Journal of Theoretical and Applied Information Technology
15th May 2024. Vol.102. No 9

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OPTIMIZING HEART DISEASE PREDICTION MODELS USING GENETIC ALGORITHMS: A METAHEURISTIC APPROACH

SUSHILA PALIWAL[1], SURAIYA PRAVEEN[2], M. AFSHAR ALAM[3], JAWED AHMED[4]

[1]Research Scholar, Department of Computer Science & Engineering, SEST, Jamia Hamdard, India
[2]Associate Professor, Department of Computer Science & Engineering, SEST, Jamia Hamdard, India
[3]Professor, Department of Computer Science & Engineering, SEST, Jamia Hamdard, India
[4]Associate Professor, Department of Computer Science & Engineering, SEST, Jamia Hamdard, India

E-mail: [1]sushila_paliwal@yahoo.com, [2]husainsuraiya@gmail.com, [3]aalam@jamiahamdard.ac.in, [4]jawed2047@gmail.com

ABSTRACT

Cardiovascular diseases, including heart disease, remain a significant global health concern. Early detection and accurate prediction of heart disease risk factors are crucial for effective prevention and intervention. This research paper uses a metaheuristic approach with genetic algorithms (GAs) to optimize hyperparameter settings to improve predictive accuracy of heart disease models. Using genetic algorithms as a metaheuristic approach, this research article aims to improve the accuracy and generalizability of heart disease prediction models. The primary objective is to converge towards optimal model parameters, ultimately maximizing accuracy or minimizing error. The study investigates the effectiveness of combining genetic algorithms with machine learning in improving heart disease prediction, with a focus on enhancing accuracy, generalizability, and scalability. Through comparison with traditional methods, the research assesses the superiority of the proposed approach and its potential contributions to heart disease diagnosis and treatment. Notably, the research stands out for its use of genetic algorithms in conjunction with cross-validation to optimize hyperparameters, identify optimal model parameters, and evaluate performance by minimizing errors. The application of the Support Vector Machine (SVM) classifier with optimized hyperparameters yielded a significant 97% improvement in accuracy with the heart disease dataset, surpassing results from previous studies. This research thus highlights the promise of genetic algorithms in enhancing heart disease prediction models and advancing healthcare analytics

Keywords: *Metaheuristic Approach, Genetic Algorithm, Hyperparameter Optimization, Support Vector Machine, Heart Disease Prediction*

1. INTRODUCTION

Heart disease is an important global health concern, being the leading cause of mortality worldwide, which represents a substantial burden on individuals, families, and healthcare systems [1]. Cardiovascular disease accounts for 40% of all deaths in China and is the leading cause of death worldwide, according to data provided by the country's national mortality surveillance system [2]. It is crucial to prioritize research and interventions to effectively address this public health problem.

The escalating cardiovascular disease epidemic is a global issue of great concern. If allowed to escalate without intervention, the resulting morbidity, mortality, and economic repercussions could have far-reaching effects worldwide. Efforts to address this epidemic must involve a multi-faceted approach, including promoting healthy lifestyle habits, increasing awareness about cardiovascular disease risk factors, and improving access to quality healthcare services. Governments, healthcare providers, and individuals must work together to implement prevention and management strategies to combat this growing crisis. With coordinated action and an emphasis on early detection and treatment, we can strive to reduce the burden of cardiovascular disease and ultimately save lives.

Early detection and accurate prediction of heart disease risk factors are crucial because they allow timely intervention and prevention strategies. Identifying individuals who are at increased risk of developing heart disease allows for targeted

interventions, such as lifestyle modifications and medication, which can significantly reduce the likelihood of developing the disease. Additionally, accurate prediction of heart disease risk factors can help healthcare providers allocate resources more efficiently and effectively, leading to better patient outcomes.

This research seeks to assess the efficacy of combining genetic algorithms with machine learning techniques in predicting heart disease, evaluating enhancements in predictive accuracy, generalizability, and scalability. It plays a crucial role in advancing healthcare analytics by investigating novel approaches to improving the precision and efficiency of heart disease prediction models. By comparing results between genetic algorithms combined with machine learning algorithms and traditional optimization methods, researchers can ascertain the superiority of these advanced techniques. This exploration will inform how such advanced methods may aid in the diagnosis and treatment of heart disease.

The remainder of this paper is organized as follows. Section 2 provides an overview of the relevant research conducted on the topic. Section 3 of this research focuses on the methodology used, which includes collecting data, preparing the data, and doing a thorough analysis of hyperparameter optimization using genetic algorithms and SVM classification algorithms. Moreover, it provides detailed information about the methodology of decoding a binary chromosome, the procedure of selecting parents, the use of crossover to produce future generations and random mutations of individuals. In Section 4, the results of all experiments are analyzed and discussed in detail, including the various metrics used in machine learning models. In Section 5, the conclusion and future direction of the research work have been explored in detail.

2. RELATED WORK

AI algorithms can effectively identify patterns and indicators of heart disease at an early stage, enabling timely intervention and treatment [3]. The effectiveness of AI in detecting cardiovascular-related diseases from wearable devices has been systematically reviewed, with meta-analyzed sensitivity and specificity reaching high levels, indicating the potential for accurate detection and diagnosis [4]. The potential of AI in the early detection of heart disease has also been explored through machine learning approaches, with studies focusing on the development of AI models for the accurate and reliable detection of heart disease.

Current heart disease prediction models often face challenges in achieving high predictive accuracy and generalizability. This is due to the complexity and variability of heart disease risk factors, as well as the limitations of traditional optimization methods. However, the use of genetic algorithms as a metaheuristic approach to hyperparameter tuning can address these challenges. Genetic algorithms can search through a large space of hyperparameters and find the optimal combination that maximizes the model's performance. This can help overcome the limitations of traditional methods and improve the accuracy and generalization ability of heart disease prediction models.

Genetic algorithms are used by researchers in the classification and accurate diagnosis of diseases in medical fields, such as acute coronary syndrome, breast cancer, and diabetes [5]. Authors in [6] combined a genetic algorithm and neural network to create a system for complex categorization problems, with the aim of a classification accuracy of 94.17. The final weights of the system are used to predict the risk of heart disease. Furthermore, the study by [7] supports the idea of using genetic algorithms and the Adaptive Neuro-Fuzzy Inference System (GA-ANFIS) to create a simplified structure of the ANFIS model for detecting heart disease, reducing the number of costly tests and datasets. This hybrid technique shows high accuracy in predicting heart disease in both primary and secondary datasets, making it suitable for other datasets on heart disease and healthcare issues.

Genetic algorithms can improve the accuracy of breast cancer diagnosis using a wrapper approach for feature selection [8]. Researchers [9] introduced the potential applications of the genetic algorithm in various medical specialties, including radiology, oncology, pediatrics, cardiology, endocrinology, surgery, obstetrics, pulmonology, orthopedics, neurology, pharmacotherapy, and healthcare management. In [10], the authors have conducted an investigation and presented an approach that integrates genetic algorithms and decision trees to optimize hyperparameters for C-SVMs. The study focuses on searching for optimal values for the regularization parameter, the cost of classes, and the parameters of the RBF kernel function for SVM. [11] offers an exhaustive review of cutting-edge research in evolutionary computation pertaining to feature selection. A common strategy in genetic algorithms (GA), genetic programming (GP), and

particle swarm optimization (PSO) involves enhancing the representation to concurrently select features and optimize classifiers, such as support vector machines (SVM).

Despite AI algorithms' tremendous promise for detecting patterns and symptoms of heart disease via wearable devices, current heart disease prediction models have problems obtaining high predictive accuracy and generalizability. These issues are due to the complexity and variability of heart disease risk variables, as well as limitations in existing optimization methods.

While genetic algorithms have been identified as a viable metaheuristic strategy for hyperparameter tuning, notably in medical diagnostics, there is still a gap in the use of genetic algorithms specifically designed for optimizing heart disease prediction models. Thus, the task at hand is to design and assess an optimal heart disease prediction model employing genetic algorithms to improve predictive accuracy and generalizability, thereby facilitating early detection and intervention techniques. Research Question: How does integrating genetic algorithms with machine learning algorithms enhance the performance of heart disease prediction models compared to traditional optimization methods?

3. METHOD

In this research we applied the Support Vector Machines classifier and employed a Metaheuristics approach, specifically utilizing Genetic Algorithms for Hyperparameter optimization. Metaheuristics are optimization algorithms that are used to solve complex problems that cannot be easily solved using traditional methods. In the context of heart disease prediction, metaheuristics can be used to optimize the performance of predictive models by searching for the best combination of hyperparameters. The SVM algorithm involves numerous parameters, with Kernel, Degree, C, and Gamma being the four most commonly utilized hyperparameters. In our research, we specifically focus on the C and Gamma hyperparameters. C functions as a regularization parameter, influencing the accuracy of the hyperplanes in data separation. It manages the balance between minimizing the error in the training data and maximizing the weight norm. However, Gamma is a parameter associated with the Gaussian kernel, determining the reach of influence for a single training observation. Lower values indicate a broader influence, signifying distance, while higher values indicate a more localized influence, signifying proximity. We have implemented genetic algorithms using the Python programming language, and we have developed the algorithm without relying developed the algorithm without relying on any external libraries.

3.1 Data Collection

The Cardiovascular Disease Dataset [12] is used for this research from the Mendeley database. This dataset was obtained from a multispecialty hospital in India. With more than 14 common features as shown in Table 1, it is one of the most comprehensive heart disease dataset available for research purposes to date. Comprising 1000 subjects and 14 characteristics, this dataset is valuable for developing early-stage heart disease detection systems and generating predictive machine learning models.

3.2 Preprocessing of data

Data cleaning is a crucial step in the data preprocessing pipeline, ensuring the quality and reliability of the dataset used for analysis or machine learning models. One key aspect is handling missing values, where the goal is to identify and address gaps in the data by either filling them in with appropriate values or removing the instances with missing information. Additionally, removing duplicates is essential to prevent biases in models caused by identical rows, maintaining the diversity and accuracy of the dataset. Another essential data preprocessing step involves identifying and eliminating outliers. Outliers are data points that deviate significantly from the expected pattern, and their presence can adversely affect the performance of your model [13].

The next critical step involves examining whether the data are appropriately scaled, particularly in the context of variables related to heart disease, such as age and cholesterol levels. For example, age might be represented by two digits, while cholesterol levels could span two or three digits. Currently, cholesterol's larger numerical values might disproportionately influence the model, potentially overshadowing the true impact of age on predicting heart disease outcomes. However, in reality, age might have a more substantial impact than cholesterol levels. To address this, we standardize all variables on a consistent scale, typically between zero and one. This normalization ensures that each variable contributes equally to the heart disease prediction model, avoiding skewed influence based on their original numeric ranges [14].

Table 1: DATASET FEATURE INFORMATION

S.No.	Attribute	Description	Unit	Type
1	patientid	Patient Identification Number	Number	Numeric
2	age	Age	In Years	Numeric
3	gender	Gender	0, 1 (0-Female,1-Male)	Binary
4	chestpain	Chest pain type	0,1,2,3 (0 – typical angina, 1 – atypical angina, 2 – non-anginal pain, 3 – asymptomatic)	Nominal
5	restingBP	Resting blood pressure	94-200 (in mm HG)	Numeric
6	serumcholestrol	Serum cholesterol	126-564 (in mg/dl)	Numeric
7	fastingbloodsugar	Fasting blood sugar	0,1 > 120 mg/dl (0-False,1-true)	Binary
8	restingrelectro	Resting electrocardiogram results	0,1,2 (0 – Normal, 1 – having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of >0.05 mV) 2 – showing probable or definite left ventricular hypertrophy by Estes' criteria)	Nominal
9	maxheartrate	Maximum heart rate achieved	71-202	Numeric
10	exerciseangia	Exercise induced angina	0,1(0-no,1-yes)	Binary
11	oldpeak	Oldpeak =ST	0-6.2	Numeric
12	slope	Slope of the peak exercise ST segment	1,2,3 (1-unsloping, 2-flat,3-downsloping)	Nominal
13	noofmajorvessels	Number of major vessels	0,1,2,3	Numeric
14	target	Classification	0,1(0 – Absence of Heart Disease, 1 – Presence of Heart Disease)	Binary
				Binary

Here is a brief explanation of how the scaling works:

For each feature, the minimum and maximum values are computed during the fitting process. Then, each data point in the feature is transformed according to the formula:

$$XScaled = \frac{X - XMin}{XMax - XMin} \quad (1)$$

where X is the original value, X*Min* is the minimum value of the feature, and X*Max* is the maximum value of the feature. After this transformation, the scaled data can be used for machine learning algorithms and ensures that each feature contributes equally to the model training process without being disproportionately influenced by its original scale [15].

In the final preprocessing step, the data were normalized, and subsequently, divided into two subsets referred to as training and testing data. The split was carried in a way in which 70% of the total data was designated for training, while the remaining 30% was allocated for testing. This division enabled the training and evaluation of the machine learning classifier, allowing its accuracy to be tested on the same dataset, during both the training and testing phases.

3.3 Support vector machine

Support vector machines (SVMs) are a powerful supervised learning algorithm used for classification and regression tasks [16]. They are effective in high-dimensional spaces, making them suitable for tasks with large features, particularly in fields like bioinformatics and text classification. SVMs are less prone to overfitting and can handle non-linear decision boundaries. They can be versatile, using the kernel trick to map input data into high-dimensional feature spaces. SVMs also have global optimization, making them less dependent on initialization and more reliable. They are effective in small sample sizes, memory-efficient, and can model complex decision boundaries using different kernel functions. However, the choice of algorithm depends on the specific characteristics of the data and the task.

Additionally, SVM models can be computationally expensive and require careful tuning of hyperparameters to achieve optimal performance [17]. The algorithm operates by maximizing the margin between the classes, as delineated in the feature space. Furthermore, SVM models may struggle with handling imbalanced datasets, where the number of positive cases (heart disease patients) is significantly smaller than the number of negative cases (healthy individuals). These limitations should be taken into consideration

when evaluating the potential of SVM for heart disease detection.

3.4 Hyperparameter Optimization

Hyperparameter optimization is crucial in machine learning algorithms because it involves tuning the configuration settings of a model, known as hyperparameters, to achieve the best performance [18]. Hyperparameters are parameters that are not learned from the training data but are set prior to training. They significantly impact the model's learning process and its ability to generalize well to unseen data. It improves performance by enhancing accuracy, reducing overfitting, and generalizing to new data. It also aids in time and resource efficiency by automating the search for optimal hyperparameter values. Well-tuned hyperparameters contribute to model robustness, as they are less sensitive to input data variations and perform consistently across different datasets. They also help avoid underfitting and overfitting by striking the right balance between hyperparameter values.

Common techniques for hyperparameter optimization include grid search, random search, and more advanced methods such as Bayesian optimization and genetic algorithms. The choice of method depends on factors like the size of the hyperparameter search space, available computing resources, and the specific characteristics of the problem at hand. [19] introduces a proficient machine learning (ML) diagnosis system designed to detect heart disease. Prior to model implementation, optimal accuracy is achieved by employing the GridsearchCV hyperparameter method and the five-fold cross-validation technique.

Grid Search is a hyperparameter optimization technique employed in machine learning to systematically explore various combinations of hyperparameter values within predefined ranges. In the context of Support Vector Machines (SVM), a popular classification algorithm, Grid Search involves the exhaustive evaluation of different sets of hyperparameters. For SVM, two crucial hyperparameters are the regularization parameter (C) and the kernel coefficient (gamma). As an example, if we consider C values of [10, 100, 1000] and gamma values of [0.001, 0.01], the Grid Search would systematically test all possible combinations, resulting in the evaluation of six sets: [10, 0.001], [10, 0.01], [100, 0.001], [100, 0.01], [1000, 0.001], and [1000, 0.01]. This exhaustive search process aids in identifying the combination of hyperparameters that optimally enhance the model's performance, ensuring a comprehensive exploration of the hyperparameter space.

By employing this grid search approach, the ability to identify the most optimized parameters for achieving the highest accuracy is constrained due to the limitation imposed on exploring only a predefined set of combinations. One might contemplate expanding the search space by considering every conceivable combination, such as setting C from 1 to 10,000 and Gamma from 0.001 to 0.9999. This method is referred to as brute force, wherein every potential combination is systematically tested, and the combination yielding the highest accuracy is selected. However, this approach is characterized by its exhaustive nature, making it impractical and time-consuming. Not only does it take an extensive amount of time to execute, but it also fails to guarantee the discovery of the truly optimal solution, rendering it impractical for practical use [20].

The genetic algorithm elevates the concept of grid search to a higher plane. While grid search systematically explores numerous combinations to determine the optimal parameters for the best accuracy in a dataset, the genetic algorithm takes this a step further. It not only tests a multitude of combinations but also refines its search by extracting promising combinations and attempting to generate even better ones from them. In essence, the genetic algorithm surpasses grid search by not only exploring a more extensive set of combinations but also evolving over time. It operates through generations, where a diverse set of combinations or solutions from one generation is used to spawn the next. This evolutionary process aims to converge towards an optimal solution [21]. The genetic algorithm, therefore, stands out for its breadth and depth compared to the simplicity and ease of implementation of grid search.

3.5 Support Genetic Algorithm (GA) Hyperparameter Optimization

Genetic algorithms work exactly as real life does [22] as shown in Figure 1. It starts with an initial random population, where each individual has a set of characteristics or traits. These individuals undergo selection, where only the fittest individuals are chosen to reproduce. Through crossover and mutation, new offspring with traits from their parents are created. This process is repeated over multiple generations, allowing the population to evolve and adapt to their

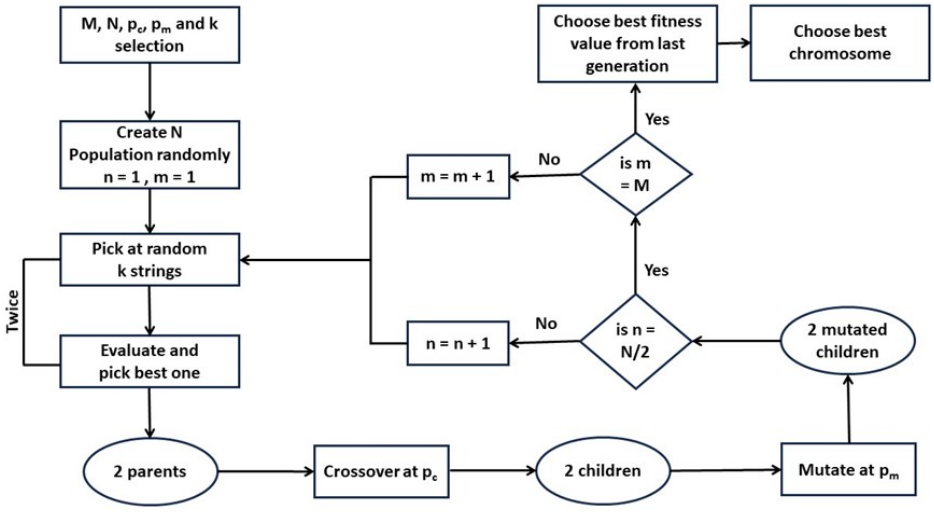

Figure 1 : FLOWCAHRT OF GENETIC ALGORITHM

environment, mimicking the principles of natural selection.

Consider a minimization problem defined over a set of feasible solutions, denoted as S, with a corresponding cost function f: S → R, applicable to all s ε S. The conventional approach involves computing the objective function for each s and selecting the minimum. However, practical real-life problems often entail an extensive S, rendering this direct approach impractical. This necessity for handling large solution spaces is where metaheuristics come into play. In our research, we specifically employ the Genetic Algorithm, a widely utilized metaheuristic, to address this challenge.

Genetic Algorithm – Parameter Notations

M – number of generations

N – population size

pc – probability of crossover

pm - probability of mutation

k – tournament selection

3.6 Integration of SVM and GA

In this research endeavor, our primary objective is to determine optimal values for the gamma and C hyperparameters in our support vector machine algorithm, focusing on a selected dataset. However, a crucial preliminary step involves understanding the process of calculating the objective function value. In the context of discussing genetic algorithm operators and techniques, particularly in the context of parent selection for crossover, it is essential to emphasize the significance of identifying optimal parents for the process. When selecting three, four, or five random solutions, the criterion for determining the best among them is crucial. In this context, "best" refers to the chromosome or solution that exhibits the highest accuracy. Each solution is a result of a combination of C and Gamma values, and this combination determines the accuracy of the solution.

This selection process is not only pertinent to parent selection but is also applicable when determining the final answer. To make informed decisions, it is imperative to calculate the objective function value or fitness value. This computation allows us to ascertain the accuracy and fitness of the chromosome. However, before delving into accuracy or fitness value calculations, a critical initial step involves decoding or translating the binary representation (zeros and ones) of the chromosome into actual numerical values. When preparing to input a chromosome into the support vector machine, it is impractical to feed it in binary form. Therefore, understanding the decoding process is pivotal to transform binary information into meaningful numerical data for effective integration into the support vector machine.

Decoding the chromosome in the context of genetic algorithms involves translating a binary representation within a specified range (a, b) into a meaningful numerical value [23]. The chromosome length, denoted as l, determines the precision of this decoding process.

Range (a, b)
Chromosome length, l

$$Precision = \frac{b-a}{2^l - 1} \quad (2)$$

$$Decoding = (\sum bit * 2i) * precision + a \quad (3)$$

Precision is calculated as in (1), indicating the granularity of the possible values. The decoding equation (2) is fundamental to the conversion. Here, 'bit' represents each binary digit in the chromosome, and 'i' denotes its position. The summation captures the cumulative contribution of each bit to the overall value. Multiplying by precision scales the sum to fit within the specified range and adding 'a' ensures the final decoded value falls within the desired interval (a, b). This systematic approach allows genetic algorithms to interpret and manipulate binary representations of potential solutions within a defined numerical context.

The high-level pseudocode in Algorithm 1 outlines the optimization process for hyperparameters in a Support Vector Machine (SVM) using a genetic algorithm. This includes decoding a binary chromosome, calculating hyperparameter values, performing k-fold cross-validation, training SVM models, and returning the optimized hyperparameters along with the average error. The main program initializes the chromosome and obtains the best hyperparameters.

Algorithm 1: Decoding a binary chromosome

Input: x, y, chromosome
Output: c_hyperparameter, gamma_hyperparameter, avg_error

1: Define lower and upper bounds for hyperparameters
2: Calculate precision for hyperparameters
3: Initialize variables for decoding binary chromosome
4: Decode binary chromosome for hyperparameter C
5: Initialize variables for decoding binary chromosome (gamma)
6: Decode binary chromosome for hyperparameter gamma
7: Calculate hyperparameter values
8: Initialize KFold cross-validation
9: Initialize sum_of_error variable
10: Perform cross-validation
11: Calculate average error across folds
12: Return tuned hyperparameter values and average error

In the genetic algorithm methodology, particularly when employing tournament selection, roulette wheel, or any other selection mechanism, the process involves choosing pairs of parents for crossover, leading to the generation of two offspring. Subsequently, these offspring undergo mutation, resulting in two mutated children, and this iterative cycle continues. The crux of the matter lies in the precise method used to select parents from the population, whether it be the initial population or one from a specific generation.

In the case of tournament selection, for instance, the procedure entails randomly selecting three or four solutions (chromosomes) from the existing population. Each of these solutions is represented by a binary sequence of zeros and ones. The next step involves evaluating the accuracy or error for each of these three randomly chosen chromosomes. The parent is then determined by selecting the one with either the highest accuracy or the lowest error among the three, in our case we are using the lowest error.

This selection process is iteratively performed, where another set of three solutions is randomly chosen, and the competition is repeated to identify the second parent as shown in Algorithm 2. Therefore, when formulating the function for selecting parents, it is imperative to incorporate a mechanism that allows for the random selection of three, four, or any specified number of solutions to ensure the diversity and effectiveness of the genetic algorithm. The initiation of a genetic algorithm necessitates the creation of an initial population. This initial population is best generated through a randomized approach, wherein zeros and ones are randomly selected to form the binary representation of individuals within the population. This randomness ensures diversity and unpredictability in the initial set of solutions, laying the foundation for the subsequent genetic algorithm iterations.

Algorithm 2: Selecting Parents based on fitness value

Input: x, y, Population
Output:
parent_1: First selected parent.
parent_2: Second selected parent.

1: Initialize parents as an empty matrix
2: for i in range(2) do
3: Generate a list of 3 random indices without replacement
4: Obtain possible parents using the randomly chosen indices
5: Compute objective function values for each possible parent
6: Find the minimum objective function value among the possible parents
7: Select the parent with the minimum objective function value
8: Update the parents matrix with the selected parent
9: end for
10: Extract the first and second selected parents from the parents matrix

The probability of crossover is a crucial parameter governing the frequency with which genetic material from two parents is combined to generate offspring. This probability is a user-defined parameter, and it can range from 0 to 100 percent. Opting for a probability of 100 percent implies a consistent application of crossover in every reproduction cycle, ensuring a continual exchange of genetic information between parents. However, users have the flexibility to adjust this probability to lower values, such as 90 or 85 percent, if a less frequent occurrence of crossover is desired.

Maintaining a probability of crossover at 100 percent implies an unwavering commitment to the recombination of parental genetic material, resulting in crossover events transpiring in every reproduction iteration. Consequently, it is imperative to explore the mechanisms by which crossover is executed in the context of this full probability setting.

To effectuate crossover between two parents, a two-point crossover method is employed. The initial step involves the random selection of two integers to determine the positions at which the crossover will transpire. It is noteworthy that if the two randomly selected indices happen to be identical, a reselection process is initiated to ensure distinct indices are chosen. This precautionary measure is implemented to guarantee the efficacy of the crossover operation.

Subsequently, the crossover process unfolds as follows: the genetic material preceding the first selected index from Parent one is combined with the genetic material succeeding the second selected index from Parent two to form the genetic makeup of Child one. In parallel, the genetic material preceding the first selected index from Parent two is paired with the genetic material succeeding the second selected index from Parent one to produce the genetic composition of Child two. This two-point crossover strategy thus facilitates the continual generation of diverse offspring by recombining genetic material from both parents in a systematic and reproducible manner as mentioned in Algorithm 3.

Algorithm 3: Crossover to produce the next generation

Input:
parent_1: First parent for crossover.
parent_2: Second parent for crossover.
prob_crsvr: Probability of crossover (default value is 1).
Output:
child_1: Offspring resulting from crossover with parent_1 and parent_2.
child_2: Offspring resulting from crossover with parent_1 and parent_2.

1: Initialize empty matrices for child_1 and child_2
2: Generate a random number to determine whether to perform crossover
3: if rand_num_to_crsvr_or_not < prob_crsvr:
4: Generate two random indices for crossover points, ensuring they are different
5: Identify the smaller and larger index to define the crossover segments
6: Obtain segments from both parents before, between, and after the crossover points
7: Create child_1 by combining segments of parent_1 and parent_2
8: Create child_2 by combining segments of parent_2 and parent_1
9: else:
10: Set child_1 to be the same as parent_1
11: Set child_2 to be the same as parent_2
12: Return child_1 and child_2 as output

Now that we have completed the crossover operation section and obtained two children from the two parents, it's time to introduce some variety and diversify our solutions. To achieve this, we have implemented the mutation operator. This operator acts on a specific child with a certain probability, causing alterations in its binary representation (zeros and ones). If a one is encountered, it will transform into a zero, and vice versa, based on a random number comparison.

When applying the mutation operator to a child, we iterate through its genes or alleles one by one. Starting with the first gene, we initialize an empty array with the same length as the child to store the mutated version. This array is sorted at index zero, as we proceed through the genes sequentially. The mutation occurs if the random number generated is less than the mutation probability. In such cases, a zero may become a one or a one may become a zero as depicted in Algorithm 4. However, if the random number exceeds the mutation probability, no mutation takes place for that particular gene.

Algorithm 4: Random Mutation of individuals

Input:
child_1: First child for mutation.
child_2: Second child for mutation.
prob_mutation: Probability of mutation (default value is 0.2).
Output:
mutated_child_1: Offspring resulting from mutation on child_1.
mutated_child_2: Offspring resulting from mutation on child_2.

1: Initialize empty matrices for mutated_child_1 and mutated_child_2
2: Initialize a variable t to 0
3: for each element i in child_1 do
4: Generate a random number to determine whether to perform mutation on child_1
5: if rand_num_to_mutate_or_not_1 < prob_mutation:
6: if child_1[t] is 0, set it to 1; if it is 1, set it to 0
7: Update mutated_child_1 with the modified child_1
8: Increment t by 1
9: else:
10: Update mutated_child_1 without modification
11: Increment t by 1
12: Repeat step 2-11 for child_2
13: Return mutated_child_1 and mutated_child_2 as output

4. RESULTS AND ANALYSIS

This research investigated the optimization of hyperparameters for an SVM machine learning model using a genetic algorithm with the Cardiovascular Heart Disease Dataset, sourced from the Mendeley database. To achieve favorable outcomes, Genetic Algorithms (GA) typically necessitate a sizable population size to guarantee ample variability within the elements present in the gene pool. For GA-Support Vector Machines (SVM), we opt for a population size of 50 and 100 generations as our objective is to identify the optimal solution for each generation. It is essential to maintain a record of the best-performing entity within each generation. This compilation of the best-performing individuals is crucial as, upon completion of 100 generations, we create a comprehensive list or array. This array encapsulates the elite representatives from each generation. For example, the first entry corresponds to the best chromosome from the initial generation, and subsequent entries mirror the most exceptional mutants or offspring from subsequent generations. Ultimately, the last entry in this collection represents the pinnacle of achievement, embodying the finest chromosome or most adeptly mutated progeny from the final generation.

Ultimately, if there is no inclination to examine the final result of our genetic algorithm's convergence, a straightforward approach is selecting the very best mutated child observed throughout all generations. This involves a systematic process wherein, at the conclusion of each generation, we meticulously identify and preserve the chromosome possessing the least error. In essence, as the algorithm progresses, the superior chromosome from each generation—characterized by the minimal error—is diligently stored. Consequently, if the genetic algorithm fail to converge to a desirable solution, recourse is readily

```
Final Solution (Convergence): [0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1.]
Encoded C (Convergence): [0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 0. 1. 1.]
Encoded Gamma (Convergence): [0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1.]
Final Solution (Best): [0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0.]
Encoded C (Best): [0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 1. 1.]
Encoded Gamma (Best): [0. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0.]

Decoded C (Convergence): 675.07692
Decoded Gamma (Convergence): 0.04278
Obj Value - (Convergence): 0.966

Decoded C (Best): 646.30769
Decoded Gamma (Best): 0.02443
Obj Value - (Best): 0.97
```

Figure 2: OPTIMIZED PARAMETER CONFIGURATION FOR MINIMIZED ERROR VALUE

available by revisiting the compilation of the best-performing entities from each generation. By selecting the optimal chromosome with the lowest recorded error, one can effectively pinpoint a robust solution.

Our objective is to identify the optimal solution among the mutated children in the 100th generation. This entails pinpointing the solution toward which the algorithm converges or reaches at the conclusion of all 100 generations. A genetic algorithm converges, it implies that the algorithm has identified a solution that meets the optimization criteria, and further iterations may not result in substantial improvements [24]. However, our interest extends beyond merely the best solution in the final generation; rather, we aim to ascertain the best solution across all generations. In essence, we seek two chromosomes as shown in Figure 2: the one representing the optimal solution at the algorithm's convergence point and the other representing the best solution observed throughout generations 1 to 100. The accuracy of the latter can be calculated by subtracting the objective function value from one.

We used the best value of C and Gamma to create the model and then evaluated the model using the confusion matrix as shown in Figure 3. It generates four outcomes: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative).

	precision	recall	f1-score	support
0	0.98	0.95	0.96	130
1	0.97	0.98	0.97	170
accuracy			0.97	300
macro avg	0.97	0.97	0.97	300
weighted avg	0.97	0.97	0.97	300

```
[[124   6]
 [  3 167]]
```

Figure 3: CONFUSION MATRIX

We then use these measures to calculate accuracy, sensitivity, and specificity which can be defined as follows:

Accuracy: This is the proportion of correctly classified instances (both true positives and true negatives) among the total number of instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$= \frac{167 + 124}{167 + 124 + 6 + 3}$$

$$= 0.97$$

Sensitivity (Recall): This is the proportion of true positives that are correctly identified, out of all actual positives.

$$Sensitivity = \frac{TP}{TP + FN} \quad (5)$$

$$= \frac{167}{167 + 3}$$

$$= 0.98$$

Specificity: This is the proportion of true negatives that are correctly identified, out of all actual negatives.

$$Specificity = \frac{TN}{TN + FP} \quad (6)$$

$$= \frac{124}{124 + 6}$$

$$= 0.95$$

Precision: This is the proportion of true positives out of all instances classified as positive.

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$= \frac{167}{167 + 6}$$

$$= 0.97$$

F1-score: This is the harmonic mean of precision and recall, giving a balance between the two metrics.

$$F1 - score = \frac{2 * Precision * Sensitivity}{Precision + Sensitivity} \quad (8)$$

$$= \frac{2 * 0.97 * 0.98}{0.97 + 0.98}$$

$$= 0.97$$

The area under the ROC curve (AUC-ROC) is a commonly used metric to quantify the overall performance of a binary classification model. A higher AUC-ROC value (closer to 1) indicates better discrimination ability of the model across different threshold settings as shown in Figure 4.

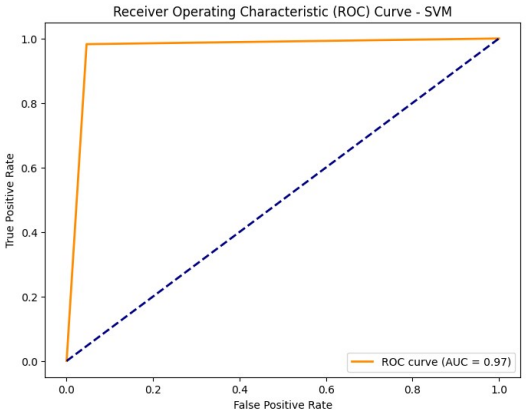

Figure 4: ROC CURVE

By comparing the results of the proposed model with existing ones, benchmarking provides valuable insights into the strengths and weaknesses of each approach. This approach helps determine whether the proposed method outperforms existing ones and enhances accuracy. Table 2 illustrates the comparison between different models and our proposed method. The proposed method employs an SVM classifier along with optimal hyperparameters determined by Genetic Algorithm. These results demonstrate the effectiveness of our approach in accurately predicting heart disease in patients when compared to existing models.

Gupta & Seth [25] vs. Bhowmick et al. [26] studies utilize the UCI Heart Disease dataset with the same number of samples (303) and features (13). Gupta & Seth achieved a higher accuracy of 86.89%, compared to Bhowmick et al. who achieved 83%. This suggests that Gupta & Seth's model outperformed Bhowmick et al.'s model in terms of accuracy on the UCI Heart Disease dataset. Assegie et al.[27] and Abdar et al. also used the UCI Heart Disease dataset, but Abdar et al.[28] additionally used the Mendeley Heart Disease dataset. Assegie et al. achieved an accuracy of 85.2%, while Abdar et al. achieved a slightly higher accuracy of 86.05%. The proposed model (Mendeley Heart Disease dataset) achieved the highest accuracy of 97%.

Table 2: Benchmarking Of The Proposed Solution With Existing Heart Disease Prediction

S.No.	Authors	Dataset	No. of Samples	No. of Features	Accuracy
1	Gupta & Seth, 2023[25]	UCI Heart Disease	303	13	86.89%
2	Bhowmick et al., 2022[26]	UCI Heart Disease	303	13	83%
3	Assegie et al., 2022 [27]	UCI Heart Disease	303	11	85.2%
4	Abdar et al., 2015[28]	UCI Heart Disease	271	13	86.05%
5	Proposed Model	Mendeley Heart Disease	1000	12	97%

5. CONCLUSION

In conclusion, this research has underscored the significance of integrating genetic algorithms with machine learning techniques to enhance the predictive accuracy, generalizability, and scalability of heart disease prediction models. This research paper uses a metaheuristics approach with genetic algorithms to improve the predictive accuracy of heart disease models. The method involves optimizing hyperparameters and minimizing errors. The goal is to converge towards a specific value or accuracy, with the emphasis on minimizing error to enhance performance. By minimizing error and maximizing accuracy, the SVM classifier with optimized hyperparameters using a genetic algorithm has demonstrated its potential to significantly improve disease prediction accuracy. The SVM classifier with optimized hyperparameters showed a 97% improvement in accuracy when applied to the heart disease dataset, outperforming previous studies. This success in the heart disease dataset suggests that further exploration of different disease datasets could yield even more promising results, ultimately leading to more accurate diagnoses and better treatment outcomes for patients. Overall, this research contributes to advancing the field of cardiovascular disease prediction by introducing a novel approach that effectively optimizes machine learning models using genetic algorithms, ultimately improving the accuracy and reliability of heart disease prediction.

Future research could explore various disease datasets to improve disease prediction accuracy. By applying the model to different disease datasets, researchers can gain insights into the model's generalizability and potential limitations. Additionally, exploring various disease datasets can help identify patterns and correlations that may not be apparent in a single dataset, further improving disease prediction accuracy and treatment outcomes.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support provided by the **"Visvesvaraya PhD Scheme for Electronics & IT."** We extend our sincere appreciation for the opportunities and resources made available through this scheme, which have significantly contributed to the success of our research endeavors.

DECLARATIONS

CONFLICT OF INTEREST

No conflict of interest in this manuscript

ETHICAL APPROVAL

This research involved the data obtained from an online source https://data.mendeley.com/datasets/dzz48mvjht/1. The use of this data was in accordance with the terms of use specified by the data provider and adhered to all relevant ethical guidelines for the use of secondary data. No additional ethical approval was required for this study as it involved the analysis of pre-existing, anonymized data.

FUNDING

This research did not receive any external funding.

AVAILABILITY OF DATA AND MATERIALS

The dataset analyzed during the current study is available online at https://data.mendeley.com/datasets/dzz48mvjht/1

REFERENCES:

[1] J. A. Finegold, P. Asaria, and D. P. Francis, "Mortality from ischaemic heart disease by country, region, and age: Statistics from World Health Organisation and United Nations," *Int. J. Cardiol.*, vol. 168, no. 2, pp. 934–945, 2013, doi: 10.1016/j.ijcard.2012.10.046.

[2] W. Wang *et al.*, "Mortality and years of life lost of cardiovascular diseases in China, 2005–2020: Empirical evidence from national mortality surveillance system," *Int. J. Cardiol.*, vol. 340, pp. 105–112, 2021, doi: 10.1016/j.ijcard.2021.08.034.

[3] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, and A. Saboor, "Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare," *IEEE Access*, vol. 8, no. Ml, pp. 107562–107582, 2020, doi: 10.1109/ACCESS.2020.3001149.

[4] S. Lee, Y. Chu, J. Ryu, Y. J. Park, S. Yang, and S. B. Koh, "Artificial Intelligence for Detection of Cardiovascular-Related Diseases from Wearable Devices: A Systematic Review and Meta-Analysis," *Yonsei Med. J.*, vol. 63, pp. S93–S107, 2022, doi:

10.3349/ymj.2022.63.S93.

[5] N. Salari, S. Shohaimi, F. Najafi, M. Nallappan, and I. Karishnarajah, "A novel hybrid classification model of genetic algorithms, modified k-nearest neighbor and developed backpropagation neural network," *PLoS One*, vol. 9, no. 11, pp. 1–50, 2014, doi: 10.1371/journal.pone.0112987.

[6] N. G. B. Amma, "Cardiovascular disease prediction system using genetic algorithm and neural network," *2012 Int. Conf. Comput. Commun. Appl. ICCCA 2012*, 2012, doi: 10.1109/ICCCA.2012.6179185.

[7] W. Rajab and V. Sharma, "A Hybrid AI approach based on Genetic Algorithm and ANFIS for Heart Disease Diagnosis," no. July, 2018.

[8] S. Aalaei, H. Shahraki, A. Rowhanimanesh, and S. Eslami, "Feature selection using genetic algorithm for breast cancer diagnosis: Experiment on three different datasets," *Iran. J. Basic Med. Sci.*, vol. 19, no. 5, pp. 476–482, 2016.

[9] A. Ghaheri, S. Shoar, M. Naderan, and S. S. Hoseini, "The applications of genetic algorithms in medicine," *Oman Med. J.*, vol. 30, no. 6, pp. 406–416, 2015, doi: 10.5001/omj.2015.82.

[10] R. Guido, M. C. Groccia, and D. Conforti, "A hyper-parameter tuning approach for cost-sensitive support vector machine classifiers," *Soft Comput.*, vol. 27, no. 18, pp. 12863–12881, 2023, doi: 10.1007/s00500-022-06768-8.

[11] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A Survey on Evolutionary Computation Approaches to Feature Selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, 2016, doi: 10.1109/TEVC.2015.2504420.

[12] "Cardiovascular Disease Dataset Mendeley Data." https://data.mendeley.com/datasets/dzz48mvjht/1 (accessed Jan. 26, 2024).

[13] A. O. Mocumbi, E. Lameira, A. Yaksh, L. Paul, M. B. Ferreira, and D. Sidi, "Challenges on the management of congenital heart disease in developing countries," *Int. J. Cardiol.*, vol. 148, no. 3, pp. 285–288, 2011, doi: 10.1016/j.ijcard.2009.11.006.

[14] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Front. Energy Res.*, vol. 9, no. March, pp. 1–17, 2021, doi: 10.3389/fenrg.2021.652801.

[15] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[17] S. Lessmann, R. Stahlbock, and S. F. Crone, "Optimizing hyperparameters of support vector machines by genetic algorithms," *Proc. 2005 Int. Conf. Artif. Intell. ICAI'05*, vol. 1, no. May 2014, pp. 74–80, 2005.

[18] A. M. Vincent and P. Jidesh, "An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms," *Sci. Rep.*, vol. 13, no. 1, pp. 1–19, 2023, doi: 10.1038/s41598-023-32027-3.

[19] N. Chandrasekhar and S. Peddakrishna, "Enhancing Heart Disease Prediction Accuracy through Machine Learning Techniques and Optimization," *Processes*, vol. 11, no. 4, 2023, doi: 10.3390/pr11041210.

[20] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, vol. 14, no. 4, p. 1502, 2016, doi: 10.12928/telkomnika.v14i4.3956.

[21] A. H. Wright, *Genetic Algorithms for Real Parameter Optimization*, vol. 1. Morgan Kaufmann Publishers, Inc., 1991.

[22] S. Forrest, "Principles of Genetic Algorithms," vol. 261, no. August, pp. 60–76, 2020, doi: 10.4018/978-1-7998-1920-2.ch004.

[23] Z. Michalewicz, "Binary or Float? BT - Genetic Algorithms + Data Structures = Evolution Programs," Z. Michalewicz, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 75–82.

[24] G. Rudolph, "Convergence Analysis of Canonical Genetic Algorithms," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994, doi: 10.1109/72.265964.

[25] P. Gupta and D. Seth, "Comparative analysis and feature importance of machine learning and deep learning for heart disease prediction," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 29, no. 1, pp. 451–459, 2023, doi:

10.11591/ijeecs.v29.i1.pp451-459.

[26] A. Bhowmick, K. D. Mahato, C. Azad, and U. Kumar, "Heart Disease Prediction Using Different Machine Learning Algorithms," *Proc. - 2022 IEEE World Conf. Appl. Intell. Comput. AIC 2022*, pp. 60–65, 2022, doi: 10.1109/AIC55036.2022.9848885.

[27] T. A. Assegie, P. K. Rangarajan, N. K. Kumar, and D. Vigneswari, "An empirical study on machine learning algorithms for heart disease prediction," *IAES Int. J. Artif. Intell.*, vol. 11, no. 3, pp. 1066–1073, 2022, doi: 10.11591/ijai.v11.i3.pp1066-1073.

[28] M. Abdar, S. R. N. Kalhori, T. Sutikno, I. M. I. Subroto, and G. Arji, "Comparing performance of data mining algorithms in prediction heart diseses," *Int. J. Electr. Comput. Eng.*, vol. 5, no. 6, pp. 1569–1576, 2015, doi: 10.11591/ijece.v5i6.pp1569-1576.