performs better in all the metrics. For the measure of accuracy, HySEOpt scores 98.01% compared to Grid Search, which scores 95% in accuracy. For the Cross-Validation AUC, HySEOpt scores 99% as opposed to Grid Search, which scores 94%. This indicator reflects better distinction between positive and negative samples. Also, for HySEOpt, the ROC AUC score of 99% surpasses that

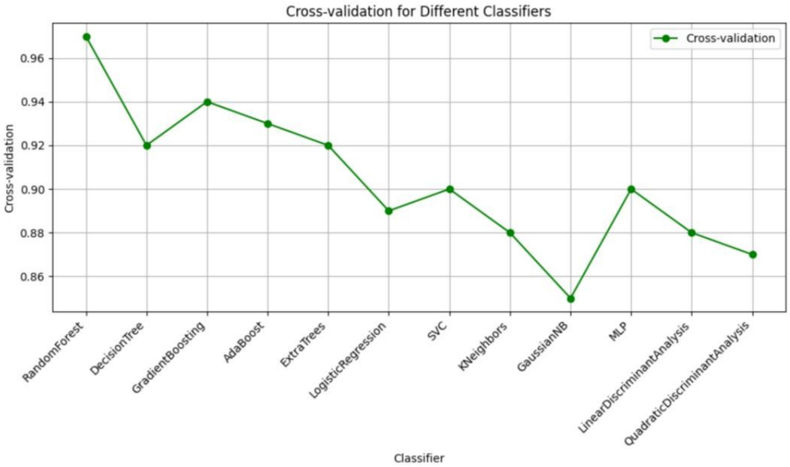

**Fig. 4**: Cross-Validation Score of Classifiers

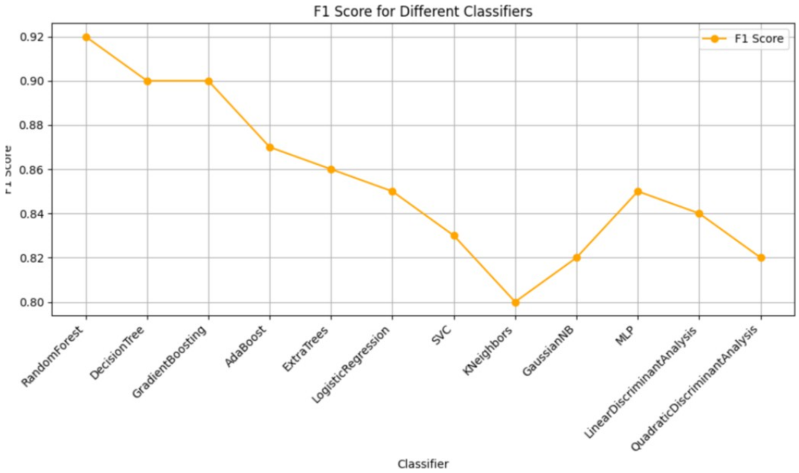

**Fig. 5**: F1 Score of Classifiers

of Grid Search, which is 93% . Additionally, the F1 score of 0.95 for HySEOpt is higher than that of Grid Search, 0.92,which shows precision and recall. The results reflect the efficacy of HySEOpt in hyper-parameter tuning for significantly elevating the performance of the model compared to Grid Search.

The time efficiency of HySEOpt compared to Grid Search and Evolutionary Algorithm and Swarm Algorithm in Figure 8. Grid Search takes an average of 813.6 seconds per iteration because of its exhaustive search method in high-dimensional space. HySEOpt finishes 5 iterations in 1875 seconds, with an average of 375.0 seconds per iteration, showing much faster convergence with an adaptive mutation rate, parallel processing, and dynamic exploration strategies. Evolutionary Algorithms, with total training time of 2160 seconds and an average of 432.0 seconds per iteration, are slower because their population-based approach may have sensitivity to parameter tuning.

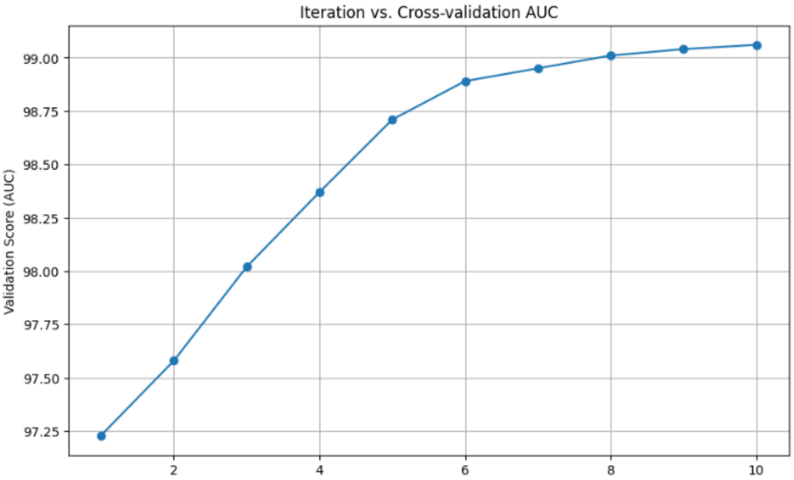

**Fig. 6**: Iteration vs. Cross-Validation for HySEOpt Algorithm

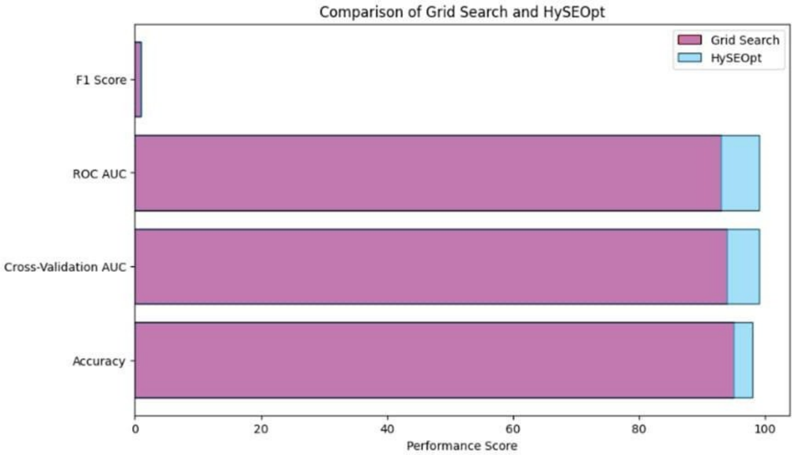

**Fig. 7**: Comparison of Performance Metrics: Grid Search vs. HySEOpt on Random Forest

Similarly, the Regular Swarm Algorithm takes 2580 seconds for 5 iterations, averaging 516.0 seconds per iteration, facing challenges such as slow convergence and sensitivity in settings. HySEOpt seems more efficient than both approaches in terms of time. In general, these results prove the higher efficiency and effectiveness of HySEOpt in the task of hyper-parameter optimization, especially useful when there is little time and limited computational resources.

These results as seen in Table 7 underline that HySEOpt is an efficient algorithm for the optimization of hyper-parameters in heart disease prediction. With the application of swarm intelligence and adaptive mutation strategies, HySEOpt is shown

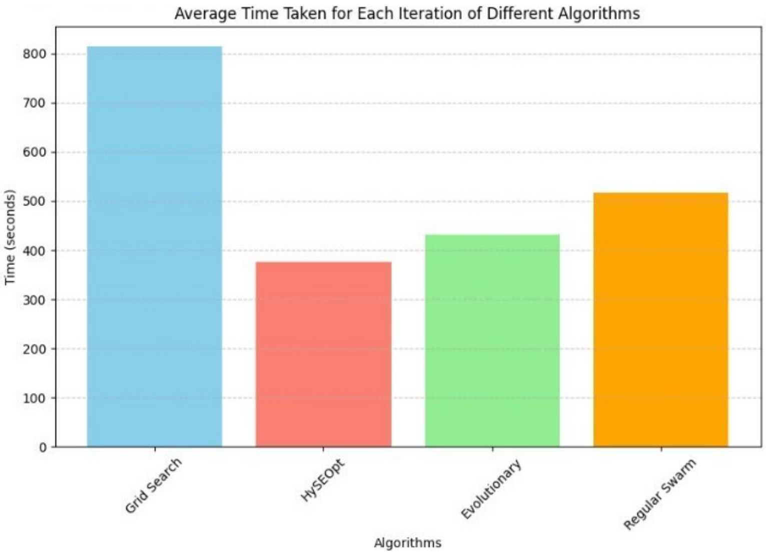

**Fig. 8**: Time Comparison of HPO algorithms

to outperform traditional methods, including Grid Search. The robust optimization provided by HySEOpt enhances the predictive accuracy and generalization ability of the models—essential for accurate diagnosis and treatment planning in heart disease. Moreover, the success shown in the experiment with tree-based methods underscores their efficiency in handling high-dimensional datasets and capturing the inherent complexity of heart disease datasets. Integration of HySEOpt with tree-based classifiers like Random Forest, Decision Tree, and Gradient Boosting will go a long way in improving heart disease prediction models and, in turn, patient care and healthcare decisions. Table 8 presents a comparison of accuracy metrics for various algorithms. HySEOpt demonstrates the highest accuracy at 98.01%, surpassing other methods evaluated in recent studies

**Table 7**: HySEOpt Performance Metrics

| Metric | Value |
|---|---|
| Accuracy | 98.01% |
| Cross-Validation AUC | 99.00% |
| ROC AUC | 99.00% |
| F1 Score | 0.95 |
| Sensitivity | 98.12% |
| Specificity | 99.09% |
| Final Model AUC | 0.9906 |
| Average Time per Iteration | 375.0 s |
| Total Time for 5 Iterations | 1875 s |

**Table 8**: Comparison with latest research

| Name | Algorithm | Accuracy |
|---|---|---|
| HySEOpt | HySEOpt | 98.01% |
| Nayeem et al. (2022) [8] | Random Forest | 95.63% |
| Bhatt et al. (2023) [5] | XGBoost | 87.02% |
| Hossain et al. (2023) [2] | Random Forest | 90.00% |

Our findings suggest that using HySEOpt with tree-based classifiers, such as RandomForest, substantially enhances the accuracy of heart disease predictions. Future efforts should focus on improving feature engineering, incorporating genetic and imaging data, and leveraging ensemble models. It's also essential to validate models across various settings and optimize techniques to ensure practical effectiveness.

# 5 Conclusion

Our experiments show that HySEOpt combined with tree-based classifiers, such as RandomForest, can drastically increase the correctness of heart disease prediction models and got as accuracy of 98.01%. This leads to enhanced identification of people at risk of heart disease and early intervention, which will directly impact the improvement of patient outcomes. Many other avenues are still open for further exploration and refinement. For instance, newer data including genetic data and imaging could bring about a better understanding of risk factors and protective factors for heart disease. Feature engineering, including feature selection and dimensionality reduction, could optimize model performance through the identification of the most informative features. Further, ensemble models by combining the predictions of various base models could further enhance predictive accuracy. Best optimization techniques, hyper-parameter fine-tuning, and validation of predictive models in external datasets of diverse populations are key to determining model generalizability and scalability. Extending the model for clinical validation studies by healthcare professionals and deployment in a healthcare system can facilitate real-world implementation. Continuous monitoring of the model with updates to it is paramount to ensuring accuracy and effectiveness. Further work in these directions may improve the field of heart disease prediction and, therefore, improve healthcare outcomes for people worldwide.

# 6 Statements and Declarations

Funding - Not applicable.

Conflict of interest - The authors declare no conflict of interest.
Ethics approval and consent to participate - Not Applicable
Consent of Publication - Not applicable.
Data Availability -The datasets used in this research is openly accessible for academic use on Kaggle [30]
Materials Availability - Not Applicable.
Code Availability - Not Applicable.
Author Contribution – V.P. conceived the research idea, designed experiments,

implemented the various hyper-parameter optimization algorithms, performed data analysis, interpreted results, wrote the manuscript, and created visualizations.
B. S. contributed to investigation, provided resources, participated in writing, reviewing, and editing the manuscript, and contributed to data analysis and interpretation of results.
A.B. curated and prepared datasets for training and evaluation, and participated in writing, reviewing, and editing the manuscript.
A.M. supervised the project, provided assistance in research, reviewed and edited the manuscript, and administered the project.

## References

[1] D. Salhi, T.A. E. Dhai, Kechadi, T.: Using machine learning for heart disease prediction (2020) https://doi.org/10.1007/978-3-030-69418-0_7

[2] M. I. Hossain, M.A.R.K.e.a. M. H. Maruf: Heart disease prediction using distinct artificial intelligence techniques: performance analysis and comparison. Iranian Journal of Computer Science **6**, 397–417 (2023) https://doi.org/10.1007/s42044-023-00148-7

[3] Y. Muhammad, M.H.e.a. M. Tahir: Early and accurate detection and diagnosis of heart disease using intelligent computational model. Scientific Reports **10**, 19747 (2020) https://doi.org/10.1038/s41598-020-76635-9

[4] V. Sharma, S.Y., Gupta, M.: Heart disease prediction using machine learning techniques. In: 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 177–181 (2020). https://doi.org/10.1109/ICACCCN51052.2020.9362842

[5] C. M. Bhatt, T.G. P. Patel, Mazzeo, P.L.: Effective heart disease prediction using machine learning techniques. Algorithms **16**(2), 88 (2023) https://doi.org/10.3390/a16020088

[6] Chandrasekhar, N., Peddakrishna, S.: Enhancing heart disease prediction accuracy through machine learning techniques and optimization. Processes **11**(4), 1210 (2023) https://doi.org/10.3390/pr11041210

[7] S. Patidar, A.J., Gupta, A.: Comparative analysis of machine learning algorithms for heart disease predictions. In: 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1340–1344 (2022). https://doi.org/10.1109/ICICCS53718.2022.9788408

[8] M. J. N. Nayeem, S.R., Islam, M.R.: Prediction of heart disease using machine learning algorithms. European Journal of Artificial Intelligence and Machine Learning **1**(3), 22–26 (2022) https://doi.org/10.24018/ejai.2022.1.3.13

[9] A. Marelli, A.L.e.a. C. Li: Machine learning informed diagnosis for congenital heart disease in large claims data source. JACC Advances **3**(2) (2024) https://doi.org/10.1016/j.jacadv.2023.100801

[10] N. Nissa, S.J., Mohammad, S.: Heart disease prediction using machine learning techniques. Wesleyan Journal of Research **13**(67) (2021)

[11] Valarmathi, R., Sheela, T.: Heart disease prediction using hyper parameter optimization (hpo) tuning. Biomedical Signal Processing and Control **70** (2021) https://doi.org/10.1016/j.bspc.2021.103033

[12] D. Asif, M.S.A. M. Bibi, Mukheimer, A.: Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization. Algorithms **16**(6), 308 (2023) https://doi.org/10.3390/a16060308

[13] Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research **13**(10), 281–305 (2012)

[14] D. Maclaurin, D.D., Adams, R.P.: Gradient-based hyperparameter optimization through reversible learning. arXiv preprint (2015) 1502.03492

[15] K. Swersky, J.S., Adams, R.P.: Multi-task bayesian optimization. In: Advances in Neural Information Processing Systems, vol. 26. https://proceedings.neurips.cc/paper/2013/file/f33ba15effa5c10e873bf3842afb46a6-Paper.pdf

[16] E. Hazan, A.K., Yuan, Y.: Hyperparameter optimization: A spectral approach. arXiv preprint (2017) 1706.00764

[17] Freitas, A.: A review of evolutionary algorithms for data mining, (2010). https://doi.org/10.1007/978-0-387-09823-4_19

[18] M. Jaderberg, S.O.W.M.C.J.D.A.R.e.a. V. Dalibard: Population based training of neural networks. arXiv preprint (2017) 1711.09846

[19] D. Ho, X.C.I.S. E. Liang, Abbeel, P.: Population based augmentation: Efficient learning of augmentation policy schedules. In: International Conference on Machine Learning, pp. 2731–2741 (2019). https://doi.org/10.48550/arXiv.2003.03384

[20] E. Real, D.S. C. Liang, Le, Q.: Automl-zero: Evolving machine learning algorithms from scratch. International Conference on Machine Learning, 8007–8019 (2020) https://doi.org/10.48550/arXiv.1611.01578

[21] Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint (2016) 1611.01578

[22] M. Shahhosseini, G.H., Pham, H.: Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. Machine Learning with

Applications **7** (2022) https://doi.org/10.1016/j.mlwa.2022.100251

[23] J. P. Folch, B.S.D.W.C.T.M.v.d.W. R. M. Lee, Misener, R.: Combining multi-fidelity modelling and asynchronous batch bayesian optimization. Computers & Chemical Engineering **172**, 108194 (2023) https://doi.org/10.1016/j.compchemeng.2023.108194

[24] G. I. Diaz, G.N. A. Fokoue-Nkoutche, Samulowitz, H.: An effective algorithm for hyperparameter optimization of neural networks. IBM Journal of Research and Development **61**(4/5), 9–1 (2017) https://doi.org/10.48550/arXiv.1705.08520

[25] H. Liu, K.S., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint (2018) 1806.09055

[26] L. Li, G.D.A.R. K. Jamieson, Talwalkar, A.: Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research **18**(185), 1–52 (2018)

[27] Luo, L., Xiong, Y., Liu, Y., Sun, X.: Adaptive gradient methods with dynamic bound of learning rate. arXiv preprint (2019) 1902.09843

[28] Ali, Y., Awwad, E., Al-Razgan, M., Maarouf, A.: Hyperparameter search for machine learning algorithms for optimizing the computational complexity. Processes **11**, 349 (2023) https://doi.org/10.3390/pr11020349

[29] Ibrahim, A., *et al.*: Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm. IEEE Access **9**, 125787–125804 (2021) https://doi.org/10.1109/ACCESS.2021.3111408

[30] Indicators of Heart Disease (2022 UPDATE). https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease (2022)