



Article

# Multilayer Perceptron Neural Network with Arithmetic Optimization Algorithm-Based Feature Selection for Cardiovascular Disease Prediction

Fahad A. Alghamdi <sup>1</sup>, Haitham Almanaseer <sup>2</sup>, Ghaith Jaradat <sup>2</sup>, Ashraf Jaradat <sup>3</sup>, Mutasem K. Alsmadi <sup>1,\*</sup>, Sana Jawarneh <sup>4</sup>, Abdullah S. Almurayh <sup>5</sup>, Jihad Alqurni <sup>5</sup> and Hayat Alfagham <sup>1</sup>

<sup>1</sup> Department of MIS, College of Applied Studies and Community Service, Imam Abdulrahman Bin Faisal University, Dammam P.O. Box 1982, Saudi Arabia; faghamdi@iau.edu.sa (F.A.A.); hmalgham@aiu.edu.sa (H.A.)

<sup>2</sup> Department of CS, Faculty of Computer Sciences and Informatics, Amman Arab University, Amman P.O. Box 2234-11953, Jordan; haitham\_manaseer@yahoo.com (H.A.); g.jaradat@aau.edu.jo (G.J.)

<sup>3</sup> College of Business Administration, American University of the Middle East, Egaila 54200, Kuwait; ashraf.jaradat@aum.edu.kw

<sup>4</sup> Computer Science Department, The Applied College, Imam Abdulrahman Bin Faisal University, Dammam P.O. Box 1982, Saudi Arabia; sijawarneh@iau.edu.sa

<sup>5</sup> Department of Educational Technologies, College of Education, Imam Abdulrahman Bin Faisal University, Dammam P.O. Box 1982, Saudi Arabia; asalmurayh@iau.edu.sa (A.S.A.); jalqurni@iau.edu.sa (J.A.)

\* Correspondence: mailto:mkalsmadi@iau.edu.sa

**Abstract:** In the healthcare field, diagnosing disease is the most concerning issue. Various diseases including cardiovascular diseases (CVDs) significantly influence illness or death. On the other hand, early and precise diagnosis of CVDs can decrease chances of death, resulting in a better and healthier life for patients. Researchers have used traditional machine learning (ML) techniques for CVD prediction and classification. However, many of them are inaccurate and time-consuming due to the unavailability of quality data including imbalanced samples, inefficient data preprocessing, and the existing selection criteria. These factors lead to an overfitting or bias issue towards a certain class label in the prediction model. Therefore, an intelligent system is needed which can accurately diagnose CVDs. We proposed an automated ML model for various kinds of CVD prediction and classification. Our prediction model consists of multiple steps. Firstly, a benchmark dataset is preprocessed using filter techniques. Secondly, a novel arithmetic optimization algorithm is implemented as a feature selection technique to select the best subset of features that influence the accuracy of the prediction model. Thirdly, a classification task is implemented using a multilayer perceptron neural network to classify the instances of the dataset into two class labels, determining whether they have a CVD or not. The proposed ML model is trained on the preprocessed data and then tested and validated. Furthermore, for the comparative analysis of the model, various performance evaluation metrics are calculated including overall accuracy, precision, recall, and F1-score. As a result, it has been observed that the proposed prediction model can achieve 88.89% accuracy, which is the highest in a comparison with the traditional ML techniques.

**Keywords:** feature selection; cardiovascular diseases; multilayer perceptron; neural network; arithmetic optimization algorithm



**Citation:** Alghamdi, F.A.; Almanaseer, H.; Jaradat, G.; Jaradat, A.; Alsmadi, M.K.; Jawarneh, S.; Almurayh, A.S.; Alqurni, J.; Alfagham, H. Multilayer Perceptron Neural Network with Arithmetic Optimization Algorithm-Based Feature Selection for Cardiovascular Disease Prediction. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 987–1008. <https://doi.org/10.3390/make6020046>

Academic Editor: Isaac Triguero

Received: 13 February 2024

Revised: 12 April 2024

Accepted: 19 April 2024

Published: 5 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

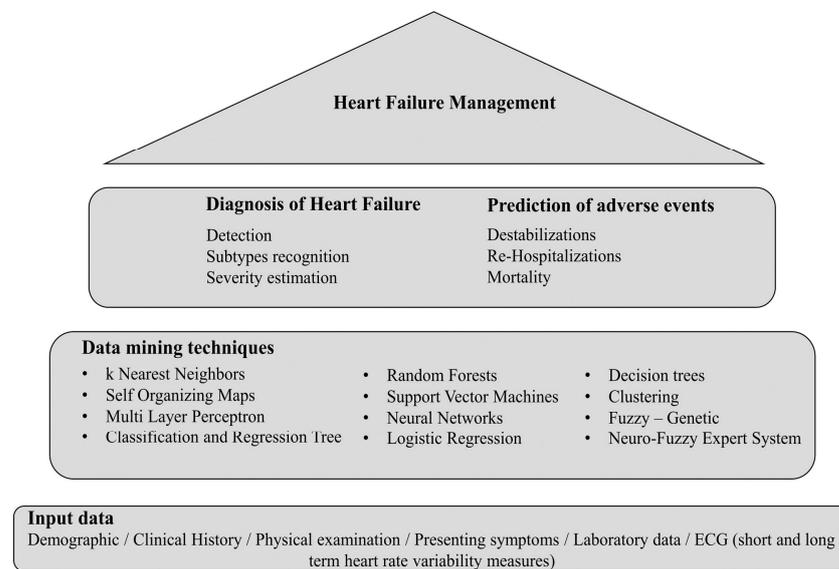
## 1. Introduction

Cardiovascular disease (CVD) is a critical health problem caused by disorders in the heart and blood vessels, and it is one of the significant causes of mortality worldwide. CVD includes several categories, such as coronary heart disease, strokes, and others. Machine learning (ML) has been effectively applied in biomedicine fields, introducing easy methods of photovoltaic power forecasting as well as diagnosing diseases such as CVD and other

diseases [1,2]. Medical data are usually significant in volume. The analysis of extensive data requires a lot of resources and time for implementation, which increases the computational complexity and reduces the efficiency of ML models. Furthermore, not all features in the dataset may contribute effectively to CVD diagnosis.

According to the World Heart Federation (WHF), more than 17 million people die from CVDs yearly, and the World Health Organization (WHO) states that the leading cause of death worldwide is CVD [3]. About 26 million adults are diagnosed with heart disease each year, according to the European Society of Cardiology (ESC). Accurate and early diagnosis of CVD risk in patients is essential in reducing its associated risks [3]. Many reports indicate that CVD is one of the main causes of sudden death in industrialized countries. These increasing death rates, especially prominent in developed countries, affect the health, financial resources, and budgets of individuals [3]. Diagnosing and treating heart diseases are very complicated procedures, especially in developing countries, due to the scarcity of diagnostic devices and the lack of medical cadres and resources [3]. An electrocardiogram (ECG) is used and is considered the gold standard in CVD detection and its risk analysis. However, this method is expensive, requires a high level of technical expertise, and is time-consuming. Therefore, researchers must find cheaper and more effective alternative methods.

Based on Figure 1, a diagnosis is often made based on a doctor’s experience and assumptions. Therefore, it is possible to make wrong decisions that may be fatal in some cases. Hence, ML has become a popular method of efficiently predicting disease rather than relying exclusively on human knowledge [4]. With the increasing availability of electronic health data and the resolution of the complexities of CVD diagnosis, computational methods, such as support vector machine (SVM), K-nearest neighbor (K-NN), decision tree (DT), AdaBoost (AB), artificial neural network (ANN), etc., are becoming more applicable, with exploratory value in disease prediction [5].



**Figure 1.** A generic framework for heart failure management (e.g., CVD diagnosis).

Medical datasets may contain a large volume of data. One problem with analyzing such data is the dimensional curse. These data suffer from a higher dimension with fewer numbers. Reducing feature dimensions improves the effectiveness of prediction models. Therefore, it is necessary to find an efficient method for selecting significant features, removing irrelevant features, and paving the way for effective CVD prediction. Research is still ongoing to find the correct CVD prediction model. The performance of models drops significantly without a proper selection of significant features. Therefore, there is

an urgent need to find an efficient methodology for selecting them for the diagnosis of the disease [2,4].

This research aims to improve the accuracy of the CVD prediction model by developing a predictive model consisting of a multilayer perceptron neural network (MLPNN) with an arithmetic optimization algorithm (AOA) (MLPNN-AOA). The AOA is an optimization algorithm used to select the most relevant features [6]. The Cleveland dataset was utilized to evaluate the method employed. This dataset contains 304 instances and 14 features with a CVD diagnosis [7].

Many previous studies have examined CVD, including [8–12], but the present research is the first to utilize AOA to select the most relevant features for diagnosis of CVD. The primary goal of the AOA is to obtain optimal fitness solutions and the best performance convergence [6]. The performance of MLPNN-AOA was measured in terms of accuracy, the area under the receiver operating characteristic (AROC) curve, the mean square error (MSE), and the precision, recall, specificity, sensitivity, and F1-score. Many deep learning (DL) methods are widely applied to diagnose and predict CVD [13]. One of these methods is MLPNN, which is a type of artificial neural network (ANN) that generates a set of outputs from inputs. MLPNN is an example of a feed-forward neural network (FFNN). MLPNN consists of several input layers connected as a directed graph between the input and output layers [14]. Backpropagation is used for MLPNN training in MLPNN-AOA.

Feature selection (FS) is a measurable property of the observed process using a feature set that reduces computational requirements, improves prediction, and uses specific metrics to find a subset of features [15].

Several effective decision support models based on ML tools have been proposed widely in previous studies to detect CVD. However, most of these models focus on feature preprocessing only. Unfortunately, datasets may contain redundant and irrelevant features that affect prediction accuracy, precision, processing speed, and problems posed by the predictive model, such as underfitting and overfitting problems. These problems can be solved using MLPNN-AOA (e.g., via data augmentation, simplified neural network, or early stopping) to eliminate irrelevant features, find the best features, and increase prediction accuracy in both the training and testing datasets.

Hence, the primary objective of this research is to implement a predictive CVD-diagnosing model based on the AOA as an optimization algorithm and MLPNN as a prediction model.

The specific objectives of this research are:

1. To select the best features that affect the accuracy of MLPNN using an AOA.
2. To compare MLPNN-AOA with other similar models on the same dataset to verify its performance.
3. To compare AOA with some other optimization algorithms in selecting the most relevant features in the Cleveland dataset.
4. To eliminate the problems of overfitting and underfitting in the CVD prediction model by developing a hybrid MLPNN-AOA algorithm.

The practical importance of the research is due to the fact that medical diagnostics must be specialized, reliable, and supported by computer technologies to reduce the costs of diagnostic tests. Therefore, most researchers want to develop new algorithms to predict heart disease. The main contribution of this research is to develop an MLPNN-AOA prediction model, which has two characteristics. First, the combination of MLPNN and AOA extends the ability to learn and generalize across different specifications of CVD datasets. Second, the implementation of AOA focuses on selecting the most relevant features and then performing the prediction process. This may prevent overfitting or biased classification, ignore irrelevant and redundant features to increase prediction accuracy, and reduce classification time.

In addition, our research paper contributes in the following ways:

- Presents practical and academic knowledge to researchers.
- Helps health professionals, especially doctors, in CVD diagnosis.

- Supports anyone interested in optimization algorithms and ML techniques, especially in the utilization of AOA and MLPNN in many applications.

In this research, MLPNN-AOA is developed to select the most relevant features by AOA to reduce the dimensionality of the dataset that affects the accuracy of the CVD prediction model representing MLPNN. This research used a free, open-source dataset from the UCI Machine Learning repository of 76 attributes [7,16]. All published papers used a portion of this set of 14 features. The most relevant feature is the 'target', which indicates that the patient has CVD and represents integer values from 0 (the probability of no injury) to 4 (the high-rate probability of infection). Patient names and National Health Service user numbers (SNS) are removed from the dataset and replaced by dummy values. Different performance measures will be used to evaluate the effectiveness of MLPNN-AOA through accuracy, AROC, MSE, precision, recall, sensitivity, specificity, and F1-score.

The rest of this research is organized as follows. Section 2 presents a research background about ML-based FS and FS using optimization algorithms. In addition, a review of the previous literature for the CVD prediction model and an analysis of related studies are conducted to identify research gaps. Section 3 describes the solution that has been adopted to select the most relevant features and the approach that is used for CVD prediction. Section 4 describes the experiments that have been performed in this research and an analysis of the obtained results. It shows a comparison between MLPNN-AOA, MLPNN, DT, SVM, and RFC. In addition, it presents a comparison between the results of the AOA and the prior optimization algorithms in selecting the most relevant features in the Cleveland dataset. Finally, Section 5 gives the conclusions and suggestions for future work.

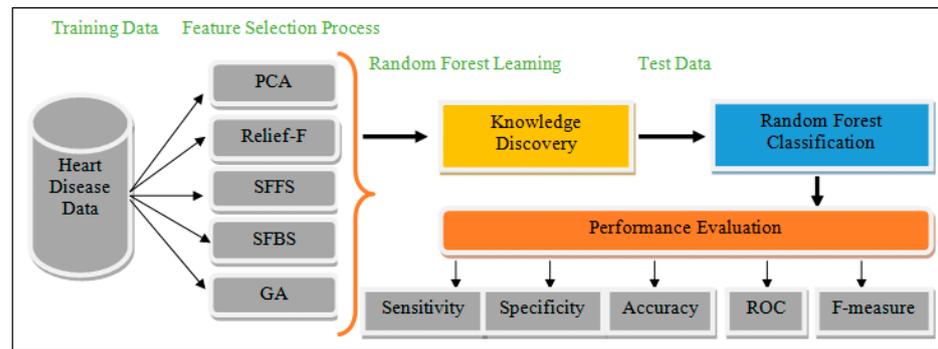
## 2. Literature Review

Recently, many ML techniques have been suggested for diagnosing CVD and examining its efficient predictive model. This section describes the soundest methods employed in this domain. In addition, it illustrates the importance of utilizing optimization algorithms to select only the relevant features in its in increasing the efficiency and effectiveness of prediction models.

Different ML algorithms are used to predict and classify data, relying on training data. A classification task is used to classify items in the dataset into a predefined set of class labels [2,17]. ML has three types: supervised ML, unsupervised ML, and reinforcement machine learning [2,18].

Supervised and unsupervised learning are used to overcome numerous issues with pattern recognition. In supervised learning, considerable classifiers are utilized to classify data, such as self-organizing maps, K-NN, and DT [19]. The training data are used to make a function. This includes a pair of input vectors and a class label. The training function is performed to evaluate the approximate length between the input and the output to build a classifier. When the classifier is created, the classification can be performed to classify novel classes based on known class labels [17]. In this research, MLPNN and FS were utilized for the CVD prediction. MLPNN is widely used in classification algorithms, and it has outstanding classification accuracy [17]. The FS selection is utilized to reduce the features in the dataset by choosing the most relevant one [20].

Feature selection (FS) is a difficult task that needs an optimization algorithm to select the best subset of features that has an impact on the classification accuracy. It is a method to analyze all attributes on a full dataset. Some suitable features for the issue are selected. Its main purpose is to improve classification accuracy and decrease computational time [21]. See Figure 2.



**Figure 2.** A general example of CVD prediction using supervised learning based on a feature selection mechanism.

Eliminating some attributes does not mean they are without important information, but they may not have significant statistical relationships with others. FS methods are required for evaluation and analysis. As demonstrated by [22] and applied by [1,2,23], FS starts with generating subsets from the whole dataset. Then, the evaluation function chooses the features associated with the problem by employing either a wrapper or a filter technique. Finally, the validation stage takes place for the model's efficiency and consistency. Further descriptions are provided by [23], where the types and capabilities of optimization algorithms are used for the FS task.

Currently, there are wide bodies of research covering a wide range of techniques, which can be used as an integral part of predicting CVD using ML methods. Accurate and timely CVD diagnosis is primary for the prevention and treatment of heart failure. Diagnosis of CVD by conventional medical history is unreliable in many respects. To classify healthy people and people with heart disease, noninvasive methods such as machine learning are reliable and effective [5].

A CVD prediction model was proposed by [24] using ML algorithms based on National Health Insurance Service Health Screening datasets (a cardiovascular disease group). An efficient two-layer convolutional neural network (CNN) was proposed by [25] to classify highly unbalanced clinical data for predicting the incidence of coronary heart disease (CHD). A study was presented by [26] that used many classification methods like SVM, naïve Bayes, DT, RFC, and logistic regression (LR) using the Waikato Environment for Knowledge Analysis (Weka) tool for predicting cardiovascular disease. The study of [27] used eight classification algorithms (DT, J48, logistic model tree, RFC, naïve Bayes, KNN, and SVM) to foresee heart disease and perform predictive analysis using data mining techniques to infer efficient algorithms from those algorithms.

In general, as demonstrated by the mentioned studies' techniques, their limitations are mainly in slow computation, due to the large dataset sizes. Hence, several state-of-the-art techniques have utilized optimization algorithms to perform a feature selection mechanism to select a subset of the most relevant data to reduce the dimensionality in the prediction model. The main objectives of feature selection are to avoid overfitting or mismatch, enhance generalization, improve model performance, reduce model training time, simplify the model, provide faster and more cost-effective models, and improve prediction and classification accuracy [28]. The selected feature set needs a search and routing mechanism for choosing the sub-feature. The objective of the job is to estimate specific features, the terms of termination, and the evaluation outcomes [15].

A hybrid algorithm, genetic algorithm–linear discriminant analysis (GA-LDA), was proposed by [4] for CAD diagnostics. A GA was combined with an LDA to identify and select significant features in the coronary heart disease dataset. A similar model was proposed by [29], the feature optimization by discrete weights (FODW) model. A hybrid model was proposed by [30] consisting of bi-directional long short-term memory with conditional random field (BiLSTM-CRF) to predict heart disease. An improved functionality based on SVM was also proposed by [31]. To select the most relevant features, GA was

used. A hybrid model was proposed by [32], consisting of a random search algorithm (RSA) for FS and an RFC for prediction. The proposed model has been improved using a network search algorithm. Similarly, a hybrid model (artificial neural network and deep neural network) was proposed by [33] to eliminate redundant features and a deep neural network for prediction. The proposed model achieved a prediction accuracy of 93.33%, but a limitation of time complexity is not determined. A hybrid ML-based cardiac diagnostic system was developed by [5] using a set of ML algorithms to select important features. Three algorithms were used to validate the proposed model: relief, mRMR, and LASSO. The K-fold validation method was used. A feature selection approach was proposed by [34] based on a multipurpose artificial bee colony algorithm combined with the nondominant screening procedure and genetic operators.

ML is used as an effective support system in health diagnosis that contains a large volume of data. More commonly, parsing such a large volume of data consumes more resources and execution time. In addition, not all features in the dataset support the solution to the specific problem. Thus, there is a need to use an efficient FS algorithm to find the most significant features that contribute the most to disease prognosis. Based on previous research, it is concluded that employing optimization methods to choose the most relevant features will improve the CVD model's accuracy, reduce its computational complexity and execution time, and reduce overfitting and underfitting issues. As a result of this research, an MLPNN-AOA algorithm is proposed, in which an AOA is employed to choose the most relevant features from the Cleveland dataset. Although many studies utilized different feature selection mechanisms, they lack a few capabilities due to the way they either implemented or coded the structure of these mechanisms. These limitations are the following: a normal distribution assumption on features is needed; they are not suitable for rare categories (imbalanced dataset); computation to create a cross-validation evaluation of some potential subsets is costly. In addition, the hybrid approaches are not scaled sufficiently with complexity, and most of them do not measure the AROC, the MSE, or the confusion matrix, which misleads the performance evaluation. In addition, the optimization algorithms (e.g., particle swarm optimization) used in the literature are mainly swarm-optimization-based algorithms, which easily lead to an early convergence towards a local optimum, and their iterative process results in a low convergence rate in general.

Hence, we intended to utilize the AOA proposed by [6]. It has the advantage that its implementation is so easy and direct; based on its mathematical presentation, it is able to adapt to and address new improvement problems and undertakes its execution according to a mathematical view. AOA is mathematically designed and implemented in vast areas of research to perform optimization processes [6].

### 3. Materials and Methods

Considerable research has obtained surprising results when using neural networks (NNs) in various applications [35]. NNs have two learning algorithm types: supervised and unsupervised learning [36]. The present research utilized supervised learning because it could conclude a general function depending on the training data, and it would be able to test the data reasonably. Optimization algorithms are used to select the most relevant features that positively affect the prediction model accuracy, execution time, and problems such as overfitting and underfitting [20].

In this research, the MLPNN-AOA model will be implemented. The AOA is used to select the most significant features in the Cleveland dataset, while MLPNN is used for prediction. According to the problem noted in the introduction section of the research problem section, it is necessary to find an adequate CVD prediction model. Therefore, AOA was utilized to select the relevant features and find the best ones on the Cleveland dataset. Then, the diagnosis is made by MLPNN. In general, the proposed methodology comprises five steps:

1. Extracting medical data that are obtained from the web in a tabular form containing different data types.
2. Data preprocessing using normalization techniques, e.g., chi-square and gain ratio, including handling missing data, then splitting the dataset into training and testing datasets.
3. The AOA optimizer is used for the feature selection task to determine the best subset of features from the training dataset.
4. The MLPNN classifier is then employed on the training dataset to train the prediction model for the classification task based on the best subset of features.
5. Finally, the MLPNN classifier is employed on the testing dataset for classifying the unlabeled data into two classes for the prediction model.

As shown in Figure 3, three main phases have been undertaken to implement the MLPNN-AOA model.

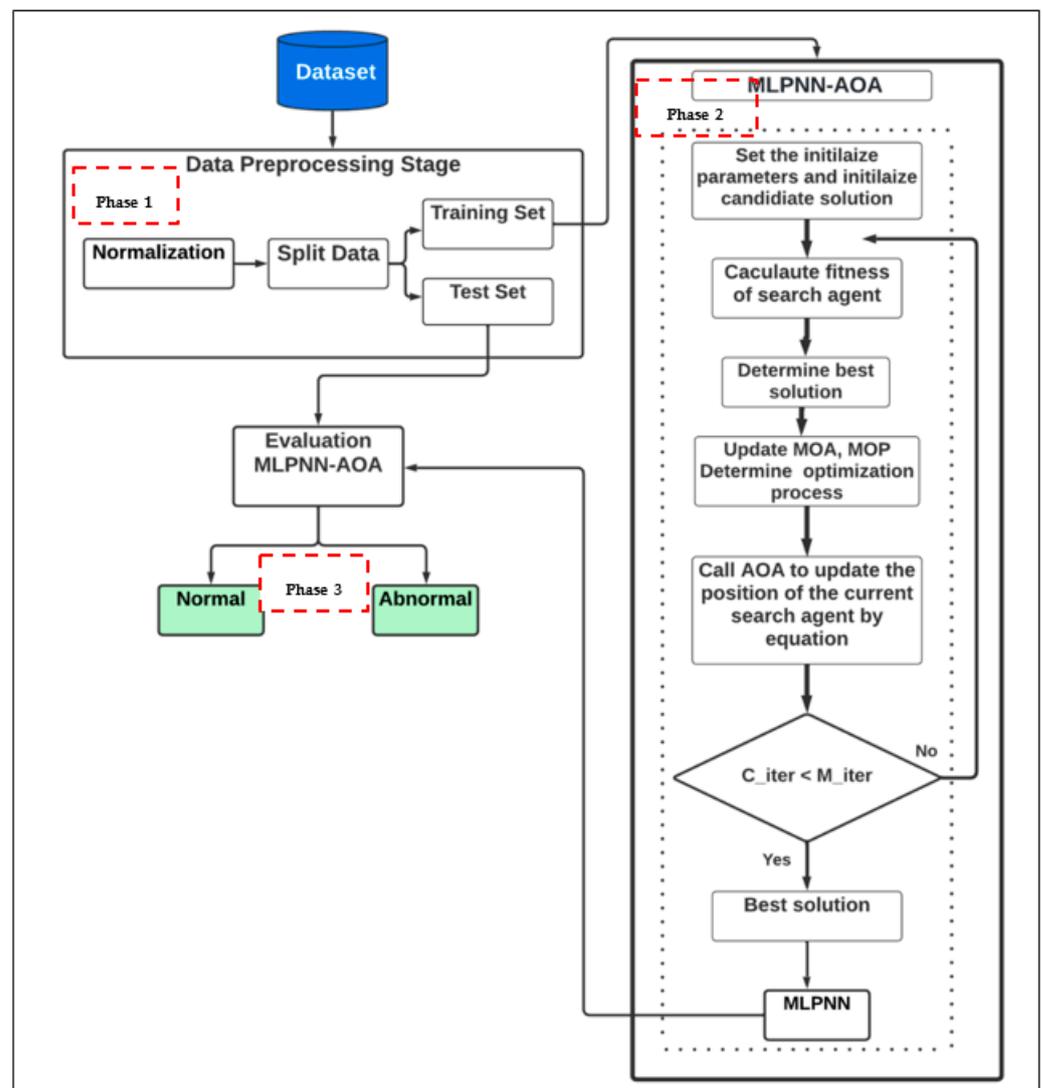


Figure 3. The MLPNN-AOA model implementation phases.

### 3.1. Phase 1: Data Preprocessing

A freely available dataset (namely, the Cleveland dataset [7,37]) is utilized in this research. It is an open-source dataset obtained from the UCI repository, holding 14 numeric features. The most important of these features is a class feature labeled as the ‘goal’, which refers to whether the patient has heart disease or not. We have chosen this dataset

solely because it was widely used in the literature and has been studied comprehensively. Also, other datasets in the same repository for the same heart disease prediction task (e.g., datasets from Hungary, Switzerland, Long Beach VA, and Statlog) are not implemented in our research work because they have missing values.

The dataset must be prepared to obtain good prediction accuracy by removing redundant and duplicate records. Furthermore, most ML algorithms only deal with numeric feature values. Issues with noise, missing values, and inconsistency are expected, particularly in the medical field. When operating with data of low quality, low-quality results are obtained. Usually, feature records have missing values [38]. Therefore, it is better to process non-numeric values to obtain many results. Therefore, the initial step in any ML approach is dataset preparation, for attain an appropriate format that is most valuable for the modeling stage [39]. The following is a review of the Cleveland dataset preparation steps that have been performed:

Step 1: Normalization is a data scaling method, which is the procedure for decreasing attribute values to a limited degree [40]. Usually, it is performed before FS and modeling stages according to different attribute scales, which confuse attribute comparison and impair the learning capability of the algorithms.

Step 2: Since the prediction model deals with only two classes, to improve the accuracy of the CVD prediction model, the classes (0,1,2,3,4) in the class label are transformed into only two: zero (if the original value is zero, then there is no CVD) and one (if the value is greater than or equal to one, then there is a CVD).

### 3.2. Phase 2: Data Reduction

FS is a data-reduction technique that involves selecting a subset of relevant features without changing feature dimensions to build a prediction model. The FS needs a search strategy and direction to select the sub-feature set, an objective function to evaluate the chosen features, termination condition, and outcome evaluation [41].

The main essence of optimization algorithms lies in finding new solutions with rules set that differ from one algorithm to another. These solutions are frequently evaluated to find the best one. These algorithms seek to find the best solution, as it has become important to not be satisfied with one process. The probability of reaching an optimal solution increases with the increase in the random number of these solutions and the number of iterations with substantial enough improvements [42].

Optimization processes are divided into two main phases: exploration and exploitation. The exploration phase aims to explore a wide range of research areas, using proxies to avoid local solutions. The exploitation phase aims to reach promising solutions close to improving their efficiency locally. The efficiency of the optimization algorithm requires an appropriate balance between the previous two stages.

In this research, the AOA was utilized to identify the most relevant features, where a subset of them was selected, consisting of twelve whose performance was greater than or equal to the performance offered by the other thirteen.

AOA is one of the optimization algorithm types which can solve optimization problems without counting their derivatives (meta-heuristic optimization population algorithms especially can achieve this). The exploration and exploitation stages are represented in this algorithm based on simple mathematical operations: (A "+"), (S "-"), (M "×"), and (D "/"). More details on the mechanism of the exploration and exploitation phases in the AOA can be found in [6].

Arithmetic is a fundamental part of number theory. It is one of the most significant parts of modern mathematics, along with algebra, geometry, and analysis. The traditional arithmetic measures used to study numbers are simple arithmetic operators (M, D, S, and A) [43].

The main inspiration for AOA stems from the use of the simple arithmetic operators above in solving arithmetic problems. To choose the best solution, AOA uses these factors

as mathematical optimization. The selected solutions are subject to specific criteria to be selected from a solution set.

The behavior and influence of arithmetic factors in AOA start by filtering several solutions that are generated randomly. The best solution obtained in each iteration is considered the best solution. First, the search stage must be determined as exploration or exploitation before the AOA starts working. Math optimizer accelerated (MOA) is a parameter used in the exploration and exploitation stages, where it utilizes the current iteration ranging from 1 to the maximum number of iterations as a termination condition of non-improvement criteria. The AOA employs the exploration and exploitation processes of a solution space using (D) and (M) operators, aiming to discover a semi-optimal solution, which can be deduced after many iterations. This also supports the second stage (exploitation) in improving the search process via enhanced communication between two search strategies.

The exploration stage explores the search area in several areas and uses methods to find the best solution using two arithmetic operations (D) and (M). The implementation of (D) or (M) is conditional on the MOA function and a random variable ( $r1$ ) that fulfills the condition  $r1 > MOA$ . As shown in Equation (1), the implementation of (D) (the first rule in the equation) is conditional on  $r2 < 0.5$ ;  $r2$  is a random variable; otherwise, (M) is executed.

$$x_{i,j}(C\_Iter + 1) = \begin{cases} best(x_j) \div MOP \times ((UB_j - LB_j) \times \mu + LB_j), & r2 < 0.5 \\ best(x_j) \times MOP \times ((UB_j - LB_j) \times \mu + LB_j), & otherwise \end{cases} \quad (1)$$

The parameter ( $C\_Iter$ ) refers to the position of a solution in the current iteration, which by default is considered the best solution found so far. On the other hand, the parameter  $C\_Iter + 1$  is the  $i$ th solution in the following iteration controlled by upper and lower bounds. Then, math optimizer probability (MOP) is implemented; this is the coefficient, where  $MOP(C\_Iter)$  is the value of the function at the current iteration  $C\_Iter$ , and  $M\_Iter$  is the maximum iterations number.

The exploitation stage is conducted using (S) or (A) operations which are meant to explore the search space, aiming at finding a near-optimal solution after a predetermined number of iterations. The operation of the exploitation stage is conditional on the value of the MOA function, where it must meet a condition ( $r1$ ) that is less than the value of the MOA ( $C\_Iter$ ). The implementation of (S) (first rule in the equation) is conditional on  $r3 < 0.5$ ; otherwise, (A) is executed. Producing a random number at each iteration, especially in the last iteration, sustains the exploration process by avoiding local optima stagnation. The estimation of the semi-optimal solution that is finally obtained can be randomly placed within a range that is determined by the positions of (D, M, S, and A) in the search range.

This can be summed up as follows: the AOA algorithm begins with random solutions. Factors (D, M, S, and A) estimate where solutions are in relation to an optimal solution. Then, each solution revamps its site to approach the best solution. The factor MOA will change its value from 0.2 to 0.9. Whenever the value of  $MOA < r1$ , it moves away from a near-optimal solution. If it is  $MOA > r1$ , then it approaches a near-optimal solution. Eventually, the AOA algorithm is stopped by reaching the criterion, as shown in Algorithm 1, pseudo-code of the AOA, in Algorithm 1 [6].

**Algorithm 1** Pseudo-code of the AOA algorithm (source: [6])

---

```

1: Initialize the Arithmetic Optimization Algorithm parameters  $\alpha, \mu$ 
2: Initialize the solutions' positions randomly (). (Solutions:  $i = 1, \dots, N$ )
3: while ( $C\_Iter < M\_Iter$ ) do
4:   Calculate the fitness function for the given solutions
5:   Find the best solution (Determined best so far).
6:   Update the MOA value.
7:   Update the MOP value.
8:   for ( $i = 1$  to solutions) do
9:     for ( $j = 1$  to positions) do
10:      Generate a random value between [0, 1] ( $r_1, r_2$ , and  $r_3$ )
11:      if  $r_1 > MOA$  then
12:        Exploration phase
13:        if  $r_2 > 0.5$  then
14:          (1) Apply the Division math operator ( $D'' \div ''$ ).
15:          Update the  $i$ th solutions' positions.
16:        else
17:          (2) Apply the Multiplication math operator ( $M'' \times ''$ ).
18:          Update the  $i$ th solutions' positions.
19:        end if
20:      else
21:        Exploitation phase
22:        if  $r_3 > 0.5$  then
23:          (1) Apply the Subtraction math operator ( $S'' - ''$ ).
24:          Update the  $i$ th solutions' positions.
25:        else
26:          (2) Apply the Addition math operator ( $A'' + ''$ ).
27:          Update the  $i$ th solutions' positions.
28:        end if
29:      end if
30:    end for
31:  end for
32:   $C\_Iter = C\_Iter + 1$ 
33: end while
34: Return the best solution

```

---

**3.3. Phase 3: Classification Task**

The MLPNN model is one an ANN. As information flows from one layer to the next layer, it is called a feed-forward model. MLPNN refers to networks consisting of multiple layers of cognition with threshold activation. MLPNN in its simplest form consists of three layers of nodes: the first layer is called the input layer, the intermediate layer is hidden, and the last layer is the output where the resulting output is obtained. Each layer consists of a specified number of nodes. Each node is a neuron that uses a nonlinear activation function except for the input nodes. Each node in each layer is connected to each node of the next and previous layer. The connections are called links or synapses. MLPNN is classified by the number of hidden layers, i.e., the number of all layers except the input and output layers. MLPNN uses a supervised learning method in training, which utilizes backpropagation. Backpropagation algorithms are widely used ML algorithms for training ANNs [44].

Backpropagation algorithms are used for calculating gradients; this is important in this model and in neural networks in general. The term is used to refer to the entire learning algorithm, including how gradients are used, such as random gradients. Backpropagation generalizes at the expense of delta-base gradation, which is the monolayer version of backpropagation; this, in turn, is generalized through auto-differentiation, where backpropagation is a special case of reverse accumulation (or "reverse mode") [45].

In MLPNN-AOA, this step relies on training, experimentation, and the comparison of algorithm parameters for improving the MLPNN's accuracy in predicting the probability of infection.

The MLPNN's set configuration parameters are as follows:

1. The hidden layer's number: four hidden layers with four neurons for each layer and two output units.
2. The biases and weights were first initialized randomly.
3. The maximum number of epochs is 500.
4. The activation function was set via a "set" method.

The MLPNN training method is implemented using the following algorithms: gradient descent, with (0.6, 0.05, and 500) for configurable learning rate (0.6), momentum (0.05), and size of batches, respectively.

#### 4. Results

MLPNN-AOA consists of AOA to select the most relevant features; then, MLPNN is used to predict the probability of CVD. The dataset used here is the Cleveland dataset; this was chosen because it is applied in many state-of-the-art approaches for predicting CVD. The dataset is split into two sets: training and testing (70% for training and 30% for testing). The training set is utilized to build the classifier and the testing set is utilized to evaluate it. The validation set is the same as the testing set. Many preliminary experiments are performed to obtain the best configurations that give the best results.

This section aims to explain the working environment that has been used for implementation, the criteria used to evaluate it, and the method of implementing it. A review of the obtained results is provided. Finally, the obtained results are compared with the results achieved using the MLPNN, DT, SVM, KNN, naïve Bayes, and RFC without FS.

##### 4.1. Experimental Setup

The metrics that are used in experiments have the same cessation conditions. There are nine main evaluation metrics that are utilized to estimate the proposed model.

*Accuracy*: This represents how the ML algorithm is accurate in classification or prediction. Accuracy is defined as the ratio of correctly predicted data to all data. It is defined mathematically as several data that the algorithm correctly classified as true or false, segmented using the sum of the data categorized as true or false. Equation (2) shows how to calculate it [46].

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2)$$

The area under the receiver operating characteristic (AROC) is a widely utilized statistic for evaluating the discriminative power of species distribution models. The area under the ROC curve is calculated by determining how accurate the quantitative diagnostic test is [47].

*Execution Time*: This is time taken by the prediction models to predict the probability of developing CVD. Equation (3) shows how to calculate it. See Figure 4.

$$ExecutionTime = (FinishingTime - StartingTime) \quad (3)$$

*Geometric mean (Geomean)*: This is the product of several series by the inverse of the total length of the series. The Geomean standard is most useable when the numbers tend to have large fluctuations or the numbers in the series are dependent. Equation (4) shows how to calculate it [48].

$$(1 + R_1) + (1 + R_2) + \dots + (1 + R_n)^{1/n} \quad (4)$$

where  $R_1 \dots R_n$  is the average of the observations and  $n$  is the total number of observations.

**F1-Score:** This represents a combination of precision and recall classifiers in one metric by taking the harmonic mean of them. It is often used to compare the results of two different classifiers. Equation (5) shows how to calculate it [49].

$$F1 - Score = 2 \times ((precision \times recall) / (precision + recall)) \quad (5)$$

**False Positive (FP):** This represents the number of negatively classified categories. In other words, it answers the following question: which categories are incorrectly predicted as being positive categories?

**False Negative (FN):** This represents the number of positively classified categories. In other words, it answers the following question: which categories are incorrectly predicted as being negative categories?

**Mean Square Error (MSE):** This is the mean squared difference between the evaluated subject and the evaluated values. MSE is the easiest and most expected loss function in ML. As illustrated in Equation (6), the MSE means the difference between the model's predictions and the ground truth, squares them, and modifies them across the entire dataset [50,51].

$$MSE = \frac{1}{n} \sum_{i=1}^n |X_i - X|^2 \quad (6)$$

**Precision:** This represents how the ML algorithm can determine how close the prediction results are to each other, regardless of whether those predictions are accurate or not. Equation (7) shows how to calculate it [46].

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

**Recall (sensitivity, also called true positive rate):** This represents the sick patient classification probability, which means the capability of a test to recognize those with the disease. Equation (8) shows how to calculate it [46].

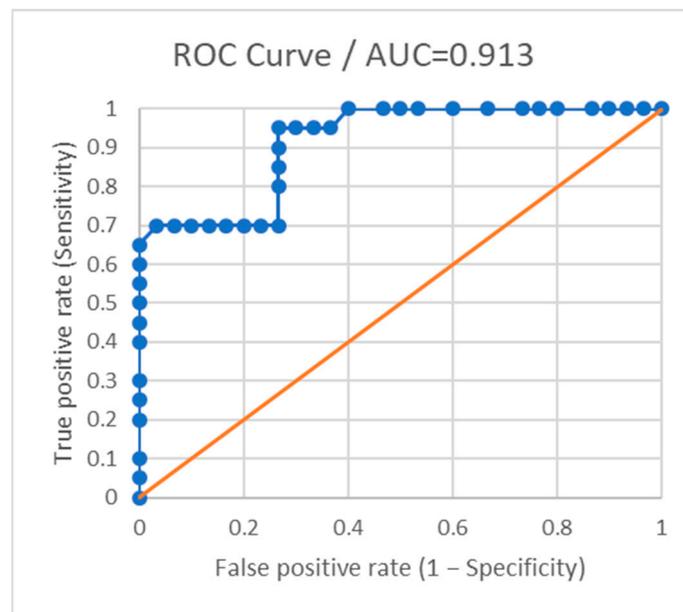
$$Recall = \frac{TP}{TP + FN} \quad (8)$$

**Specificity:** This is the ratio of the true negative that the model correctly predicts. Equation (9) shows how it is calculated [46].

$$Specificity = \frac{TN}{TN + FP} \quad (9)$$

**True Positive (TP):** This represents the number of positively classified categories. **True Negative (TN):** This represents the number of negatively classified categories.

MLPNN-AOA was implemented on a Lenovo workstation with Intel(R) processor Intel core-i5 4460 CPU 3.20 GHz; it has a 4 GB DDR3 RAM, Windows 8, a 64-bit operating system, and an x64-based processor. The program was written using MATLAB R2020a language. It is a powerful computational package that is dependent on a proprietary computational language that provides tools for users with a wide range of programming knowledge; it is utilized in different applications.



**Figure 4.** ROC curve for MLPNN.

#### 4.2. Testing and Analysis

In this research, two methods for diagnosing and predicting CVD are studied, analyzed, and compared. The first method deals with CVD prediction using MLPNN, DT, SVM, RFC, KNN, and naïve Bayes without FS. The second one uses AOA to select the most relevant features in the Cleveland dataset and then predicts using MLPNN. Below is a review of the results of the two techniques.

##### 4.2.1. CVD Prediction without Using FS

Several experiments have been conducted on MLPNN by random choice for the configuration of parameters, such as the number of neurons in each hidden layer, learning rate, number of epochs, and momentum alpha. Table 1 shows the performance metrics that are used to determine what the best MLPNN configuration parameters are. It can be concluded that the best configuration parameters are (4, 4, 0.6, 500, and 0.05) for the number of hidden layers, the number of neurons in each layer, the learning rate, the number of epochs, and the momentum alpha, respectively; these achieve (84.444%, 0.156, and 0.711) in terms of accuracy, MSE, and AROC, respectively.

The experimental results in the CVD prediction problems of MLPNN, SVM, DT, KNN, naïve Bayes, and RFC in terms of accuracy, MSE, recall, precision, F1-score, AROC, Geomean, and execution time are shown in Table 2. It can be concluded that SVM outperforms the other classifiers in all performance metrics. In detail, SVM achieves (81.1111%, 0.18889, and 0.822) in terms of accuracy, MSE, and AROC, respectively. MLPNN achieves (84.44%, 0.156, and 0.711). KNN achieves (61.11%, 0.39, and 0.6944). DT achieves (56.67%, 0.43, and 0.231). Naïve Bayes achieves (42.22%, 0.58, and 0.1). RFC achieves (0, 1, and 0.29).

**Table 1.** Performance comparison of MLPNN configuration parameters.

| NO | Epoch | Learning Rate | Momentum Alpha | Hidden Layers | Neurons | Accuracy (%) | MSE  | Recall | Precision | Specificity | F1-Score | Geomean | Execution Time (s) |
|----|-------|---------------|----------------|---------------|---------|--------------|------|--------|-----------|-------------|----------|---------|--------------------|
| 1  | 300   | 0.3           | 0.04           | 4             | 10      | 80           | 0.20 | 0.67   | 0.67      | 0.92        | 0.67     | 0.74    | 32                 |
| 2  | 500   | 0.4           | 0.02           | 4             | 10      | 80           | 0.20 | 0.70   | 0.70      | 0.89        | 0.70     | 0.74    | 52                 |
| 3  | 500   | 0.5           | 0.05           | 4             | 10      | 80           | 0.20 | 0.67   | 0.67      | 0.92        | 0.67     | 0.74    | 53                 |
| 4  | 500   | 0.4           | 0.02           | 3             | 10      | 77.8         | 0.22 | 0.61   | 0.61      | 0.94        | 0.61     | 0.71    | 49                 |
| 5  | 500   | 0.4           | 0.02           | 3             | 4       | 77.8         | 0.22 | 0.63   | 0.63      | 0.92        | 0.63     | 0.71    | 44                 |
| 6  | 500   | 0.4           | 0.05           | 3             | 10      | 75.6         | 0.24 | 0.70   | 0.70      | 0.80        | 0.70     | 0.69    | 23                 |
| 7  | 500   | 0.4           | 0.05           | 3             | 4       | 77.8         | 0.22 | 0.74   | 0.74      | 0.80        | 0.74     | 0.71    | 18                 |
| 8  | 5000  | 0.4           | 0.05           | 3             | 4       | 72.2         | 0.28 | 0.63   | 0.63      | 0.79        | 0.63     | 0.65    | 183                |
| 9  | 500   | 0.3           | 0.05           | 3             | 4       | 77.8         | 0.22 | 0.68   | 0.68      | 0.85        | 0.68     | 0.71    | 23                 |
| 10 | 1000  | 0.4           | 0.05           | 4             | 10      | 80           | 0.20 | 0.72   | 0.72      | 0.87        | 0.72     | 0.74    | 59                 |
| 11 | 500   | 0.4           | 0.05           | 4             | 4       | 81.1         | 0.19 | 0.90   | 0.90      | 0.75        | 0.90     | 0.76    | 43                 |
| 12 | 500   | 0.6           | 0.05           | 4             | 4       | 84.4         | 0.16 | 0.71   | 0.71      | 0.98        | 0.71     | 0.80    | 24                 |
| 13 | 500   | 1             | 0.05           | 3             | 4       | 78.9         | 0.21 | 0.83   | 0.83      | 0.74        | 0.83     | 0.73    | 18                 |
| 14 | 1000  | 1             | 0.05           | 3             | 4       | 71.1         | 0.29 | 0.75   | 0.75      | 0.67        | 0.75     | 0.64    | 37                 |
| 15 | 5000  | 0.4           | 0.05           | 3             | 10      | 72.2         | 0.28 | 0.80   | 0.80      | 0.64        | 0.80     | 0.65    | 23                 |
| 16 | 500   | 0.4           | 0.05           | 5             | 10      | 41.1         | 0.59 | 1.00   | 1.00      | 0.02        | 1.00     | 0.33    | 32                 |
| 17 | 500   | 0.2           | 0.05           | 3             | 4       | 75.6         | 0.24 | 0.76   | 0.76      | 0.76        | 0.76     | 0.69    | 18                 |
| 18 | 500   | 0.3           | 0.05           | 4             | 10      | 71.1         | 0.29 | 0.60   | 0.60      | 0.79        | 0.60     | 0.64    | 28.1               |
| 19 | 500   | 0.2           | 0.05           | 3             | 4       | 78.9         | 0.21 | 0.60   | 0.60      | 0.96        | 0.60     | 0.73    | 19.0               |
| 20 | 500   | 0.7           | 0.05           | 4             | 10      | 74.4         | 0.26 | 0.74   | 0.74      | 0.75        | 0.74     | 0.68    | 27.8               |

**Table 2.** Comparison of classifiers (all features).

| Classifier  | Accuracy (%) | MSE   | Recall | Precision | Specificity | F1-Score | AROC  | Geomean | Execution Time (s) |
|-------------|--------------|-------|--------|-----------|-------------|----------|-------|---------|--------------------|
| DT          | 56.6         | 0.433 | 0.231  | 0.231     | 0.824       | 0.231    | 0.231 | 0.481   | 0.336              |
| SVM         | 81.1         | 0.189 | 0.822  | 0.822     | 0.800       | 0.822    | 0.822 | 0.755   | 0.909              |
| MLPNN       | 84.4         | 0.156 | 0.711  | 0.711     | 0.978       | 0.711    | 0.711 | 0.796   | 24                 |
| RFC         | 0            | 1     | 0      | 0         | 0           | 0        | 0     | 0       | 3                  |
| KNN         | 61.1         | 0.39  | 0      | NaN       | 1           | NaN      | 0     | 0.527   | 0.614              |
| Naïve Bayes | 42.2         | 0.58  | 0.1    | 0.2       | 0.68        | 0.133    | 0.1   | 0.340   | 0.078              |

#### 4.2.2. CVD Prediction Using MLPNN-AOA

Several experiments were performed on MLPNN-AOA to choose the best function, the number of solutions, the iterations, the lower and upper bounds, and the dimensions. Table 3 shows the best-obtained solutions. As shown in Table 3, the best AOA functions that outperformed the others are (F8, F11, F13, F20, F21, F22, and F24). The best function of them is that F20 achieves (88.890%, 0.110, and 0.840) for accuracy, MSE, and AROC, respectively; it achieves (two and ten) for the iteration number and the number of solutions, [0, 1] for the upper and lower bounds, and thirteen for the dimensions.

**Table 3.** Performance comparison of MLPNN-AOA configuration parameters (70%–30%) experiment.

| Solution No | F_obj | M_Iter | NO Neuron | LB   | UB  | Dim | Epochs | SF | Accuracy (%) | MSE  | Recall | Precision | Specificity | F1-Score | Geomean | Execution Time (s) |
|-------------|-------|--------|-----------|------|-----|-----|--------|----|--------------|------|--------|-----------|-------------|----------|---------|--------------------|
| 20          | 20    | 50     | 4         | −4   | 1   | 13  | 500    | 7  | 81.1         | 0.2  | 0.80   | 0.85      | 0.83        | 0.82     | 0.75    | 25                 |
| 20          | 24    | 50     | 4         | −4   | 1   | 13  | 500    | 9  | 83.3         | 0.2  | 0.79   | 0.85      | 0.87        | 0.82     | 0.78    | 25                 |
| 5           | 24    | 10     | 4         | −4   | 1   | 13  | 2000   | 7  | 82.2         | 0.2  | 0.86   | 0.79      | 0.79        | 0.82     | 0.768   | 100                |
| 20          | 21    | 50     | 4         | −4   | 1   | 13  | 2000   | 8  | 81.1         | 0.2  | 0.73   | 0.86      | 0.89        | 0.79     | 0.755   | 107                |
| 20          | 22    | 50     | 10        | −4   | 1   | 13  | 2000   | 10 | 82.2         | 0.2  | 0.71   | 0.88      | 0.92        | 0.78     | 0.768   | 133                |
| 20          | 24    | 50     | 10        | −500 | 500 | 13  | 5000   | 10 | 80.0         | 0.2  | 0.68   | 0.77      | 0.88        | 0.72     | 0.741   | 230                |
| 2           | 24    | 10     | 4         | −500 | 500 | 13  | 5000   | 9  | 80.0         | 0.20 | 0.74   | 0.78      | 0.84        | 0.76     | 0.741   | 235                |
| 2           | 8     | 10     | 4         | −4   | 1   | 13  | 500    | 8  | 83.3         | 0.2  | 0.80   | 0.78      | 0.85        | 0.79     | 0.782   | 28                 |
| 2           | 8     | 10     | 4         | −4   | 1   | 13  | 500    | 8  | 85.6         | 0.1  | 0.84   | 0.86      | 0.87        | 0.85     | 0.809   | 33                 |
| 2           | 11    | 10     | 4         | −4   | 1   | 13  | 500    | 6  | 83.3         | 0.2  | 0.84   | 0.78      | 0.83        | 0.81     | 0.782   | 22                 |
| 2           | 13    | 10     | 4         | −4   | 1   | 13  | 500    | 11 | 84.4         | 0.2  | 0.84   | 0.80      | 0.85        | 0.82     | 0.796   | 35                 |
| 2           | 20    | 10     | 4         | 0    | 1   | 13  | 500    | 12 | 88.9         | 0.1  | 0.84   | 0.89      | 0.92        | 0.86     | 0.852   | 27                 |
| 2           | 20    | 7      | 4         | 0    | 1   | 13  | 500    | 11 | 85.6         | 0.1  | 0.85   | 0.83      | 0.86        | 0.84     | 0.809   | 26                 |
| 2           | 20    | 5      | 4         | 0    | 1   | 12  | 500    | 12 | 85.6         | 0.1  | 0.79   | 0.93      | 0.93        | 0.85     | 0.809   | 27                 |

To guarantee the usefulness of MLPNN-AOA, it was tested by splitting the dataset into 80% for training and 20% for testing, in addition to a 10-fold cross-validation. As shown in Table 4, the best AOA functions that outperformed the others are (F11, F20, and F24). The best one is F20, which achieves (86.67%, 0.1333, and 0.85) for accuracy, MSE, and AROC, respectively; it achieves (twenty and ten) for the iteration number and the number of solutions,  $[-100, 100]$  for the upper and lower bounds, and thirteen for the dimensions. Table 5 shows the experimental results for the 10-fold cross-validation. The best AOA function is F20, which achieves (60.00%, 0.40, and 0.47) for accuracy, MSE, and AROC, respectively; it achieves (ten and two) for the iteration number and the number of solutions,  $[0, 1]$  for the upper and lower bounds, and thirteen for the dimensions. From Tables 1 and 2, F20 selects the highest feature number, equal to twelve, which means that all features in the Cleveland dataset positively affect CVD predictions, excluding feature number 10; this represents exercise-induced ST depression, compared to the rest, which do not (see Figure 5). Therefore, it can be concluded that the first question of the research has been answered, and the first objective has been achieved.

Figure 6 shows the experimental results in the CVD prediction problems of MLPNN-AOA, SVM, MLPNN, DT, KNN, naïve Bayes, and RFC, in terms of accuracy, MSE, AROC, recall, precision, F1-score, and Geomean. MLPNN-AOA exceeds the other classifiers in all performance metrics. In detail, SVM achieves (81.11%, 0.18, and 0.822), MLPNN achieves (84.444%, 0.156, and 0.711), KNN achieves (61.11%, 0.39, and 0.6944), DT achieves (56.67%, 0.433, and 0.231), naïve Bayes achieves (42.22%, 0.58, and 0.1), RFC achieves (0, 1, and 0.29), and MLPNN-AOA achieves (88.890%, 0.110, and 0.840) in terms of accuracy, MSE, and AROC. Therefore, it can be concluded that the second objective of this research has been achieved.

**Table 4.** Performance comparison of MLPNN-AOA configuration parameters (80%–20%) experiment.

| Solution No | F_obj | M_Iter | NO Neuron | LB   | UB  | Dim | Epochs | SF | Accuracy (%) | MSE  | Recall | Precision | Specificity | F1-Score | Geomean | Execution Time (s) |
|-------------|-------|--------|-----------|------|-----|-----|--------|----|--------------|------|--------|-----------|-------------|----------|---------|--------------------|
| 20          | 20    | 50     | 4         | 0    | 1   | 13  | 500    | 10 | 81.7         | 0.18 | 0.80   | 0.77      | 0.83        | 0.78     | 0.761   | 25                 |
| 20          | 24    | 50     | 4         | −4   | 1   | 13  | 500    | 7  | 85.0         | 0.2  | 0.68   | 0.94      | 0.97        | 0.79     | 0.803   | 25                 |
| 5           | 24    | 10     | 4         | −4   | 1   | 13  | 2000   | 9  | 73.3         | 0.3  | 0.67   | 0.67      | 0.78        | 0.67     | 0.662   | 103                |
| 20          | 21    | 50     | 4         | −4   | 1   | 13  | 2000   | 10 | 81.7         | 0.9  | 0.64   | 0.95      | 0.97        | 0.77     | 0.761   | 104                |
| 20          | 23    | 50     | 10        | 0    | 10  | 13  | 2000   | 12 | 80.0         | 0.2  | 0.71   | 0.71      | 0.85        | 0.71     | 0.569   | 143                |
| 2           | 8     | 10     | 10        | −4   | 1   | 13  | 500    | 7  | 71.7         | 0.3  | 0.62   | 0.75      | 0.81        | 0.68     | 0.741   | 34                 |
| 2           | 10    | 10     | 10        | −4   | 1   | 13  | 500    | 8  | 73.3         | 0.3  | 0.74   | 0.63      | 0.73        | 0.68     | 0.682   | 34                 |
| 2           | 11    | 10     | 10        | −4   | 1   | 13  | 500    | 8  | 85.0         | 0.2  | 0.88   | 0.79      | 0.82        | 0.84     | 0.643   | 34                 |
| 2           | 11    | 10     | 10        | −600 | 600 | 13  | 500    | 7  | 65.0         | 0.4  | 0.59   | 0.74      | 0.73        | 0.66     | 0.662   | 34                 |
| 2           | 12    | 10     | 4         | −50  | 50  | 13  | 500    | 9  | 63.3         | 0.4  | 0.67   | 0.63      | 0.60        | 0.65     | 0.803   | 28                 |
| 2           | 20    | 10     | 4         | 0    | 1   | 13  | 500    | 12 | 81.7         | 0.2  | 0.88   | 0.73      | 0.77        | 0.80     | 0.569   | 26                 |
| 2           | 20    | 7      | 4         | 0    | 1   | 13  | 500    | 12 | 83.3         | 0.2  | 0.81   | 0.86      | 0.86        | 0.83     | 0.551   | 26                 |
| 2           | 20    | 5      | 10        | 0    | 1   | 12  | 500    | 12 | 78.3         | 0.22 | 0.75   | 0.78      | 0.81        | 0.76     | 0.761   | 33                 |
| 20          | 2     | 50     | 10        | −50  | 50  | 10  | 500    | 5  | 76.7         | 0.23 | 0.74   | 0.61      | 0.78        | 0.67     | 0.782   | 26                 |
| 10          | 20    | 20     | 10        | −100 | 100 | 13  | 500    | 9  | 86.7         | 0.13 | 0.85   | 0.85      | 0.88        | 0.85     | 0.721   | 25                 |
| 20          | 4     | 50     | 10        | −100 | 100 | 13  | 500    | 5  | 75.0         | 0.3  | 0.70   | 0.73      | 0.79        | 0.72     | 0.701   | 25                 |

**Table 5.** Performance comparison of MLPNN-AOA cross-validation configuration parameters.

| Solution No | F_obj | M_Iter | NO Neuron | LB   | UB  | Dim | Epochs | SF | Accuracy (%) | MSE    | Recall | Precision | Specificity | F1-Score | Geomean | Execution Time (S) |
|-------------|-------|--------|-----------|------|-----|-----|--------|----|--------------|--------|--------|-----------|-------------|----------|---------|--------------------|
| 20          | 20    | 50     | 4         | −4   | 1   | 13  | 500    | 6  | 60.0         | 0.40   | 0.47   | 0.73      | 0.77        | 0.57     | 0.516   | 35                 |
| 20          | 24    | 50     | 4         | −4   | 1   | 13  | 500    | 6  | 53.3         | 0.467  | 0.35   | 0.67      | 0.77        | 0.46     | 0.447   | 36                 |
| 5           | 24    | 10     | 4         | −4   | 1   | 13  | 2000   | 8  | 53.3         | 0.467  | 0.35   | 0.67      | 0.77        | 0.46     | 0.447   | 87                 |
| 20          | 21    | 50     | 4         | −4   | 1   | 13  | 2000   | 8  | 50.0         | 0.50   | 0.33   | 0.50      | 0.67        | 0.40     | 0.414   | 149                |
| 20          | 22    | 50     | 10        | 0    | 10  | 13  | 2000   | 11 | 53.3         | 0.467  | 0.35   | 0.67      | 0.77        | 0.46     | 0.447   | 142                |
| 20          | 23    | 50     | 10        | 0    | 10  | 13  | 2000   | 13 | 56.7         | 0.433  | 0.41   | 0.70      | 0.77        | 0.52     | 0.481   | 148                |
| 2           | 24    | 10     | 10        | −500 | 500 | 13  | 500    | 8  | 60.0         | 0.40   | 0.43   | 0.60      | 0.75        | 0.50     | 0.516   | 32                 |
| 2           | 8     | 10     | 10        | −4   | 1   | 13  | 500    | 13 | 43.3         | 0.567  | 0.00   | NaN       | 1.00        | NaN      | 0.350   | 32                 |
| 2           | 10    | 10     | 4         | −4   | 1   | 13  | 500    | 9  | 53.3         | 0.467  | 0.35   | 0.67      | 0.77        | 0.46     | 0.447   | 22                 |
| 2           | 11    | 10     | 4         | −4   | 1   | 13  | 500    | 9  | 53.3         | 0.4667 | 0.35   | 0.67      | 0.77        | 0.46     | 0.447   | 22                 |
| 2           | 11    | 10     | 4         | −600 | 600 | 13  | 500    | 8  | 56.7         | 0.433  | 0.41   | 0.70      | 0.77        | 0.52     | 0.481   | 22                 |
| 2           | 12    | 10     | 4         | −50  | 50  | 13  | 500    | 10 | 53.3         | 0.467  | 0.35   | 0.67      | 0.77        | 0.46     | 0.447   | 23                 |
| 2           | 20    | 7      | 4         | 0    | 1   | 13  | 500    | 13 | 56.7         | 0.433  | 0.47   | 0.67      | 0.69        | 0.55     | 0.481   | 24.1               |
| 2           | 20    | 10     | 4         | 0    | 1   | 13  | 500    | 10 | 60.0         | 0.40   | 0.47   | 0.73      | 0.77        | 0.57     | 0.516   | 23.16              |
| 2           | 20    | 5      | 10        | 0    | 1   | 12  | 500    | 12 | 53.3         | 0.467  | 0.35   | 0.67      | 0.77        | 0.46     | 0.447   | 32.12              |

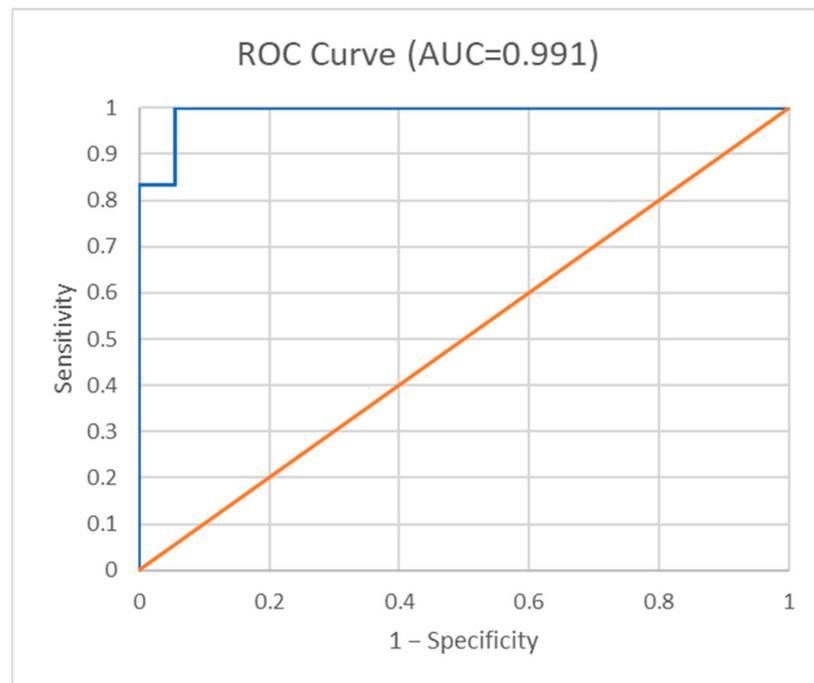


Figure 5. ROC curve for MLPNN-AOA.

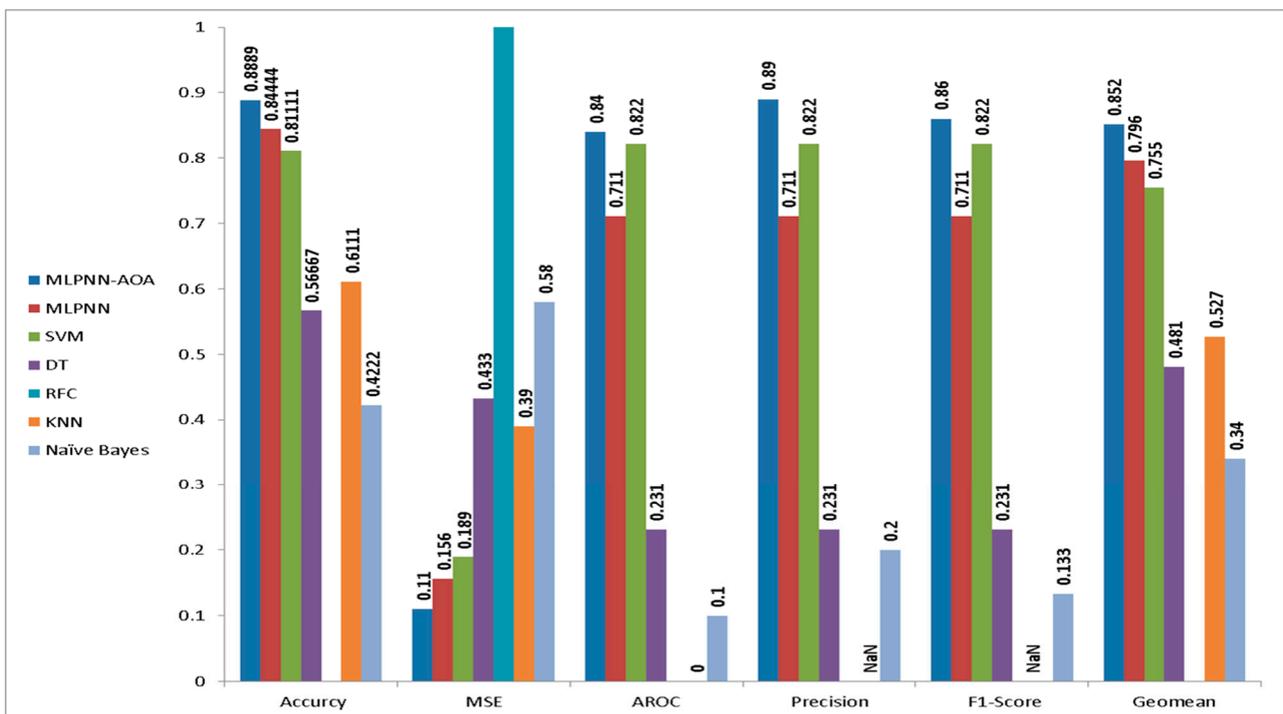


Figure 6. Comparison of classifier performances.

#### 4.2.3. Comparison of MLPNN-AOA with MLPNN

Based on the evaluation metrics improvement percentage, Table 6 shows the comparison between MLPNN-AOA and MLPNN in terms of accuracy, average MSE, AROC, F1-score, and Geomean.

**Table 6.** Improvement ratio for MLPNN-AOA.

| Method          | Accuracy (%) | MSE   | AROC  | F1-Score | Geomean |
|-----------------|--------------|-------|-------|----------|---------|
| MLPNN           | 84.444       | 0.156 | 0.711 | 0.711    | 0.796   |
| MLPNN-AOA       | 88.89        | 0.11  | 0.84  | 0.860    | 0.852   |
| Improvement (%) | 5.26         | 29.48 | 18.14 | 20.95    | 7.03    |

In terms of accuracy, it can be seen that the MLPNN-AOA model surpasses the MLPNN model. In detail, MLPNN-AOA reaches 88.890% when the number of epochs is (500); meanwhile, MLPNN achieves 84.444%. Thus, it can be inferred that the accuracy increases when the most relevant features are chosen by AOA. Therefore, the second question of the research has been answered. On the other hand, in terms of the average MSE, it can be seen that MLPNN-AOA exceeds the MLPNN. In detail, MLPNN-AOA reaches 0.11 when the number of epochs is (500); meanwhile, MLPNN achieves 0.156. Hence, it can be inferred that the MSE reduces when AOA chooses the most relevant features in the Cleveland dataset. In addition, in terms of AROC, it can be seen that MLPNN-AOA exceeds the MLPNN. In detail, MLPNN-AOA reaches 0.840 when the number of epochs is (500); meanwhile, MLPNN achieves 0.711. Also, in terms of Geomean, it can be seen that MLPNN-AOA exceeds the MLPNN. In detail, MLPNN-AOA reaches 0.852 when the number of epochs is (500); meanwhile, MLPNN achieves 0.796. The same goes with the F1-score as well.

It can be concluded that MLPNN-AOA significantly improved the performance of MLPNN-based FS. Table 7 shows statistical significance at the level of 0.0001, a confidence interval of 5.158, and the degrees of freedom of MLPNN-AOA and MLPNN. The results proved that MLPNN-AOA is statistically feasible.

**Table 7.** T-test for MLPNN-AOA and MLPNN.

| Method    | Mean   | Standard Deviation | Standard Error Mean |
|-----------|--------|--------------------|---------------------|
| MLPNN-AOA | 83.332 | 2.504              | 0.669               |
| MLPNN     | 78.174 | 3.448              | 0.921               |

#### 4.2.4. Comparison of MLPNN-AOA with Other State-of-the-Art Models

In this subsection, the experimental results for MLPNN-AOA are compared with other FS approaches, such as correlation-based feature selection (CFS), relief, filtered subset, PSO, info gain, chi-squared, consistency subset, filtered attribute, one-attribute-based approach, GA, and gain ratio. Table 8 shows the comparison with some state-of-the-art models in terms of the number of FS approaches and the prediction accuracy of MLPNN after they selected the most relevant features. It is concluded that MLPNN-AOA is superior to other models in terms of prediction accuracy on a Cleveland dataset with twelve features; it is noted that all the other optimization algorithms selected feature 10 except for AOA. So, the third objective of this research is achieved.

As shown in Table 8, the performance of MLPNN-AOA in terms of accuracy is compared with some previously proposed prediction models using FS before predicting CVD using MLPNN on the Cleveland dataset. It can be concluded that the accuracy of the MLPNN-AOA model outperformed that of all other models.

The research outcomes confirmed that MLPNN-AOA surpassed the SVM, MLPNN, DT, KNN, naïve Bayes, and RFC in terms of accuracy, MSE, and AROC. Further, it outperforms other models based on FS, such as PSO-MLP. AOA has shown its ability to decrease the number of features and select the best ones in the Cleveland dataset, where it ignores the irrelevant feature number ten. So, MLPNN-AOA facilitates learning the dataset by reducing the total feature number; as a result, it eliminates the problem of overfitting and underfitting. Hence, the third question of the research has been answered, and the fourth objective has been achieved. Further, it can be concluded that the objectives of the research have been achieved.

**Table 8.** Comparison with the FS algorithms.

| No | FS Algorithms                 | No. of FS | FS  | Accuracy (%) |
|----|-------------------------------|-----------|---|--------------|
| 1  | Relief                        | 13        | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 | 78.14        |
| 2  | Info gain                     | 13        | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 | 80.37        |
| 3  | Chi squared                   | 13        | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 | 80.37        |
| 4  | Filtered subset               | 6         | 3, 8, 9, 10, 12, 13                       | 78.88        |
| 5  | One-attribute-based algorithm | 13        | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 | 79.25        |
| 6  | Consistency based             | 10        | 1, 2, 3, 7, 8, 9, 10, 11, 12, 13          | 78.14        |
| 7  | Gain ratio                    | 13        | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 | 78.88        |
| 8  | Filtered attribute            | 13        | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 | 80.37        |
| 9  | CFS                           | 8         | 3, 7, 8, 9, 10, 11, 12, 13                | 82.22        |
| 10 | GA                            | 6         | 3, 7, 8, 9, 10, 13                        | 79.81        |
| 11 | PSO                           | 6         | 3, 7, 8, 9, 10, 13                        | 80.54        |
| 12 | MLPNN-AOA                     | 12        | 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13     | 88.89        |

Despite the noticeable improvement of MLPNN-AOA over MLPNN in terms of MSE, AROC, and F1-score, the improvement percentage was slight in accuracy. Also, MLPNN-AOA selected twelve features, excluding only feature number ten from the Cleveland dataset; meanwhile, other state-of-the-art models such as the filtered subset select six, which reduces the computational complexity of the dataset and reduces the problems of overfitting and underfitting.

## 5. Conclusions

In conclusion, the development of an intelligent system for accurately diagnosing cardiovascular diseases (CVDs) represents a crucial advancement in the healthcare field. In this research, we proposed an automated ML model for various kinds of CVD prediction and classification. Our prediction model consists of multiple steps. Firstly, a benchmark dataset is preprocessed using filter techniques. Secondly, the novel AOA is implemented as a feature selection technique to select the best subset of features that influence the accuracy of the prediction model. Thirdly, the classification task is implemented using a multilayer perceptron neural network to classify the instances of the dataset into two class labels: determining whether a CVD is present or not. The AOA is used as an optimization algorithm. It is one of the robust FS algorithms utilized in various fields, where it selects the most relevant features that can improve accuracy and overall performance measurements. The limitation of the proposed model in this research is generally shown in the exact number of features needed for the CVD prediction model to significantly increase. It depends on how it predicts undefined classes efficiently in terms of accuracy, MSE, precision, recall, F1-score, AROC, Geomean, execution time, and the total number of selected features. The Cleveland dataset has been utilized in training and testing MLPNN-AOA.

The results of the two methods used in this research have been compared. The first is the prediction of CVD without FS and the second is MLPNN-AOA. The results demonstrate that MLPNN-AOA outperformed the other six classifiers on all performance measures. Moreover, the results show an improvement between MLPNN-AOA and MLPNN without FS; in contrast, MLPNN-AOA improved MLPNN by 14.74% in accuracy, 48% in MSE, and 1.9% in AROC. Moreover, the prediction accuracy of MLPNN-AOA was compared with other prediction models proposed in previous studies that use FS methods in terms of prediction accuracy. The results showed the superiority of MLPNN-AOA over the other models by selecting 12 features, excluding feature number 10, which was selected by most of the other models. In future work, hybridizing MLPNN with any other optimization algorithm may be proposed to choose efficient and unexplored features that improve the significance of this research and develop efficient prediction models for CVD problems. Other deep learning methods can also be utilized instead of MLPNN, such as CNNs.

**Author Contributions:** Conceptualization, M.K.A. and G.J.; methodology, H.A. (Haitham Almanaseer); software, F.A.A.; validation, M.K.A., G.J. and A.J.; formal analysis, A.S.A.; investigation, J.A.; resources, S.J.; data curation, H.A. (Hayat Alfagham); writing—original draft preparation, H.A. (Haitham Almanaseer); writing—review and editing, G.J.; visualization, M.K.A.; supervision, G.J.; project administration, M.K.A.; funding acquisition, F.A.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The deanship of Scientific Research, Imam Abdulrahman Bin Faisal University supported this work, grant numbers 2019-416-ASCS.

**Data Availability Statement:** The dataset can be found at <https://archive.ics.uci.edu/ml/datasets/heart+Disease>. Accessed on 1 June 2023.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

- Munsif, M.; Khan, H.; Khan, Z.A.; Hussain, A.; Ullah, F.U.; Lee, M.Y.; Baik, S.W. PV-ANet: Attention-Based Network for Short-term Photovoltaic Power Forecasting. In Proceedings of the 8th International Conference on Next Generation Computing, Jeju, Republic of Korea, 6–8 October 2022; pp. 133–135.
- Khan, H.; Haq, I.U.; Munsif, M.; Mustaqeem; Khan, S.U.; Lee, M.Y. Automated Wheat Diseases Classification Framework Using Advanced Machine Learning Technique. *Agriculture* **2022**, *12*, 1226. [[CrossRef](#)]
- Vijayashree, J.; Sultana, H.P. A machine learning framework for feature selection in heart disease classification using improved particle swarm optimization with support vector machine classifier. *Program. Comput. Softw.* **2018**, *44*, 388–397. [[CrossRef](#)]
- Prakash, V.J.; Karthikeyan, N.K. Enhanced Evolutionary Feature Selection and Ensemble Method for Cardiovascular Disease Prediction. *Interdiscip. Sci. Comput. Life Sci.* **2021**, *13*, 389–412. [[CrossRef](#)] [[PubMed](#)]
- Haq, A.U.; Li, J.P.; Memon, M.H.; Nazir, S.; Sun, R. A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms. *Mob. Inf. Syst.* **2018**, *2018*, 3860146. [[CrossRef](#)]
- Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [[CrossRef](#)]
- Detrano, R.; Janosi, A.; Steinbrunn, W.; Pfisterer, M.; Schmid, J.; Sandhu, S.; Guppy, K.; Lee, S.; Froelicher, V. International application of a new probability algorithm for the diagnosis of coronary artery disease. *Am. J. Cardiol.* **1989**, *64*, 304–310. [[CrossRef](#)] [[PubMed](#)]
- Tash, A.A.; Al-Bawardy, R.F. Cardiovascular Disease in Saudi Arabia: Facts and the Way Forward. *J. Saudi Heart Assoc.* **2023**, *35*, 148–162. [[CrossRef](#)] [[PubMed](#)] [[PubMed Central](#)]
- Xu, T.; Gao, Z.; Zhuang, Y. Fault Prediction of Control Clusters Based on an Improved Arithmetic Optimization Algorithm and BP Neural Network. *Mathematics* **2023**, *11*, 2891. [[CrossRef](#)]
- Dritsas, E.; Trigka, M. Efficient Data-Driven Machine Learning Models for Cardiovascular Diseases Risk Prediction. *Sensors* **2023**, *23*, 1161. [[CrossRef](#)]
- Dweekat, O.Y.; Lam, S.S. Cervical Cancer Diagnosis Using an Integrated System of Principal Component Analysis, Genetic Algorithm, and Multilayer Perceptron. *Healthcare* **2022**, *10*, 2002. [[CrossRef](#)]
- Zafar, A.; Hussain, S.J.; Ali, M.U.; Lee, S.W. Metaheuristic Optimization-Based Feature Selection for Imagery and Arithmetic Tasks: An fNIRS Study. *Sensors* **2023**, *23*, 3714. [[CrossRef](#)] [[PubMed](#)]
- Al-Dulaimi, K.; Banks, J.; Al-Sabaawi, A.; Nguyen, K.; Chandran, V.; Tomeo-Reyes, I. Classification of HEp-2 Staining Pattern Images Using Adapted Multilayer Perceptron Neural Network-Based Intra-Class Variation of Cell Shape. *Sensors* **2023**, *23*, 2195. [[CrossRef](#)] [[PubMed](#)]
- Raj, P.; Evangeline, P. *The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases*; Academic Press: Cambridge, MA, USA, 2020.
- Desuky, A.S.; Hussain, S.; Kausar, S.; Islam, M.A.; El Bakrawy, L.M. EAOA: An Enhanced Archimedes Optimization Algorithm for Feature Selection in Classification. *IEEE Access* **2021**, *9*, 120795–120814. [[CrossRef](#)]
- Janosi, A.; Steinbrunn, W.; Pfisterer, M.; Detrano, R. Heart Disease Dataset—UCI Machine Learning Repository. Center for Machine Learning and Intelligent Systems. 2021. Available online: <https://archive.ics.uci.edu/ml/datasets/heart+Disease> (accessed on 21 October 2021).
- Géron, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.
- Douglass, M.J. *Book Review: Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*, 2nd ed.; Aurélien, G., Ed.; 1005 Gravenstein Highway North; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2020.
- Alrajeh, N.A.; Khan, S.; Shams, B. Intrusion detection systems in wireless sensor networks: A review. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 167575. [[CrossRef](#)]
- Hichem, H.; Elkamel, M.; Rafik, M.; Mesaaoud, M.T.; Ouahiba, C. A new binary grasshopper optimization algorithm for feature selection problem. *J. King Saud Univ-Comput. Inf. Sci.* **2019**, *34*, 316–328. [[CrossRef](#)]

21. Alweshah, M.; Khalaileh, S.A.; Gupta, B.B.; Almomani, A.; Hammouri, A.I.; Al-Betar, M.A. The monarch butterfly optimization algorithm for solving feature selection problems. *Neural Comput. Appl.* **2020**, *34*, 11267–11281. [[CrossRef](#)]
22. Chen, C.W.; Tsai, Y.H.; Chang, F.R.; Lin, W.C. Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. *Expert Syst.* **2020**, *37*, e12553. [[CrossRef](#)]
23. Parthiban, R.; Usharani, S.; Saravanan, D.; Jayakumar, D.; Palani, D.U.; StalinDavid, D.D.; Raghuraman, D. Prognosis of chronic kidney disease (CKD) using hybrid filter wrapper embedded feature selection method. *Eur. J. Mol. Clin. Med.* **2021**, *7*, 2511–2530.
24. Kim JO, R.; Jeong, Y.S.; Kim, J.H.; Lee, J.W.; Park, D.; Kim, H.S. Machine Learning-Based Cardiovascular Disease Prediction Model: A Cohort Study on the Korean National Health Insurance Service Health Screening Database. *Diagnostics* **2021**, *11*, 943. [[CrossRef](#)]
25. Dutta, A.; Batabyal, T.; Basu, M.; Acton, S.T. An efficient convolutional neural network for coronary heart disease prediction. *Expert Syst. Appl.* **2020**, *159*, 113408. [[CrossRef](#)]
26. Gadde, H. Heart Disease Predictions Using Machine Learning Algorithms and Ensemble Learning. *Int. J. Eng. Trends Appl.* **2020**, *7*, 4.
27. Kumar, M.N.; Koushik KV, S.; Deepak, K. Prediction of heart diseases using data mining and machine learning algorithms and tools. *International Journal of Scientific Research in Computer Science. Eng. Inf. Technol.* **2018**, *3*, 887–898.
28. Zaffar, M.; Hashmani, M.A.; Savita, K.S.; Khan, S.A. A review on feature selection methods for improving the performance of classification in educational data mining. *Int. J. Inf. Technol. Manag.* **2021**, *20*, 110–131. [[CrossRef](#)]
29. Al-Yarimi, F.A.M.; Munassar, N.M.A.; Bamashmos, M.H.M.; Ali, M.Y.S. Feature optimization by discrete weights for heart disease prediction using supervised learning. *Soft Comput.* **2021**, *25*, 1821–1831. [[CrossRef](#)]
30. Manur, M.; Pani, A.K.; Kumar, P. A prediction technique for heart disease based on long short term memory recurrent neural network. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 31–33. [[CrossRef](#)]
31. Gokulnath, C.B.; Shantharajah, S.P. An optimized feature selection based on genetic approach and support vector machine for heart disease. *Clust. Comput.* **2019**, *22*, 14777–14787. [[CrossRef](#)]
32. Javeed, A.; Zhou, S.; Yongjian, L.; Qasim, I.; Noor, A.; Nour, R. An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection. *IEEE Access* **2019**, *7*, 180235–180243. [[CrossRef](#)]
33. Ali, L.; Rahman, A.; Khan, A.; Zhou, M.; Javeed, A.; Khan, J.A. An automated diagnostic system for heart disease prediction based on chi<sup>2</sup> statistical model and optimally configured deep neural network. *IEEE Access* **2019**, *7*, 34938–34945. [[CrossRef](#)]
34. Hancer, E.; Xue, B.; Zhang, M.; Karaboga, D.; Akay, B. Pareto front feature selection based on artificial bee colony optimization. *Inf. Sci.* **2018**, *422*, 462–479. [[CrossRef](#)]
35. Jamro, W.A.; Shaikh, H.; Mahar, J.A. Comprehensive Analysis of Neural Network Techniques in Computational Linguistic Applications. *Asian Journal of Engineering. Sci. Technol.* **2016**, *2016*, 15.
36. Svozil, D.; Kvasnicka, V.; Pospichal, J. Introduction to multi-layer feed-forward neural networks. *Chemom. Intell. Lab. Syst.* **1997**, *39*, 43–46. [[CrossRef](#)]
37. Marateb, H.R.; Goudarzi, S. A noninvasive method for coronary artery diseases diagnosis using a clinically-interpretable fuzzy rule-based system. *J. Res. Med. Sci.* **2015**, *20*, 214–223. [[CrossRef](#)] [[PubMed](#)] [[PubMed Central](#)]
38. Hu, Z.; Melton, G.B.; Arsoniadis, E.G.; Wang, Y.; Kwaan, M.R.; Simon, G.J. Strategies for handling missing clinical data for automated surgical site infection detection from the electronic health record. *J. Biomed. Inform.* **2017**, *68*, 112–120. [[CrossRef](#)] [[PubMed](#)]
39. Lv, F. Data Preprocessing and Apriori Algorithm Improvement in Medical Data Mining. In Proceedings of the 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatre, India, 8–10 July 2021; pp. 1205–1208.
40. Singh, D.; Singh, B. Investigating the impact of data normalization on classification performance. *Appl. Soft Comput.* **2020**, *97*, 105524. [[CrossRef](#)]
41. Derhab, A. A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection. *IEEE Access* **2019**, *7*, 30373–30385.
42. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
43. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [[CrossRef](#)]
44. Pinkus, A. Approximation theory of the MLP model in neural networks. *Acta Numer.* **1999**, *8*, 143–195. [[CrossRef](#)]
45. Goodfellow, I.; Bengio, Y.; Courville, A. Back-propagation and other differentiation algorithms. *Deep. Learn.* **2016**, *2016*, 200–220.
46. Powers, D. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Mach. Learn. Technol.* **2011**, *2*, 37–63.
47. DeLong, E.R.; DeLong, D.M.; Clarke-Pearson, D.L. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics* **1988**, *44*, 837–845. [[CrossRef](#)] [[PubMed](#)]
48. Gruell, H.; Vanshylla, K.; Tober-Lau, P.; Hillus, D.; Schommers, P.; Lehmann, C.; Kurth, F.; Sander, L.E.; Klein, F. mRNA booster immunization elicits potent neutralizing serum activity against the SARS-CoV-2 Omicron variant. *Nat. Med.* **2022**, *28*, 477–480. [[CrossRef](#)] [[PubMed](#)]
49. Taha, A.A. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging* **2015**, *15*, 29. [[CrossRef](#)] [[PubMed](#)]

50. Gareth, J.; Witten, D.; Trevor, H.; Robert, T. *An Introduction to Statistical Learning: With Applications in R*; Springer: Berlin/Heidelberg, Germany, 2021; ISBN 978-1071614174.
51. Sikalidis, A.K.; Kristo, A.S.; Reaves, S.K.; Kurfess, F.J.; DeLay, A.M.; Vasilaky, K.; Donegan, L. Capacity Strengthening Undertaking—Farm Organized Response of Workers against Risk for Diabetes: (C.S.U.—F.O.R.W.A.R.D. with Cal Poly)—A Concept Approach to Tackling Diabetes in Vulnerable and Underserved Farmworkers in California. *Sensors* **2022**, *22*, 8299. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.