
**Arab American University**

**Faculty of Graduate Studies**

# Intelligent Medical Diagnosis and Decision Support Model Based on Neural Networks and Rule Based System

By

**Mohammed Ibrahim Nemir Dwaikat**

Supervisor

**Prof. Dr. Mohammed Awad**

**This Thesis was submitted in partial fulfillment of the requirements for the Master's degree in Computer Science.**

May / 2020

**© Arab American University – 2020**
**All rights reserved**

# Intelligent Medical Diagnosis and Decision Support Model Based on Neural Networks and Rule Based System

By

**Mohammed Ibrahim Nemir Dwaikat**

This thesis was defended successfully on **06th /May/2020** and approved by:

| Committee Members | Signature |
|---|---|
| **1.** Supervisor: **Prof. Dr. Mohammed Awad** |  |
| **2.** Internal Examiner: **Dr.Rami Hadrob** |  |
| **3.** External Examiner: **Dr.Fady Draidi** |  |

## Declaration

I declare that this thesis entitled “Intelligent Medical Diagnosis and Decision Support Model Based on Neural Networks and Rule Based System” is my own work and has been composed solely by myself and does not contain and work form others researcher and has not been submitted for and other degree or scientific except the reference is made.

## Dedication

I dedicate this thesis to my family and friends for their unconditional love and support they have shown and given to me. To the person who is no longer around when I needed the most, whom her absence made everything much more difficult than it already is. So, to your absence which I have filled writing this thesis, I dedicate it.

.

## Acknowledgments

I would like to take this opportunity to express my deep regards to Prof. Mohammed Awad for his advices, support, and time which he spent on reviewing my work. Prof. Mohammed provided valuable suggestions that have had significant impact, and helped in overcoming many obstacles in writing this thesis in the best way.

## Abstract

The human body is a vital source of data, which is important for human health. Medical imaging is one of the processes that produce different kinds of human body data. While some data in the form of images, the others are signals. This data daily used to diagnose different kinds of diseases. There are many different bio-signals that can be collected; some important signals are Electrocardiogram (ECG), Electroencephalogram (EEG) and Electromyogram (EMG). These signals are collected from different human organs and utilized in diagnosed different diseases. The designing and implementation of intelligent computer programs that try to emulate with human intelligence are a sign of the integration of various sciences and areas of knowledge. One important field is the improvement that allows appropriate assistance to physicians in decision-making. The development of technologies associated with Artificial Intelligence (AI) techniques that are applied on medicine, represents a novel perspective, which can reduce costs, time, and medical errors.

The integration between artificial intelligence and the medical system is vital, and a lot of efforts have been made in this area. While this field still needs more and more investigation, intelligent medical and diagnostic decision support systems could consume these amounts of data, and utilize it to improve healthcare. The using of Artificial Intelligence methods in medical diagnosis can Benchmark from several of its main techniques such as expert systems (diagnosis based on rules, probabilities), fuzzy logic (diagnosis based on classification), neural networks (diagnosis based on training and recognition), applied data mining (diagnosis through the pattern recognition).

In this thesis, a new method are produced to support the take of medical decisions by combining the intelligent computational systems with medical. In general, there is an approximate shared procedure follow to manipulate with these problems; starts with de-noising the signals, and then applies feature extraction methods (reduction and selection). While the last task is to classify or recognize a different pattern used in medical diagnoses to make a decision in determined medical cases. Therefore, this study proposed a technique that using useful physiological variables for diagnosis heart disease, a hybrid system that combined expert systems and neural networks for the implementation of Intelligent Medical Diagnosis System in Decision Support in medical application is used. Another goal that is achieved was the use of optimization of neural network parameters by optimization algorithms with the objective of enhancing the system. A hybrid system that combines Genetic Algorithm (GAs), Biogeography-Based Optimization (BBO) with neural networks (NNs) [GAsBBO-MLPNNs], BBO and particle swarm optimization (PSO) methods combined with the neural network was used to improve the performance of the systems. The idea of this thesis concentrates on the medical diagnosis system for heart disease using artificial intelligence techniques. The proposed method produces better performance than previous works, where the GAsBBO-MLPNNs method performance parameters result represented as 94.5% 95.6%, 89.94% accuracy, G-mean, and F-measure respectively. The Intelligent Medical Diagnosis System has achieved a prediction accuracy of 95.097% using Neuro-Fuzzy model with triangular membership function.

**Table of Contents**

## List of Figures

## List of Tables

## List of Abbreviations

| | |
|---|---|
| EDG | Electrocardiogram |
| EEG | Electroencephalogram |
| EMG | Electromyogram |
| CAD | Coronary Artery Disease |
| CHD | Coronary Heart Disease |
| AI | Artificial Intelligence |
| LAD | Left Anterior Descending |
| LCX | Left Circumflex |
| RCA | Right Coronary Arteries |
| HF | Heart Failure |
| NNs | Neural Networks |
| CDSS | Clinical Decision Support System |
| HD | Heart Disease |
| FES | fuzzy Expert System |
| GAs | Genetic Algorithms |
| PSO | Particle Swarm Optimization |
| BBNNs | Block-Based Neural Networks |
| SVM | Support Vector Machine |
| SST | Synchro Squeezing Transform |
| CNNs | Convolutional Neural Networks |
| BBO | Biogeography-Based Optimization |
| HSI | Habitat Suitability Index |
| TP | True Positive |
| FP | False-Positive |
| FN | False-Negative |
| TN | True Negative |

## 1.1 Introduction

The medical artificial intelligence in its conception depends on the structure of medical information and set of other sciences, methods, and techniques that include computer science, the systemic analysis applied to medicine, statistics, logic, linguistics, decision-making theory and modeling [1]. Expert or knowledge-based systems which is a field of AI, is nothing more than intelligent computer programs that simulate the reasoning chains which an expert makes to solve a problem in this domain; for example, the physician makes a diagnosis. [2] To achieve this, the system is endowed with a set of principles or rules that infer new evidence from previously known information. One of the most important problems that artificial intelligence has addressed in health is the treatment of heart disease [3].

Heart disease comprises a wide range of cardiovascular diseases. Types of heart disease include Coronary Artery Disease (CAD) which is the most common type of heart disease, another types are Arrhythmia, heart failure, heart valve disease, heart muscle disease, congenital heart disease [4]. The plaque that accumulates in the inner surface of the coronary arteries causes the inner surface to become irregular and narrow. These plaque leads to blockage in the main arteries of the heart and to reduce the blood flow to the heart muscle. Over time, this blockage can lead to a heart attack [5].

There are many responsible factors for heart disease such as smoking, high blood pressure, family history, etc. These factors used to make a decision by evaluating the test result of the patients. This process is difficult because it's not easy to consider the number of factors used in the evaluation process, as a result, the diagnosing of heart disease requires a high experience from the scientists. However, recent research shows that artificial intelligence plays an important role in predicting and preventing different types of heart diseases [6].Accurate diagnosis of heart

disease in the early stage can help in saving patient life and Provide appropriate treatment, earlier stage requires grate effort because it depends on a variety of factors. For this reason, it's important to develop a medical decision support system in order to help the doctors and to save the patient life [7].

Heart disease becomes the leading cause of death in the US; About 610,000 people die of heart disease in the US every year, which is 1 of every 4 deaths. Coronary heart disease (CHD) is a common type of heart disease that kills over 370,000 people every year [8]. For that, a several tools and methods were proposed to develop an effective support medical decision support system. Moreover, day by a day, there is new methods and tools are continuing to be developed by the researcher in this field.

Artificial Intelligence (AI) is a term that implies using a computer to do intelligent behavior with minimal intervention from a human. AI term is applicable to a broad range of items such as medical diagnosis and health care. AI in medicine has two main branches: physical and virtual. [9] The physical branch is represented by using robots in order to assist in surgeries and to assist the elderly patient. While the virtual branch is represented by collect information from electronic health records and signals to use it in control health management systems, and active guidance to the doctors in diagnosing diseases and make treatment decisions [9].

The researchers collect signals from different human organs and analyze them using various techniques such as neural network, fuzzy logic, support vector machine, etc. to diagnose the diseases in high accuracy and less time. Some important signals are Electrocardiogram (ECG) and Electromyogram (EMG), in [10] the author uses an EEG signal to diagnose Alzheimer's disease. Neural Networks (NNs) are paradigms computational based on mathematical models with the ability of strong pattern recognition. They are calculation algorithms based on an

analogy of the nervous system, which tries to imitate the human ability to learn, making it learn to identify patterns of association between inputs (predictive variables) and their dependent states (outputs). Neural Networks (NNs) is the widest classification technique used, where the systems have the ability to learn through training numbers of neural networks then combine their results, and it has the ability to generalize the results from the training data [11].

Evolutionary Algorithms (EAs) represent a simulation strategy to solve complex problems about the basis of the theory of natural evolution and the theory of genetic variation [12].

In this thesis, firstly, the aim is to improve the accuracy of heart disease prediction using a combination model of NNs and EAs [13]. Secondly, building an intelligent decision support model using the Neuro-fuzzy model. So, three optimization algorithms in which biogeography based optimization (BBO) [14], genetic algorithms (GAs) [15], and particle swarm optimization (PSO) [16] were combined with neural networks that are proposed to address this problem. Also, the Neuro-Fuzzy system which combines fuzzy logic and neural networks system is used to build an intelligent decision support model based in a rule-based expert system. During this thesis, all the experiments were performed on the Z-Alizadeh Sani dataset, where the process starts with using the dataset as input and apply the preprocessing method which is feature selection and data normalization. Then using the selected features to diagnosis heart disease and to support the take of medical decisions by combining fuzzy logic (expert systems) and neural networks.

## 1.2 Objective

In the heart medical diagnoses, there is no room for error because this error related to human life. Misdiagnosis of heart problems may lead to the death of patients because they do not receive proper treatment. The general objective of this research is to present a method of classification

and rule-based expert systems to improve the accuracy of the heart disease diagnosis system. The heart disease classification phase depends on a hybrid model that combines two optimization algorithms which are genetic algorithms (GAs), and geographical based optimization (BBO) with multilayer perceptron neural networks (MLPNNs), this model is called GAsBBO-MLPNNs. The second phase is the use of a Neuro-Fuzzy model to generate an intelligent decision support model that depends on rule based expert system. The two phases of the system applied to an international dataset called the Z-Alizadeh Sani dataset. The efficiency of the applied model compared with the use of each optimization algorithm combined with MLPNNs like Particle swarm optimization (PSO), PSO-MLPNNs, BBO-MLPNNs, and GAs-MLPNNs.

## 1.3 Contribution

This thesis presents several models for heart disease classification and rule-based expert systems on the Z-Alizadeh Sani dataset. With regard to classification, GAsBBO-MLPNNs, PSO-MLPNNs, BBO-MLPNNs models were evaluated with a set of new preceding works that very close to our tests which is GAs-NNs and SMO classifiers. GAsBBO-MLPNNs model produces better performance in terms of accuracy and specificity, even if they complete it in terms of sensitivity. Also, the GAsBBO-MLPNNs algorithm got better performance than BBO-MLPNNs, PSO-MLPNNs in accuracy, G-mean, and F-measure. Regards to expert systems, an Intelligent Decision Support Model using the Neuro-Fuzzy model which combined Fuzzy-logic and neural networks is implemented to help specialists in making a medical decision.

## 1.4 Overview

The remainder of this thesis is arranged as the following. In Chapter 2, a background that includes the description of the Benchmark dataset and the National dataset, then a literature

review of the related work in heart disease diagnosis applications and some techniques used to perdition and classification of the heart disease. In Chapter 3, the description of the preprocessing phases which is: the feature selection and data normalization firstly. Secondly, GAs, PSO, and BBO algorithms were explained to be combined with MLPNNs, the general method procedure will be illustrated, and Neuro-Fuzzy expert System will be explained to implement an intelligent decision support model. Finally, different performance measurements were introduced. In Chapter 4, the standardization method selection considerations firstly. Secondly, all experiments with the Z-Alizadeh Sani dataset were illustrated. And a brief discussion of the results as described. Finally, all experiments using ANFIS model were illustrated In Chapter 5, the conclusion and the future work will be presented. Finally, the appendix part includes the results of all experiments that were performed on the Z-Alizadeh Sani dataset to optimize some parameters of the BBO-MLPNNs, GAs-MLPNNs, and PSO-MLPNNs models.

# Chapter 2

# Background

## 2.1 Background

According to statistics, heart disease becomes one of the most diseases that lead to death around the world, because of this, many tools have been developed for diagnosing heart disease. In this work, multi-layer neural networks with evolutionary algorithms (GAs, PSO and BBO) is used to build a heart disease classification models, and generate an intelligent decision support model using Neuro-Fuzzy expert system. So, in this chapter, experiments on The Z-Alizadeh Sani data set are conducted.

## 2.2 Benchmark Datasets Description

The Z-Alizadeh Sani dataset contains 303 random records of patients, each record has 54 features. [17] These features used as indicators of CAD for patients. The features and their valid ranges are represented in tables 2.1 to 2.4. According to these features, the patients categorized to CAD or Normal, a patient is categorized as CAD if at least one of the left anterior descending (LAD), left circumflex (LCX), and right coronary arteries (RCA) has stenos is greater than 50%, and otherwise, a patient considered as Normal. The features are divided into four categories: demographics, symptoms, ECG and “laboratory and echo "features. Table 2.5 shows a screen shoot of the data set after applying future selection method.

ECG: Electrocardiogram, which is a record of the heartbeat produced by electrocardiography. The ECG features represented in table 2.1.

Table 2. 1The ECG features of Z-Alizadeh Sani dataset and their valid range

| Feature name | Range |
|---|---|
| Rhythm | Sin, AF |
| Q wave | Yes, No |
| ST-elevation | Yes, No |
| ST depression | Yes, No |
| T inversion | Yes, No |
| LVH (Left Ventricular Hypertrophy) | Yes, No |
| Poor R-wave progression | Yes, No |

Symptoms: the physical changes that are regarded as indicating a condition of a particular disease. The Symptoms features are represented in table 2.2.

Table 2. 2: The Symptoms features of Z-Alizadeh Sani dataset and their valid range

| Feature name | Range |
|---|---|
| BP (Blood Pressure mm Hg) | 90–190 |
| PR (Pulse Rate ppm) | 50–110 |
| Edema | Yes, No |
| Weak peripheral pulse | Yes, No |
| Lung rales | Yes, No |
| Systolic murmur | Yes, No |
| Diastolic murmur | Yes, No |
| Typical chest pain | Yes, No |
| Dyspnea- | Yes, No |
| Function class | 1, 2, 3, 4 |
| Atypical | Yes, No |
| Nonanginal chest pain | Yes, No |
| Exertional chest pain | Yes, No |
| Low Th Ang (low-Threshold angina) | Yes, No |

Demographic: the statistical characteristics of the population such as age, income, and sex. The Demographic features are represented in table 2.3.

Table 2. 3: The Demographic features of Z-Alizadeh Sani dataset and their valid ranges.

| Feature name | Range |
|---|---|
| Age | 30–86 |
| Weight | 48–120 |
| Sex | Male, female |
| BMI (Body Mass Index Kg/m2) | 18–41 |
| DM (Diabetes Mellitus) | Yes, No |
| HTN (Hypertension) | Yes, No |
| Current smoker | Yes, No |
| Ex-smoker | Yes, No |
| FH (Family History) | Yes, No |
| Obesity | Yes if MBI > 25, No otherwise |
| CRF (Chronic Renal Failure) | Yes, No |
| CVA (Cerebrovascular Accident) | Yes, No |
| Airway disease | Yes, No |
| Thyroid disease | Yes, No |
| CHF (Congestive Heart Failure) | Yes, No |
| DLP (Dyslipidemia) | Yes, No |

Laboratory and echo: echo (echocardiogram) is a graphic outline of the heart's movement which evaluates the chambers and valves are pumping blood the heart, while the Laboratory features are obtained from laboratory tests. Laboratory and echo are represented in table 2.4.

Table 2. 4: The Laboratory and echo features of Z-Alizadeh Sani dataset and their valid range.

| Feature name | Range |
|---|---|
| FBS (Fasting Blood Sugar mg/dL) | 62–400 |
| Cr (Creatine mg/dL) | 0.5–2.2 |
| TG (Triglyceride mg/dL) | 37–1050 |
| LDL (Low-Density Lipoprotein mg/dL) | 18–232 |
| HDL (High-Density Lipoprotein mg/dL) | 15–111 |
| BUN (Blood Urea Nitrogen mg/dL) | 6–52 |
| ESR (Erythrocyte Sedimentation Rate mm/h) | 1–90 |

**The definitions and terms of the features are the following:**

1. Typical Chest Pain (Angina): Chest pain due to an inadequate supply of oxygen to the heart muscle. The pain is typically severe and crushing, and it is characterized by a feeling of pressure and suffocation just behind the breastbone.
2. Atypical (Non-angina): is a term used to describe discomfort or pain centered in the chest that is not cardiac pain, chest pain not heart-related and not of burning quality.
3. DM (diabetes mellitus): is a chronic disease associated with abnormally high levels of the sugar glucose in the blood.
4. T-inversion: Inverted T waves are associated with myocardial ischemia. The inversion of a T wave is not specific for ischemia, and the inversion itself does not correlate with a specific prognosis.
5. Region RWMA: regional wall motion abnormalities (RWMA) on an echocardiogram means that a region of the heart muscle is not contracting as it normally should. No

regional wall motion abnormality means there is no abnormality in the contraction of the various parts of the heart muscle.

6. Hypertension: (HTN or HT), also known as high blood pressure (HBP), is a long-term medical condition in which the blood pressure in the arteries is persistently elevated. Hypertensive heart disease refers to the heart working under increased pressure causes some different heart disorders.
7. TG (Triglyceride test): A lipid profile is a test that measures the level of fats in your blood, including triglycerides and cholesterol, a waxy, fatty substance found in every cell of your body. If you have high levels of both LDL (bad) cholesterol and triglycerides, you may be at an increased risk for a heart attack or stroke
8. Pulse rate: is the number of heart beats per minute. The resting pulse rate for an average adult is between 60 to 100 beats per minute.
9. Diastolic heart murmurs: are heart murmurs heard during diastole. Diastolic murmurs start at or after S2 and end before or at S1. Many involve stenosis of the atrioventricular valves or regurgitation of the semilunar valves.
10. Dyspnea: Difficulty breathing; shortness of breath. Dyspnea is a sign of serious disease of the airway, lungs, or heart.
11. ESR: Abbreviation for erythrocyte sedimentation rate, a blood test that detects and monitors inflammation in the body. It measures the rate at which red blood cells (RBCs)

Table 2. 5: Screen shoot of the data set after applying future selection method.

| Typical Chest Pain | Atypical | Age | Nonanginal | DM | Tinversion | FH | Region RWMA | HTN | TG | PR | Diastolic Murmur | Current Smoker | Dyspnea |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 53 | 0 | 0 | 1 | 0 | 0 | 1 | 250 | 80 | 0 | 1 | 0 |
| 1 | 0 | 67 | 0 | 0 | 1 | 0 | 4 | 1 | 309 | 80 | 0 | 0 | 0 |
| 1 | 0 | 54 | 0 | 0 | 0 | 0 | 2 | 0 | 103 | 100 | 0 | 1 | 0 |
| 0 | 0 | 66 | 1 | 0 | 0 | 0 | 0 | 1 | 63 | 80 | 1 | 0 | 1 |
| 0 | 0 | 50 | 0 | 0 | 0 | 0 | 0 | 1 | 170 | 80 | 0 | 0 | 1 |
| 1 | 0 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 139 | 70 | 0 | 1 | 0 |
| 1 | 0 | 55 | 0 | 0 | 1 | 0 | 4 | 0 | 83 | 80 | 0 | 0 | 0 |
| 1 | 0 | 72 | 0 | 1 | 1 | 0 | 4 | 0 | 80 | 70 | 0 | 1 | 0 |
| 0 | 0 | 58 | 1 | 0 | 0 | 0 | 0 | 0 | 79 | 50 | 0 | 0 | 1 |
| 1 | 0 | 60 | 0 | 1 | 0 | 0 | 2 | 0 | 80 | 70 | 0 | 0 | 1 |
| 1 | 0 | 58 | 0 | 0 | 0 | 0 | 0 | 1 | 160 | 70 | 0 | 0 | 1 |
| 1 | 0 | 80 | 0 | 0 | 1 | 0 | 3 | 1 | 104 | 100 | 0 | 0 | 0 |
| 0 | 1 | 70 | 0 | 1 | 1 | 0 | 4 | 1 | 235 | 84 | 0 | 0 | 0 |
| 0 | 1 | 67 | 0 | 1 | 1 | 0 | 2 | 1 | 358 | 74 | 0 | 0 | 1 |
| 0 | 1 | 66 | 0 | 1 | 0 | 0 | 0 | 1 | 125 | 80 | 0 | 0 | 1 |
| 1 | 0 | 59 | 0 | 1 | 0 | 0 | 0 | 0 | 130 | 70 | 0 | 0 | 1 |
| 0 | 0 | 41 | 0 | 0 | 0 | 0 | 0 | 0 | 69 | 80 | 0 | 1 | 1 |
| 0 | 1 | 68 | 0 | 0 | 0 | 1 | 0 | 0 | 114 | 70 | 0 | 0 | 0 |
| 1 | 0 | 60 | 0 | 1 | 0 | 0 | 0 | 1 | 170 | 74 | 0 | 0 | 0 |
| 1 | 0 | 65 | 0 | 1 | 0 | 0 | 1 | 1 | 230 | 70 | 0 | 0 | 0 |
| 0 | 1 | 47 | 0 | 0 | 0 | 1 | 0 | 0 | 106 | 70 | 0 | 0 | 0 |
| 1 | 0 | 66 | 0 | 0 | 0 | 0 | 0 | 1 | 247 | 80 | 0 | 0 | 0 |
| 1 | 0 | 66 | 0 | 1 | 1 | 0 | 0 | 1 | 74 | 80 | 0 | 0 | 0 |
| 1 | 0 | 72 | 0 | 1 | 0 | 0 | 0 | 1 | 190 | 76 | 0 | 0 | 1 |
| 1 | 0 | 50 | 0 | 1 | 1 | 1 | 0 | 0 | 84 | 60 | 0 | 0 | 0 |
| 1 | 0 | 65 | 0 | 0 | 1 | 0 | 3 | 0 | 140 | 80 | 0 | 0 | 0 |
| 0 | 0 | 56 | 0 | 0 | 0 | 0 | 0 | 0 | 103 | 75 | 1 | 0 | 1 |
| 0 | 1 | 50 | 0 | 0 | 0 | 0 | 0 | 1 | 340 | 80 | 0 | 0 | 0 |
| 1 | 0 | 80 | 0 | 0 | 0 | 0 | 4 | 1 | 168 | 70 | 0 | 0 | 0 |

## 2.3 National Dataset

From the beginning of this research and for a period of one year, it is tried to collect Heart disease dataset from Palestine. The start was searching for data in the public sector which presented by the ministry of health. National Hospital in Nablus was asked to provide the ECG signal for patients, but the available data were paper-based and extracting features from the ECG signals takes extra effort, which was not the main concern at that stage. Then it is requested a dataset from Palestine Medical Complex in Ramallah, but the dataset which they provided was incomplete and cannot be used in the scientific research. After that, it was decided to search in the private sector, 'An-Najah National University Hospital' was selected because it contains a special department for heart diseases. Several official letters in addition toa thesis proposal based on their request were sent in order to ensure that the dataset will be used in scientific research only. But then, unfortunately, the request was rejected.

## 2.4 Related Works

In [18], the authors proposed a data mining method for the diagnosis of coronary artery disease on the Z-Alizadeh Sani dataset. They made a comparison study between four algorithms: Naïve Bayes, SMO (Sequential Minimal Optimization), Bagging with SMO classifiers, and Neural Network, also they create three features which is LAD, LCX, and RCA to improve the performance of the proposed. The highest accuracy which has been achieved is 92.09% by using the SMO algorithm with feature selection and creation technique. In [19] the authors proposed, a machine learning approach that using support vector machine (SVM) to classify and diagnosis CAD. To improve the prediction of the Coronary artery disease, the proposed "feature engineering method" uses the result of three classifiers, i.e. LAD, LCX and RCA in the training dataset. The proposed applied on the Z-Alizadeh Sani dataset which extended to 500 records. It has achieved accuracy, sensitivity, specificity, 96.40%, 100%, and 88.1%, respectively for detecting CAD.

In [20], the authors build support systems to predict heart failure risks by using artificial neural network and fuzzy logic. The proposed has two stages, in the first one, 13 attributes were evaluated and ranks to determine their contribution in effect the heart failure. In the second stage, ANN learning algorithm is used to build the prediction for the HF risk, the result shows that the proposed archived prediction accuracy of 91.10%, which is 4.4% higher than the conventional ANN method. In [21], the authors proposed a Clinical Decision Support System (CDSS) that use Fuzzy inference system to examine the existence of heart diseases, and use Fuzzy Analytic Hierarchy (AHP) Process to compute the weight of the different factors that affect deploying HD. The proposed has four steps, selecting criteria and sub-criteria, weighting the sub-criteria,

assessing patient's condition, and calculating like hood of heart diseases. The system helps the specialist when a high probability of HF is determined.

In [22], big data mining and cloud computing used in "Disease Diagnosis and Treatment Recommendation System" (DDTRS).The proposed consisted of two modules: a Density-Peaked Clustering Analysis (DPCA) algorithm to identify the link between disease and symptoms based, and a disease diagnosis and treatment recommendation module. The result shows that the proposed provide a high-quality recommendation with low latency response. In [23], another decision support system based on Fuzzy Expert System (FES) was proposed for medical diagnosis. The proposed helped doctors in making more accurate medical diagnosis by entering the symptoms information as input. The proposed has two methods for patient input, the first one choice of up to two linguistic variables, and the other use a numerical range, the result shows that the multiple numeric entries method is better in diagnosing kidney stone, the proposed give a high degree of accuracy (85% accuracy in diagnosing Kidney Infection and 87.5% in diagnosing kidney Stone).

In [24], the proposed "Adaptive weighted fuzzy rule-based system" based in genetic algorithm (GA) and modified dynamic multi-swarm particle swarm optimization (MDMS-PSO) is to predict the risk of level of heart disease. The proposed woks as follows: pre-process the dataset, used a statistical method to select the effective attribute, and weighted the selected using GA. They used (MDMS-PSO) to optimize the membership function, and use the generated fuzzy knowledge base to build the ensemble FS. In [25], the author proposed a process that uses NN and SVM to extract features from four types of ECG signal. These features used to diagnosis the cardiac abnormalities: Normal, left bundle branch block, right bundle branch block, paced beats. The proposed work as follows: eliminate the noise of the signal in pre-processing stage, then use

the ECG signal to extract features which is used to classify the data using NN and SVM classifier, they achieve a high prediction performance and average accuracy of 96.67% and 98.39% in NN and SVM.

In [26], the authors proposed a system that used an ECG signal to classify the heartbeats of the patients using Neural Network, which uses Block-based Neural Network (BBN). They used the PSO algorithm to optimize and training the BBNN structure, and use the extracted features from the ECG signal as BBNN input, and used the PSO algorithm to optimize the BBNN input parameters to overcome the variation in ECG signal from person to another, the proposed provide a classification accuracy of 97%.

In [27], the author proposed an adaptive non-harmonic model and synchrosqueezing transform (SST) to describe the ECG pattern on MIT-BIH database. The proposed enhanced the detection of a heartbeat between normal and abnormal arrhythmia. They used SST to validate and train SVM classifier on portion of annotated beat database. The proposed achieved positive predictive value compared with other prediction algorithm using many more features

The author in [28], proposed a model that used neural network (NN) and genetic algorithm (GA) to predict the cardiovascular disease. It can detect the coronary artery disease without the need of invasive diagnosis method. The proposed identified the initial data using genetic algorithm, and increase the performance of neural network by 10% through enhancing the primary weight used in it. They achieved an accuracy of 93.85%, sensitivity of 97% and specificity of 92% in predicting coronary artery disease diagnosis. In [29] the authors used two and five seconds ECG signal to diagnose of Coronary Artery Disease (CAD) using convolutional neural network (CNN). The proposed differentiates between normal and abnormal ECG using deep CNN, and

helps the doctors in making a reliable decision making of CAD using ECG signals. The proposed achieves a diagnose accuracy of 94.95% for the 2 second ECG signal and 95.11% accuracy for the five-second ECG signals. The disadvantages of the proposed that it requires a fixed-length ECG signal and a huge database for the training process.

In [30], a Heart Disease Prediction System was proposed by using multilayered feed-forward neural network and back-propagation neural network in four stages which is “normal, stage1, stage2, stage3 “of heart disease. The proposed used the forward pass to calculate the output and compare it with the desired value, and backward pass to alter the value of the weights, and repeat the forward and backward passes until the error is low enough, it provides a better performance than the traditional diagnosis methods and achieve an accuracy of 92%.In [31] the authors proposed a data-mining algorithm for feature creation and selection on Z-Alizadeh Sani dataset to make a rule-based classifier. The method added three new features to the data set that regarding the LAD, LCX, and RCA. They made a comparisons between Naïve Bayes classifier, Sequential Minimal Optimization, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and C4.5 with and without using the created features. The result shows that the SMO algorithm got the highest accuracy of 91.43% using the selected features and 92.09% using the selected and created features.

In [32], the authors proposed a machine learning algorithm for diagnosing CAD on the Z-Alizadeh Sani dataset and they extend the number of the sample from 303 to 500 cases, three classifiers were used to predict the stenosis of coronary arteries LAD, LCX, and RCA. Also, they made comparisons between various types of machine learning methods which are Artificial Neural Networks (ANN), Support Vector Machine (SVM), Random Forest (RF), Naïve Bayes (NB), k-Nearest Neighbor (KNN) and ensemble learner which is the combination of these five

ML algorithms. The methods archive an average accuracy higher than 80% and the Artificial Neural Network reached 93% AUC (area under ROC) which is the best performance out of six methods.

In [33], the authors applying a machine learning approach using radial basis function (RBF) and support vector machine (SVM). The proposed handles the model uncertainty in diagnosing the stenosis major coronary arteries in individual LAD, LCX, and RCA on the Z-Alizadeh Sani dataset. They enhanced the proposed performance by using the accuracy rate and the hyper plane distance from a sample during the training phase. The proposed achieved accuracy rates of 82.67%, 83.67% and 86.43% for RCA, LCX, and LAD respectively.

In [34] the author's design 'Automatic Heart Disease Diagnosis System Based on Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference Systems' for diagnosing heart disease on Cleveland dataset. The first system is based on MLP and the second is based on Adaptive Neuro-Fuzzy Inference Systems (ANFIS) approach. The authors divide the dataset into two parts, where 80% of the dataset for training the model and the remaining 20% for testing. The ANFIS approach achieves an accuracy of 75.93%. The author in [35] proposed an 'Advisory System for Medical Assistance by using Neuro-Fuzzy System'. The proposed using Neuro-Fuzzy System with Sugeno type fuzzy model membership for diagnosing the cancer disease. The system uses the Capacitance Relaxation, PH of cancer cell, Catecholamine, and Metastasis as input parameters with three functions for each one, while the output of the system represents the cancer stages that help the specialists to provide the appropriate treatment.

But in this thesis, it is proposed that an Intelligent Medical Diagnosis and Decision Support Model is to improve the accuracy of diagnosing heart disease on the Z-Alizadeh Sani dataset

with 303 samples. GAsBBO-NN, BBO-MLPNN, and PSO-MLPNN models are used to build Intelligent Medical Diagnosis models which are models depends on hybrid models that combine multilayer perceptron with optimization algorithms to improve the accuracy of the heart disease diagnosis system. Also, the Neuro-fuzzy model is used to build an Intelligent Decision Support Model that combines fuzzy logic and neural networks. 14 features in optimization experiments and 7 features in the Neuro-fuzzy experiment based on “Weights by SVM”. [36].from [28] were used .The proposed GAsBBO-MLPNNs produced a result of 93.85% 95.6%, 89.94% of accuracy, G-mean, and F-measure respectively.

# Chapter 3

# The Proposed Method

## 1.1The Proposed Method

This chapter illustrates the proposed method which aims to improve the classification accuracy of heart disease, and implement Decision Support Model for diagnosis Heart disease on the Z-Alizadeh Sani dataset. It begins by selecting the dataset, and describing the preprocessing steps. Then the chapter illustrates the deployed models: the classification models which consist of combine evolutionary algorithms (BBO, GAs, and PSO) with MLPNN, and the Decision Support Model for diagnosis Heart disease using Neuro-Fuzzy model. Finally, it illustrates the metrics used to measure models performance.

## 1.2 Preprocessing Phase

Data preprocessing is an important step in machine learning. Different preprocessing sub-step maybe used depending on the nature of the dataset [37], Data-type portability, feature selection, and data cleaning were used in this thesis. This section will describe these steps in detail.

### 1.2.1 Feature Selection

For feature selection, the features selected in [28] is used which represented in table 3.1. The selection was done based on "Weights by SVM" method. This method uses F-score to measure the weights of the features [36].

F-score is a technique that measures the discrimination of two sets of real numbers. For a training instance $x_k$, k=1, 2,…., m, and $n_+$ is a number of positive instances, and $n_-$ is a number of the negative instances then F-score of the *i* th feature is calculated using equation 3.1. The feature is likely to be more discriminative if it has a high F-score [36].

$$F(i) \equiv \frac{\left(\bar{x}_i^{(+)} - \bar{x}_i\right)^2 + \left(\bar{x}_i^{(-)} - \bar{x}_i\right)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}\left(x_{k,i}^{(-)} - \bar{x}_i^{(-)}\right)^2} \qquad 3.1$$

Where the *i*th feature average of the whole, positive and negative instances are represented with$\bar{x}_i$, $\bar{x}_{,i}^{(+)}$,$\bar{x}_i^{(-)}$, Respectively; $x_{k,i}^{(+)}$ is the *i*th feature of the *k*th positive instance, and $x_{k,i}^{(-)}$ is the *i*th feature of the *k*th negative instance.

Table 3. 1: The Selected Feature and its Weights.

| Feature | Weight |
|---|---|
| Typical chest pain | 1 .0 |
| Atypical | 0 .88 |
| Age | 0 .88 |
| Nonanginal | 0 .58 |
| DM | 0 .44 |
| Tinversion | 0 .44 |
| FH | 0 .42 |
| Region RWMA | 0 .40 |
| HTN | 0 .40 |
| TG | 0 .35 |
| PR | 0 .33 |
| Diastolic murmur | 0 .32 |
| Current smoker | 0 .31 |
| Dyspnea | 0 .31 |
| ESR | 0 .29 |
| BP | 0 .27 |
| Function class | 0 .25 |
| Sex | 0 .24 |
| FBS | 0 .24 |
| St depression | 0 .23 |
| St elevation | 0 .21 |
| Q wave | 0 .20 |

### 1.2.2 Data Normalization

Normalizing data is an important preprocessing step in machine learning to prevent one feature dominates the other features; normalization aims to make data points of all features have the

same scale to have the same important. There are many data normalization methods such as min-max normalization and standardization [37].

1. Min-Max normalization: it performs a linear transformation on the data, it scales the attribute to a fixed range, in this work the range between [-1,1] is used, Min-Max normalization is calculated using the following equation:

$$y = 2\frac{x - x_{min}}{x_{max} - x_{min}} - 1 \qquad 3.2$$

Where *y* is the normalized value, *x* is the original value of the feature, $x_{min}$ is the minimum value of the feature and $x_{max}$ is the maximum value of the feature.

2. Standardization: it's a data transformation method that standardizes the data of each feature to have zero mean and one standard deviation. The data standardize using the following equation:

$$z_i^j = \frac{x_i^j - \mu_j}{\sigma_j} \qquad 3.3$$

Where$x_i^j$: is the j attribute of the $i^{th}$records, $\mu_j$: is the mean of the feature j, and $\sigma_j$ is the standard deviation of feature j.

## 1.3Building Models Phase

MLP and evolutionary algorithms are used to build heart disease classification of The Z-Alizadeh Sani data set. Evolutionary algorithms (genetic algorithm, particle swarm optimization, biogeography-based optimization) are an effective optimization technique that used to find a set of optimal weights for neural networks, while MLPNNs are a feed-forward artificial neural network that used to classify the data. This section will describe these algorithms in details.

### 1.3.1 Multi-Layer Perceptron Neural Networks

Neural networks are learning systems inspired by simulating the biological system of the human brain. [38]. It has the ability to learn and represent information and mapping it to the corresponding output that needs to predict. The most used type of NNs is the multilayer perceptron (MLP)[39] , it is a feed-forward neural network that consists of three or more layers: an input layer, hidden layer, and output layer, where each layer has a number of neurons n, h, and m in order. MLPNNs are fully connected; each neuron in the one layer is connected to every neuron in the next layer with a certain weight, each connection has different weight value which is determined using the learning process. The structure of MLPNNs is depicted in figure 3.4.

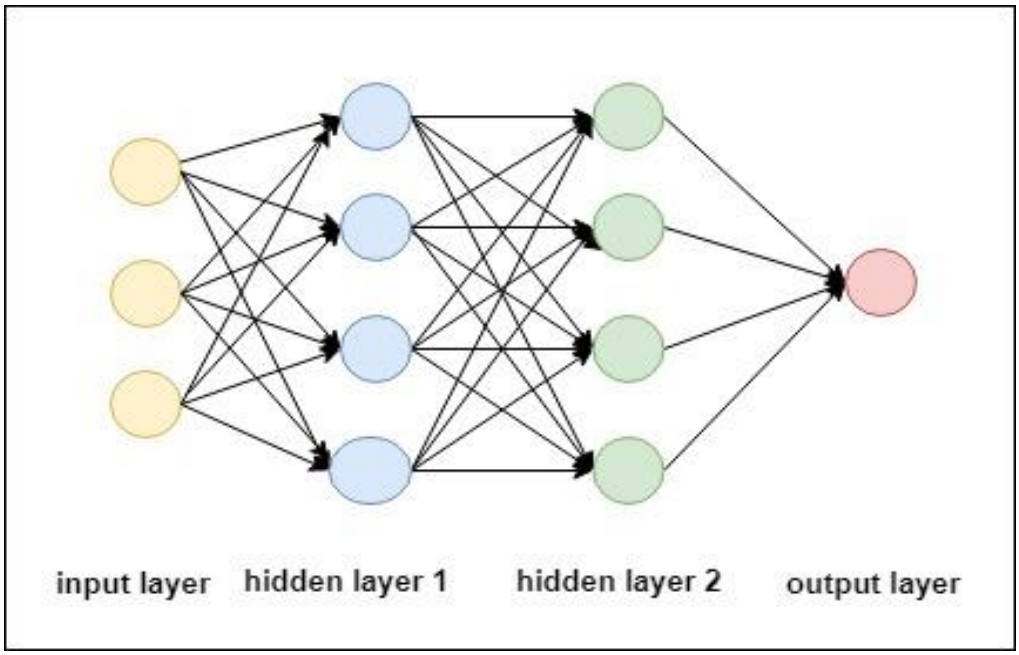


Figure 3. 1: The Structure of MLPNNs

MLPNNs has two phases: forward and backward propagation. In the forward phase, the output is predicted and the error is calculated and sent back to the backward prorogation phase. During the backward propagation, the calculated error is propagated back through the network to adjust the weights and reduce the error in the output layer.

The training process of the MLPNNs is mapping the input to the corresponding output. It begins with providing input and initial weights to the MLPNNs then adjust the weights to minimize the error between the desired and actual output of the network. The output of the MLPNNs is the weighted sums of the inputs which calculated using the following equation:

$$Y_{ij} = w_{ij}.x_i \qquad 3.4$$

Where$w_{ij}$: is the connection weight between the *i*th node in the input layer and the *j*th node in the hidden layer, and $x_i$: is the *i*th input.

To stop the training process, there is a certain threshold θ is set depends on the error of the MLPNN which represents the difference between the desired and actual output. The error is calculated using the following equation:

$$MSE = \frac{1}{2}\sum_{i}^{n}(y_d - y_i)^2 \qquad 3.5$$

The training process continues to tune the weights and minimize the error to be small enough regarding θ. The weights updated using the following equation:

$$\Delta w_{i+1} = \alpha.\mathrm{E}.x_i \qquad 3.6$$

### 1.3.2 Genetic Algorithms

Genetic algorithms (GAs) are one of the most popular variants of evolutionary algorithms (EAs), it's a search method based on natural selection and recombination, which were invented by John Holland in the 1960s [40]. GAs has the ability to give good solutions in reasonable amounts of time, but often it requires too much time to find an acceptable or optimal solution for harder

problems. The researcher made a great efforts to make GAs faster, parallel implementations is the most promising choice, multiple genetic algorithms are used to solve the same problem separately and find its solution where each algorithm use different individual for mutation/crossover, and then the best one is selected to be the solution of the problem, this technique improves the performance of the GAs and reduce the computational time because of using multi-processor [41].

The candidate solution in GAs is presented in a chromosome which consists of a number of elements called ‘genes’ that present the problem variables. GAs starts with a random initial population, then the fitness is calculated for each chromosome in the population, were the fitness function determines the goodness of each solution. GAs creates the next generation of the population through selection, mutation, and crossover. Selection: select the best chromosomes after calculating the fitness for each chromosome in the population. Crossover: exchange genes between the selected chromosomes to create the child chromosome. Mutation: randomly changes genes from the selected chromosomes to create the child chromosome with some low probability. The process continues until getting an acceptable solution determined using a certain condition depends on the value of the fitness function, or reaching a specific number of generations. The genetic algorithms flowchart is illustrated in Figure 3.2.

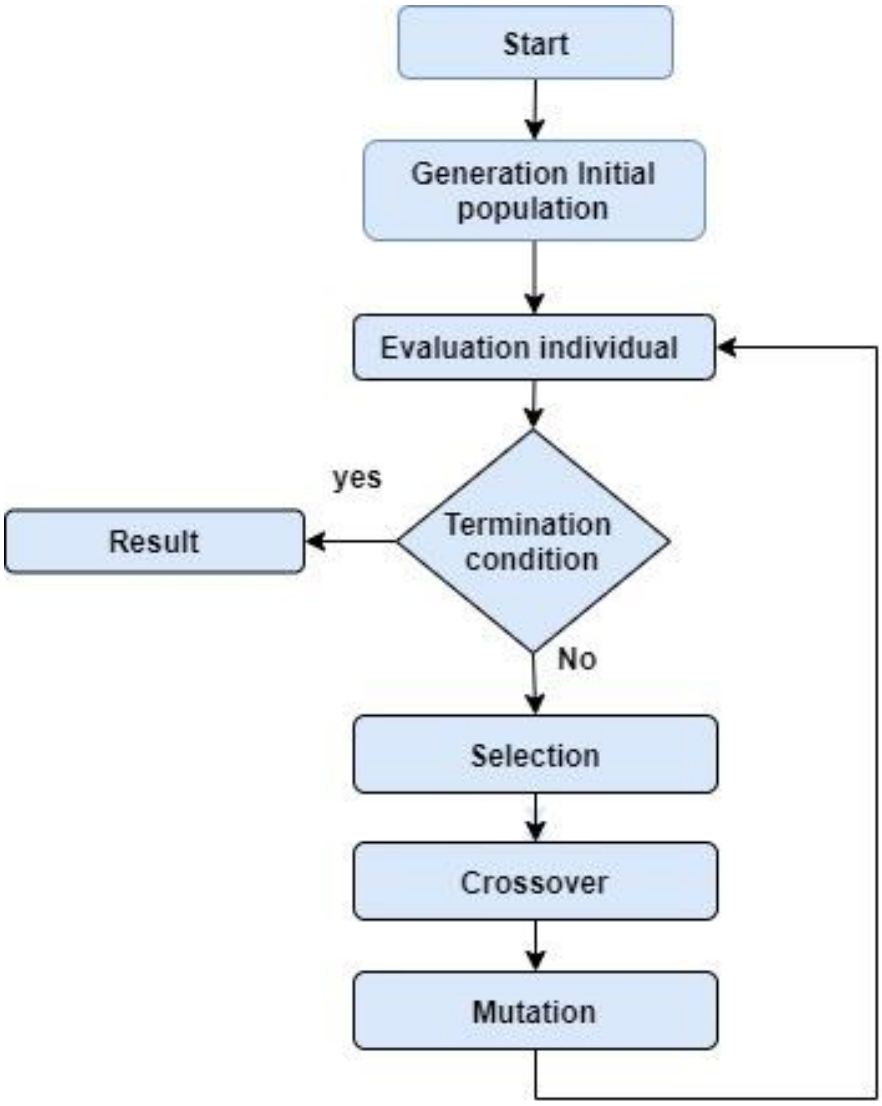


Figure 3. 2: Genetic Algorithm

### 1.3.3 Particle Swarm Optimization (PSO)

Particle swarm optimization is an optimization technique introduced by Kennedy and E Eberhart in 1995. It is based on social behavior such as birds and fish collectively foraging for food. PSO depends on the movement of the particle (Individual) in the search space with a velocity (acceleration) which adjusted according to its movement experience.[41] . Each candidate solution is represented as a particle in the D-dimensional space, the *Ith* particle is represented as

$X_i=(x_{i1}, x_{i2}, x_{i3,\ldots}, x_{id})$. Each particle $i$ having its best previous position that has the best performance which is represented as $Pi=(p_{i1}, p_{i2}, p_{i3,\ldots}, p_{id})$. The index of the particle with the best performance in the population is represented by the variable g. The velocity of the particle i is represented as $V_i = (v_{i1}, v_{i2}, v_{i3},\ldots, v_{id})$. The particle moving in the D-dimensional space using the following formula:

$$v_{id} = v_{id} + \varphi(p_{id-x_{id}}) + \varphi(p_{gd} - x_{id}) \quad 3.7$$

$$x_{id} = x_{id} + v_{id} \quad 3.8$$

.

The steps of the PSO algorithm are as follows [42]:

1- Generating an initial population of particles with random positions and velocities in D-dimension space (the search space).
2- Calculating the fitness function for each particle in *d* variable.
3- Comparing the current fitness value of the particle with its $p_{best}$. If the current fitness value is better than $p_{best}$, then set the $p_{best}$ equal to the current value, and the set *p* equal to the current location in D-dimensional space
4- Comparing the current fitness value with the overall $p_{best}$of the population, if the current $p_{best}$ value is bitter than $g_{best,}$ then set the $g_{best}$ equal to current particle
5- Changing the location and velocity of the particle according to equations3.7 and 3.8.
6- Going to step 2, repeat until meeting the termination criteria which is reaching the predefined number of iteration or getting a good fitness value.

Particle swarm optimization flowchart is illustrated in Figure 3.3

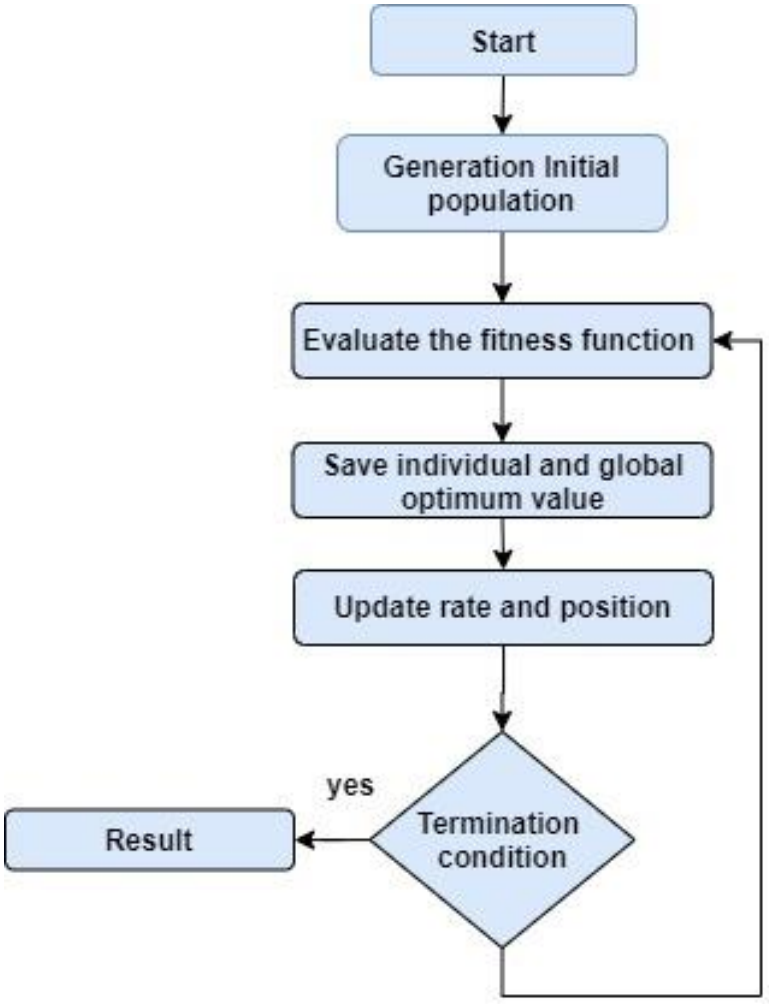


Figure 3. 3: Particle swarm optimization flowchart.

### 1.3.4 Biogeography-Based Optimization Algorithm

The BBO algorithm was first proposed by Dan Simon in 2008. The main idea of the algorithm was inspired by the study of the distribution of biological organisms over time and space. Different ecosystems represented in habitats (islands) are investigated to find the relationship between habitants in terms of emigration, immigration, and mutation [14].

BBO employs number of habitats that represent the candidate solutions, these habitats are analogous to the GAs chromosomes. Each habitat in the BBO algorithm has a number of (Habitants) species that are similar to GAs genes, which is used to present the problem variables. In addition, the Habitat Suitability Index (HSI) indicates the goodness of the solution which is similar to fitness function in GAs, habitats with high HSI have a good solution while habitats with low HSI have a poor one. The algorithm determines the number of Elites (best habitats) for the next generation depends on the HSI value.

The habitats evolve over time based on the following three rules:

- Habitats with a high HSI have a large number of species (Habitants) and more likely to emigrate to Habitats with low HSI.
- Habitats with low HSI have a small number of species (Habitants) and more likely to immigrate species (Habitants) from Habitats with high HSI.
- Habitats may have changes occurs in their species (Habitants) suddenly due to apparently random event regardless of HSI value.

These concepts lead to achieving a balance between different geographical regions; the BBO algorithm uses this concept to improve the HIS of different habitats. Which results in improve the initial random habitats of the problem.

BBO starts with random initial habitats that consist of number habitants that represent the problem variables, each habitat represents a candidate solution of the problem, and each one has a different its immigration, emigration, and mutation rate. The habitats emigrate, immigrate, and mutate their habitants using the following equations:

$$\mu_k = \left(\frac{E \times k}{N}\right) \qquad 3.9$$

Where$\mu_k$ is the emigration rate, *E* is the maximum emigration rate, k is the number of habitants in the current habitat, and N is a maximum number of the habitants which allowed to be in the habitat and it's determined by HSI.

$$\lambda_k = I\left(\frac{1-k}{N}\right) \quad 3.10$$

Where $\lambda_k$ is the immigration rate, *I* is the maximum immigration rate, k is the number of habitants in the current habitat, and N is a maximum number of the habitants which allowed to be in the habitat and it's determined by HSI.

$$m(k) = m_{max}\left(\frac{1-P_k}{P_{max}}\right) \quad 3.11$$

Where m(*k*) is the mutation rate, $m_{max}$ is the maximum mutation probability defined by the user, $P_k$ is the mutation probability for the current habitat, and $P_{max}$=argmax($P_k$), k=1,2,3,…,N.

The general steps of the BBO algorithm are as follows:

1- Generating an initially random set of habitats.
2- Calculating the HSI value for each habitat.
3- Updating the emigration, immigration, and mutation rate for each habitat according to the HSI value.
4- Modifying (Emigrate and immigrate) the habitats according to emigration, immigration rates.
5- Selecting number of habitats and mutate some of their habitants according to mutation rate.
6- Saving elite habitats for the next generation.
7- Going to step 2, repeat until meeting the termination criteria which is a pre-defined number of iteration or getting an acceptable solution.

Elitism is used to prevent immigration from corrupt the best solution when done by saving a predefined number of best solutions at each iteration. Biogeography-Based Optimization flowchart is illustrated in Figure 3.4.

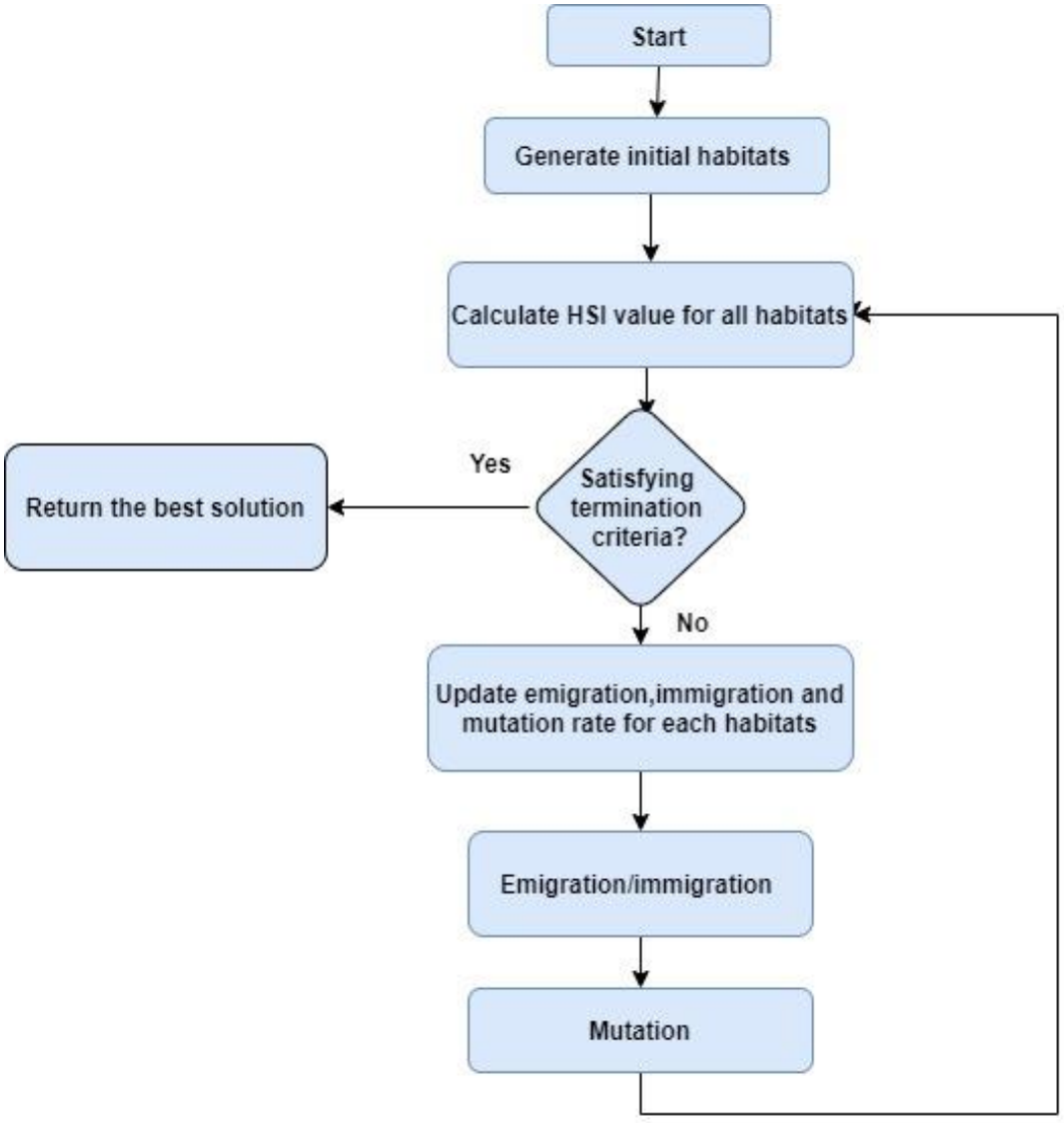


Figure 3. 4: Biogeography-Based Optimization Algorithm

### 1.3.5 Genetic Algorithm -Biogeography-Based Optimization Based Neural Network

Many Evolutionary algorithms have employed to optimize ANNs parameters and find the optimal weights to achieve a better performance of the networks [13]. BBO, PSO, and GA are

some of optimization algorithms that applied on machine learning to train the neural networks and classify datasets [43] [44]. The proposed model (GAsBBO-MLPNNs) combined two optimization algorithms which are GAs [15] and BBO [14] with a multilayer perceptron neural network to improve the Diagnosis of the Heart disease. The proposed algorithm (BBO-MLPNNs) is taking advantage of both GAs and BBO to training the multilayer perceptron neural network and adjust the weights of the networks.

GA recombines different individuals in the population and explicitly using a selection operation to create the solutions. While the BBO algorithm dose not recombine the individual, and its solution improved and maintained from one iteration to the next by migration habitants. [45] For that, GAs was used to generate a set of solutions in order to use them as initial population for the model, then BBO algorithm was used to maintain and improve the solution to find the optimal weight and basis for the network.

The proposed algorithm (GAsBBO-MLPNNs) is taking advantage of both GAs and BBO into the training process. A stopping criterion is set for GAs which is a maximum number of generations. After that, the Best population of GAs generations is set as an initial population (Habitats) for BBO which will again search for the best solution (weights and biases). The BBO stopped after a certain MSE or a maximum number of generations. Figure 3.4 illustrates the steps of the GAsBBO-MLPNNs algorithm.

The general steps of the GAsBBO-MLPNNs algorithm are described in the following steps:

1. Initialization of the GAsBBO-MLPNNs parameter. This includes a) Determination of Crossover probability, Mutation probability, Number and the size of the population; b) creating a random initial population with determine the weights and biases of the network; c)

determining the maximum number of generation for GAs; d) Number of initial population of BBO algorithm.

2. Calculating the fitness for each chromosome using the feed-forward networks (MSE).
3. Creating new generation of population through selection, crossover, and mutation operations.
4. Saving the best chromosome of the population in buffer.
5. Going to 2, repeat until reach stopping criteria which is the maximum number of generation.
6. Initializing the BBO habitats with the GAs best chromosomes from the saved buffer.
7. Calculating the Habitat Suitability Index (HSI) for each habitat.
8. Updating the emigration, immigration, and mutation rate for each habitat
9. Modifying (Emigrate and immigrate) the habitats according to emigration, immigration rate.
10. Selecting number of habitats and mutate some of their weights according to mutation rate.
11. Selecting the elite to prevent the emigration, immigration, and mutation operation from corrupt them in the next generation.
12. Going to step 7, and repeat the process until satisfying the termination criteria.

### 1.3.6 General Method Procedure

The general procedure that was used in performing GAsBBO-MLPNNs on the Z-Alizadeh Sani data set is shown in Algorithm 1and is illustrated by Figure 3.5

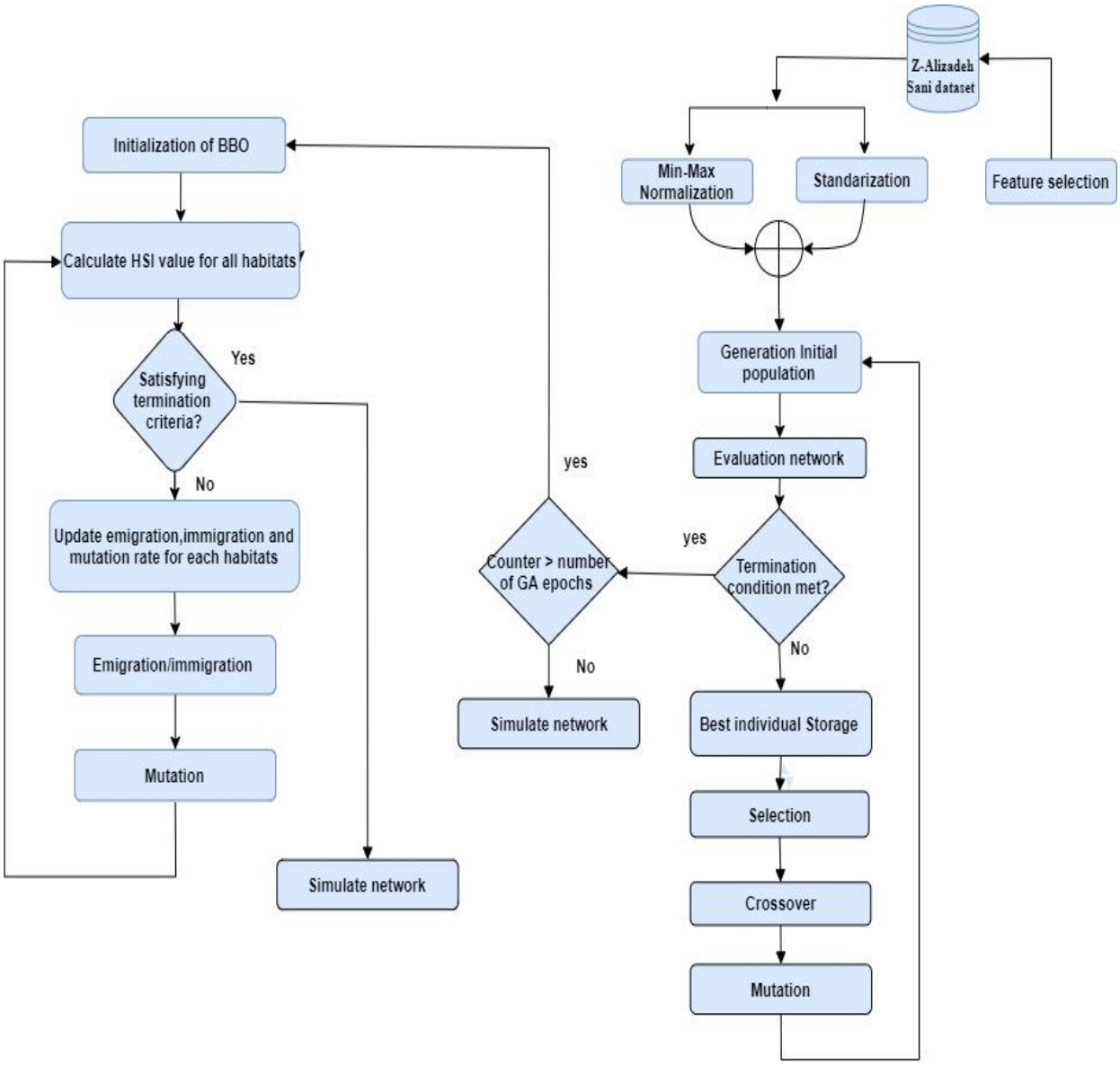


Figure 3. 5: Genetic Algorithm-bio geographical based optimization neural network

**Algorithm 1: Genetic Algorithm-bio geographical based optimization neural network**

**Input:** Dataset, cost function, number of neurons in the hidden layer, population size, crossover probability, mutation probability, Emigration probability μ, Immigration probability λ, keep_rate, Bests, BestCounter;

**Output:** Model Solution;

**Data Preprocessing:**

```
// Converting nominal feature into numerical
for feature in dataset do
    if is_nominal(feature) then
        uniqueItemsList ← uniqueItems(feature);
        for i =1 to size(feature) do
                index ← compare&finduniqueItemsList( feature[variable]);
               numfeature[variable] ← index;
         feature =← numfeature;
    end if;

// normalize the dataset features between negative 1 and 1
for feature in dataset do
    for an item in feature do
        feature[item] ← 2 (feature[item] − min(feature)) / (max(feature) − min(feature)) − 1
```

**Main:**

```
//Genetic algorithm
// initialization
BestCounter=1;
ne ← α ∗ keep_rate;
  for i=1 to size(population) do
    for j=1 to size(chromosome) do
        chromosome[j] ← random();
  for i=1 to δ do /* δ is number of iteration */
    fitness(population[i]) ← MSE();
  // Elitism best solution
  ne ← α ∗ keep_rate;
  population(1) ← best(ne); /* select the best solution and save it in the population1 */
 // Crossover
 nc← (α − ne)/2;
  for i=1 to nc do
    Index1 ← RouletteWheelSelection(population);
    Index2 ← RouletteWheelSelection(population);
    Pa ← population(index1);
    Pb ← population(index2);
    Pc, Pd ← Crossover(Pa, Pb); // crossover by certain probability
    Population2 ← Pc, Pd
 // Mutation
  for j=1 to nm do
    Index ← RouletteWheelSelection(population2);
    Pa ← population2(index);
    Pa ← Mutation(Pa, mu); // mutate the population Pa with probability mu
```

```
      Population2 ← Pa
  //updating population
   Population=population1+population2;
//save best population
SortCost(population); //sort population depending on their cost where the best population stored in population (1) //
BestSol ← population [1];
BestCost ← Cost(BestSol);
//save best population in Best array
Best(BestCounter) ← BestSol; /* save best population of each iteration in Best matrix*/
BestCounter ← BestCounter + 1;

//Bio-geographical based optimization
//initialization
  GABest ← Best; /*get bests matrix from GA algorithm* /
 for i=1 to size(population) do
    population[i] ← GABest[i];
 BestSol ← population[1];  // Best sol ever found
   for i=1 to δ do /* δ is number of iteration  */
       fitness(population[i]) ← MSE();
      // Elitism best solution
       ne ← α ∗ keep_rate;
       population(1) ← best(ne); /* select the best solution and save it in the population1 */
       // compute immigration rate and emigration rate for each habitat based on HIS
       for i=1 to size(population) do
           for k=1 to length(habitat) do
               if randomNum< λ(i)= do
                  z ← H(i);
                  H(j) ← RouletteWheelSelection(μi);
                  z(k) ← Hj(k);
               end if;
               // mutation
               if(randomNum <=Pmutation) do
                 z ← Mutation(z, Pmutation);
           H(i) ← z;
     SortCost(population); //sort population depending in their cost where the best population stored in population
(1) //
      BestSol ← population [1]; // keep
      BestCost ← Cost(BestSol);
     //updating population
     Population=population + population (1)
// Store Best Cost Ever Found to use it as weight in neural network
BestSol ← population [1];
BestCost ← Cost(BestSol);
NetWieghts ← BestSol;
TrainingOutput ← Net(input);
TestingOutput ← Simulate(Net, TestInput);
```

The GAs algorithm consists of *n*, *n*= [1, 2, …, *n*] individuals which represents the candidate solutions of the problem. Each solution has a set of properties that can be altered and mutated based on a predefined probability. The BBO algorithm consists of *n*, *n*= [1, 2, …, *n*] habitats which represent the candidate solutions of the problem. Each solution has a set of properties that can be migrated, immigrated, and mutated. The GAsBBO-MLPNNs starts with a set of randomly generated individuals for GAs, where each individual represents a set of weights for the network. Then the GAs modify the initial population to form a new population using selection, mutation and crossover operations to minimize the classification error. The new population is the weights that will be used to train the GAsBBO-MLPNNs again, this process continues until the minimum error reaches or after a maximum number of generations. If the GAs process ends without reach the minimum error, then the Best population of GAs generations is used as initial habitats for the BBO algorithm. BBO modifies the initial habitats to form a new population according to emigration, immigration, and mutation rates which determined after calculating the HSI value for the habitats in each iteration. The new habitats will be used to train the GAsBBO-MLP until they reach a maximum number of generations or a minimum error. The output of the neurons in the neural network is calculated using the formula 3.4.

The hidden and output layers have to apply an activation function to calculate and pass the output of neurons. In [46], some of the activation functions used for training the neural networks. The activation functions are chosen according to the problem to be solved and the neural network model. The step activation function is the most wield activation function applied in pattern recognition and classification problem [46]. For the GAsBBO-MLPNNs model, the sigmoidal activation function in equation 3.12 is used to calculate the hidden layer of the NNs and the step activation function in equation 3.13 to classify the final output.

Step activation function:

$$Y = \frac{1}{1 + e^{-x}} \quad 3.12$$

Sigmoidal activation function:

$$Y = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases} \quad 3.13$$

### 1.3.7 Neuro-Fuzzy Expert System

Neuro-Fuzzy is a hybrid artificial intelligence technique that combines fuzzy logic and artificial neural networks to generate a fuzzy rule from a given input-output dataset. ANFIS is a Neuro-fuzzy system which was proposed by janj in 1993 [47]. Many models of the Neuro-Fuzzy system were suggested in [48]. This combination can remove the limitations of each model and take advantage of the strength of each of them, where the fuzzy logic is good at giving inexact reasons and explaining decision, while the neural network is good at classification and pattern recognition. Neuro-Fuzzy system uses neural networks to building and reaching the rule, and using fuzzy logic to making decisions. [49]

The structure of the ANFIS (adaptive neuro-fuzzy inference system) model is similar to a multi-layer neural network. ANFIS model consists of five layers, each layer is associated with a particular step in fuzzy logic. The layers of the ANFIS model is illustrated in figure 3.6.

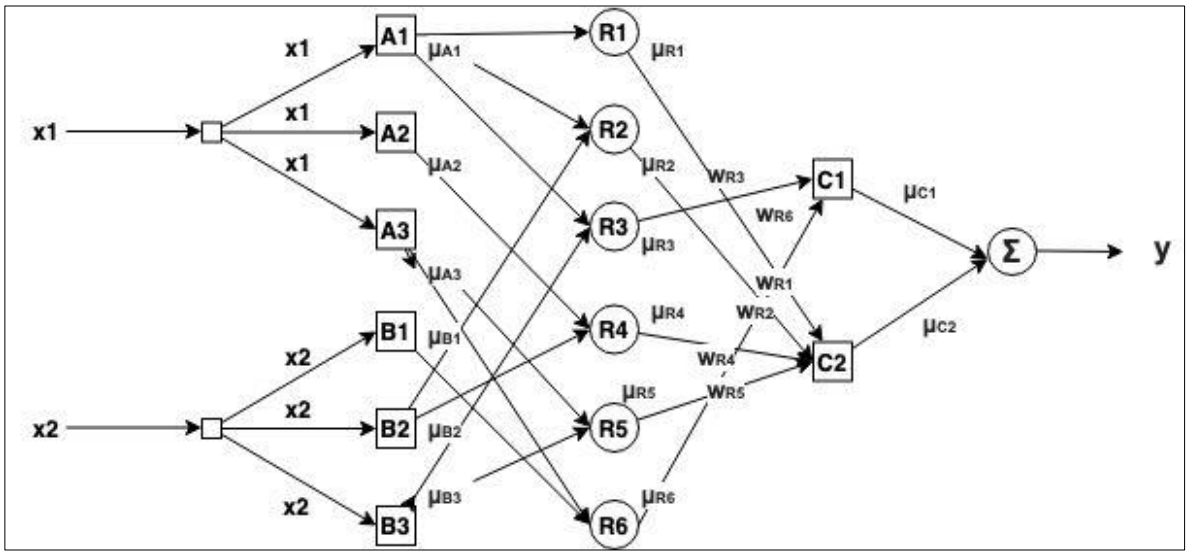


Figure 3. 6: The design of an adaptive Neuro-Fuzzy inference system model.

*Layer 1:* The input layer, each neuron in this layer represents an input variable which transmits the crisp value to the next layer.

*Layer 2:* The fuzzification layer or input membership, neurons in this layer receives a crisp value from the input layer and determine the degree to which neuron's fuzzy set this input belongs.

*Layer 3:* Fuzzy rule layer, each neuron in this layer represent a single fuzzy rule. Neurons in the fuzzy rule layer receive input from the fuzzification layer that represents the fuzzy sets.

*Layer 4:* Output membership layer, this layer combines the values that revised as input the corresponding fuzzy rule neurons, each neuron represents a fuzzy set that consequent of the fuzzy rule layer.

*Layer 5:* Output layer or de -fuzzification layer, a neuron in this layer combine the input from the output membership layer and transform the result to crisp values, each neuron in this layer represents a single output.

In this thesis, we use an Adaptive-Network-based Fuzzy Inference System (ANFIS) [47] to build an intelligent Decision Support Model for diagnosis Heart disease. ANFIS is an impeded tool in MATLAB which used to generate rules from input-output pairs.

## 1.4 Metrics Selection

There are several metrics associated with class "pattern recognition and classification" and statistically measure its performance [50]. This thesis will focus on the following metrics: Confusion matrix, True positive (TP), False positive (FP), False negative (FN), True negative (TN), Accuracy, Sensitivity (Recall), and Specificity.

In the following paragraphs, the definitions of these terms according to heart disease diagnose the problem are:

**TP:** The number of samples correctly categorized as CAD.

**FP:** The number of samples incorrectly categorized as CAD.

**TN:** The number of samples correctly categorized as Normal.

**FN:** The number of samples incorrectly categorized as Normal.

**Misclassification (1-Accuracy)**: The number of samples that categorized incorrectly.

**Confusion matrix:** also known as an error matrix, is a table that used to view the result of the classification model. The table consists of two-diminution, each row represents the predicted class values, and each column represents the Actual class values. It used to compute most of the performance measures. The following table describes the confusion matrix for heart disease to diagnose the problem.

Table 3. 2: Confusion matrix description for heart disease diagnose problem

| | | Predicted Classes | | |
|---|---|---|---|---|
| | | CAD | Normal | Total |
| Actual Classes | CAD | TP | FP | TP+FP |
| | Normal | FN | TN | FN+TN |
| | Total | TP+FN | FP+TN | TP+FP+FN+TN |

**Accuracy:** The main metric that used to measure the performance in class pattern recognition and classification, which represented by the following formula:

$$Accuracy = \frac{\mathrm{TP} + TN}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \quad 3.\ 14$$

**Sensitivity or Recall:** The percentage of records that classified correctly as CAD to all records that classified as CAD.

It is the percentage of records that are predicted to certain class correctly to all records predicted in that class. It is calculated using the following equation:

$$Recall\ = \frac{TP}{TP + F\mathrm{N}} \quad 3.\ 15$$

**Specificity:** The percentage of records correctly predicted as normal to all records predicted in Normal class.

$$Specificity = \frac{TN}{TN + \mathrm{FP}} \quad 3.\ 16$$

**Precision:** The percentage of records correctly predicted as CAD to all records predicted in CAD class.

$$Precision = \frac{TP}{TP + \mathrm{FP}} \quad 3.\ 177$$

**G-mean** [51]: is the cost function constructed based on g-mean of specificity for normal class and the sensitivity of the hostile class, it used to measure the balance between classifications. The G-mean calculated using the following formula:

$$\text{G-mean} = \sqrt{Sensitivity * specificity} \quad 3.188$$

**F-measuring** [51]: it measures the balance between precision and sensitivity (recall). The F-measure calculated using the following formula:

$$\text{F-measuring} = \frac{2 * recall * precision}{recall + precision} \quad 3.199$$

# Chapter 4

# Experiments and Results

## 4.1 Experiments and Results

The proposed was evaluated by applying it to the Z-Alizadeh Sani data set that contains 303 records. The GAsBBO-MLPNNs, BBO-MLPNNs, and PSO-MLPNNs algorithms were used to create class classification on the dataset. Before stating in describing the algorithms, it is determined the best data normalization methods to use them later in the experiments. A hybrid system that combines GAs-BBO, BBO and PSO with neural networks was used to build pattern recognition and classification model as Heart Disease Diagnosis system. The performance of GAsBBO-MLPNNs, BBO-MLPNNs, and PSO-MLPNNs depends on number of iteration (N), number of the neurons in the hidden layers (L), the activation function of the hidden layers where sigmoidal activation function was used, and the parameters of each optimization algorithm that have an important role in improving the algorithms performance which is depend on the dataset used.

The cross-validation method called K-Fold Cross-Validation [52] was used to build and evaluate the models. Based on the K-Fold Cross-Validation method the data partitioned into k equally sized folds. For each fold $i$, the data divided into k partition, the $i^{th}$the fold is used for testing while the remaining folds are used for training the model.

In this thesis, Tenfold cross-validation is used to evaluate the models where 90 percent of the dataset used for training the model and the remaining 10 percent of the dataset used to perform the testing phase. The overall accuracy was used in the parameter optimization phase, while the performance of the proposed models were measured using the overall accuracy, F-score, confusion matrix, overall accuracy, Sensitivity (Recall), and Specificity.

## 4.2 Standardization Method Selection Considerations

Min-Max normalization and standardization cleaning methods were applied on the Z-Alizadeh Sani dataset to select one of them. The experiments were performed twice on the dataset. At first, the Min-Max normalization method was used for cleaning the data, in the second time standardization method was used for cleaning the data.

These experiments were performed on Z-Alizadeh Sani dataset using MATLAB pattern recognition and classification tool, the result showed that the Min-Max normalization got better performance in term of accuracy. So, the Min-Max normalization method was used to clean the data of the experiment. Figure 4.1 shows the effect of the standardization and Min-max Normalization on the performance of MATLAB pattern recognition and classification tool.

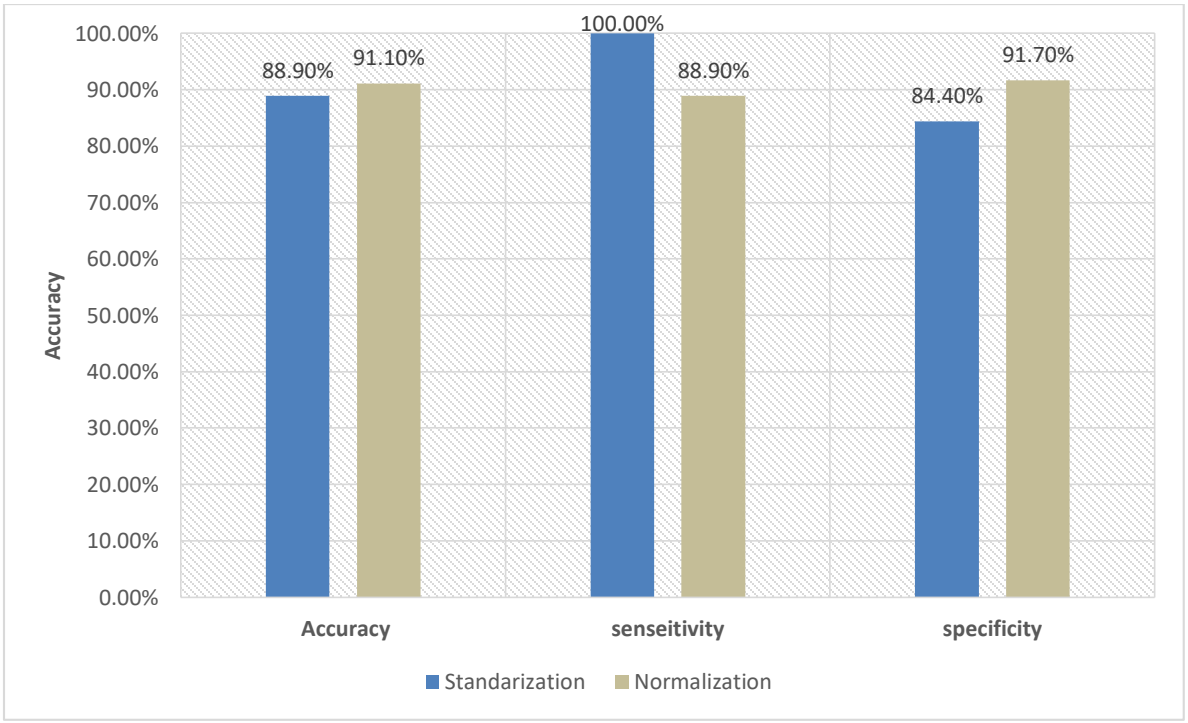


Figure 4. 1: The effect of the standardization and Min-max Normalization on the performance of MATLAB pattern recognition and classification tool.

## 4.3 Z-Alizadeh Sani dataset Experiments

BBO-MLPNNs, PSO-MLPNNs, and GAsBBO-MLPNNs were used to Heart disease detection and prediction on the Z-Alizadeh Sani dataset. The best BBO-MLPNNs, PSO-MLPNNs, and GAsBBO-MLPNNs models with certain L, and N, and P were proposed to address the handle of this problem. The following sub-sections describe the several experiments that performed to find the optimized models.

### 4.3.1 PSO-MLPNNs Experiments on Z-Alizadeh Sani Dataset

The PSO-MLPNNs algorithm was used to perform an experiment on the Z-Alizadeh Sani dataset. To improve the performance of the algorithm three parameters were adjusted: the first parameter was the number of neurons in the hidden layer (L). The second parameter was the number of Iteration (N).The third parameter was the population size (P).

The goal of the experiments performed using the PSO-MLPNNs model was to find the best L, and N, and P that will be used to build PSO-MLPNNs Heart disease detection and prediction solution. The combination of these parameters is listed in table 4.1. The experiment that was performed to achieve this goal is:

**Test**: PSO-MLPNNs were built to find the best L, and N, and P that achieved the optimized model. Table 4.1 includes the result of these models. It shows that the best L value was 10, N value was 200, and P value was 60. Figure 4.2 shows the accuracy of the PSO-MLPNNs model related to the number of iteration where the best accuracy was achieved with N=200. Figure 4.3 shows the relationship between the accuracy and the number of neurons where the best L value was 10.

**Summary:** The best model for PSO-MLPNNs was L=10, N=200, and P=60. Its assessment was88.8percent in the overall accuracy and 86.43, 80.79, 75.90, 89.20, 72.80, 90.23 percent in G-mean, F-measure, Sensitivity, Specificity, Precession, and NP.

Table 4. 1: PSO-MLPNNs Models Results (A).

| PSO-MLPNNs Models Results | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Average-fold PSO-MLPNNs | | | |
| **Training Accuracy** | | | **Test Accuracy** | | | **G-mean** | | **F-Measure** | |
| **92.64%%** | | | **88.80%** | | | **86.43%** | | **80.79%** | |
| N | L | Training Accuracy | Testing Accuracy | Sensitivity | Specificity | Precession | Negative Prediction | F-Measure | G-mean |
| 100 | 10 | 90.80% | 85.10% | 75.90% | 89.20% | 72.80% | 90.23% | 74.32% | 82.28% |
| 150 | | 91.75% | 86.40% | 75.55% | 91.51% | 79.32% | 89.34% | 77.39% | 83.15% |
| 200 | | 92.64% | 88.80% | 81.01% | 92.21% | 80.57% | 92.09% | 80.79% | 86.43% |
| 100 | 20 | 92.45% | 82.80% | 69.47% | 89.46% | 73.76% | 86.60% | 71.55% | 78.83% |
| 150 | | 92.58% | 85.80% | 75.34% | 90.17% | 75.01% | 90.25% | 75.17% | 82.42% |
| 200 | | 92.75% | 88.40% | 78.63% | 92.49% | 81.68% | 90.74% | 80.13% | 85.28% |
| 100 | 35 | 91.86% | 85.50% | 75.38% | 91.56% | 79.60% | 88.39% | 77.43% | 83.08% |
| 150 | | 92.24% | 87.40% | 81.67% | 91.01% | 77.24% | 92.51% | 79.39% | 86.21% |
| 200 | | 93.09% | 88.70% | 80.89% | 92.73% | 81.82% | 91.59% | 81.35% | 86.61% |

Table 4.1: PSO-MLPNNs Models Results (B).

| **Accuracy** | | 93.5% | | | | | | Best-fold PSO-MLPNNs | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precession | Negative Prediction |
| 100 | 10 | 90.5% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 150 | | 93.4% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 200 | | 93.8% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 100 | 20 | 93.4% | 80.6% | 8 | 1 | 17 | 5 | 61.5% | 94.4% | 88.9% | 77.3% |
| 150 | | 93.8% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 200 | | 93.4% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | 35 | 92.3% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 150 | | 92.6% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 200 | | 90.8% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table 4.1: PSO-MLPNNs Models Results (C).

| **Accuracy** | | 82.8% | | | | | | Worst-fold PSO-MLPNNs | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precession | Negative Prediction |
| 100 | 10 | 92.0% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 150 | | 92.0% | 79.3% | 5 | 3 | 18 | 3 | 62.5% | 85.7% | 62.5% | 85.7% |
| 200 | | 94.2% | 82.8% | 6 | 2 | 18 | 3 | 66.7% | 90.0% | 75.0% | 85.7% |
| 100 | 20 | 92.7% | 75.9% | 5 | 3 | 17 | 4 | 55.6% | 85.0% | 62.5% | 81.0% |
| 150 | | 94.2% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 200 | | 93.4% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 100 | 35 | 89.8% | 75.9% | 6 | 2 | 16 | 5 | 54.5% | 88.9% | 75.0% | 76.2% |
| 150 | | 92.0% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 200 | | 92.7% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |

Table 4.1 represents the PSO-MLPNNs Models results. The model achieved the best performance with N=200 and L=10, where the performance parameters of the average folds represented as 92.64%, 88.8%, 86.43%, 80.79% for training accuracy, test accuracy, G-mean, F-measure respectively. While the testing accuracy for the best fold is 93.5%, and for the worst fold is 82.8%.

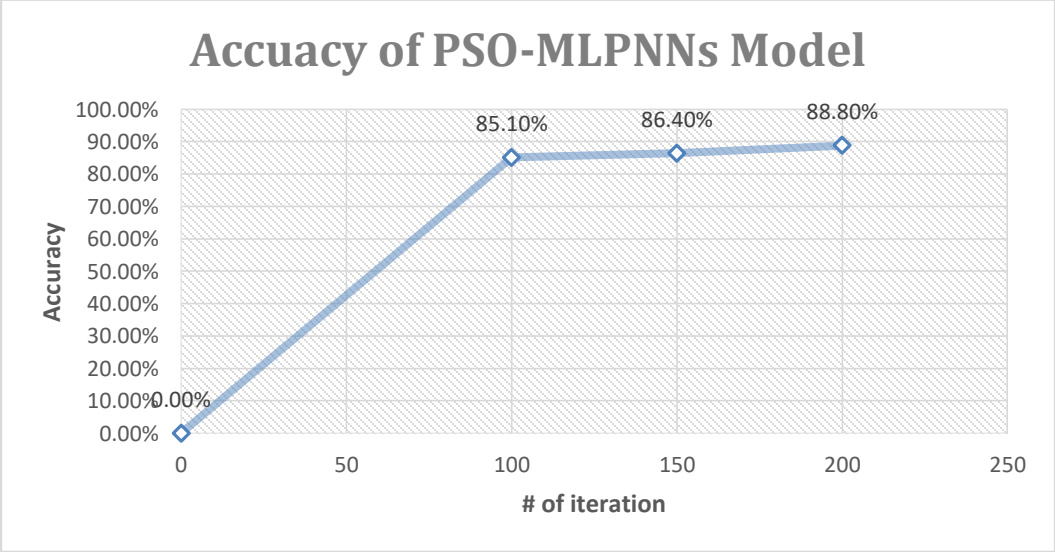


Figure 4. 2: The accuracy of PSO-MLPNNs model related to the number of iterations.

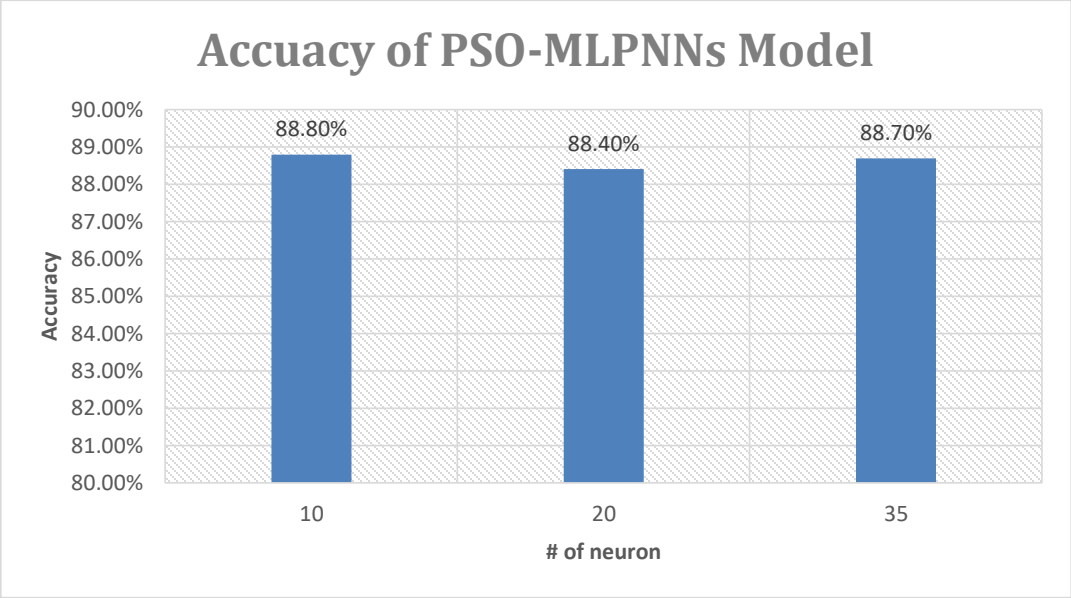


Figure 4. 3: The accuracy of PSO-MLPNNs model related to the number of neurons

Figure 4.2 shows the classification accuracy of the PSO-MLPNNs model on L=10 and N values are [100 150 200]. It shows that the accuracy is increased as the number of iteration N is increased, the model produces an accuracy of 85.10% on N=100, 86.40% on N=150, and the highest classification accuracy 88.80% using N=200.

Figure 4.3 shows a comparison between the classification accuracy of the PSO-MLPNNs model on N=100 and L values are [10 20 30]. The model produced almost equal results using a different value for L, where the best accuracy 88.80% using L=10, compared with 88.40% using L=20 and 88.70% using L=35.

### 4.3.2 BBO-MLPNNs Experiments on Z-Alizadeh Sani dataset

The BBO-MLPNNs algorithm was used to perform experiments on the Z-Alizadeh Sani dataset. To improve the performance of the algorithm, three parameters were adjusted: the number of neurons in the hidden layer (L), the number of iteration (N) and the population size (P).The goal of the experiments performed using the BBO-MLPNNs algorithm was to find the best L,N, and

P that will be used to build BBO-MLPNNs Heart disease detection and prediction solution. The combination of these parameters is listed in table 4.2. The experiment that was performed to achieve this goal is:

**Test**: several ten-fold cross-validation models of BBO-MLPNNs were built to find the best L and N that achieved the optimized model. Table 4.2 includes the result of these models. It shows that the best L value was 10, the N value was 150, and *P* value was 60. Figure 4.4shows the accuracy of the BBO-MLPNNs model related to the number of iteration which almost stabilized after this number of N. Figure 4.5shows the relationship between the accuracy and the number of neurons where the best L value was 10.

**Summary:** The best model for BBO-MLPNNs was L=10, N=150, and P=60. Its assessment was 93.01percent in the overall accuracy and 92.19, 87.97, 89.74, 81.24, 93.51, 84.73. 91.59 percent in G-mean, F-measure, Sensitivity, Specificity, Precession, NP.

Table 4. 2: BBO-MLPNNs Models Results. (A)

| BBO-MLPNNs result with $L = 10$, $N = 150$, $P = 60$. | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Average result BBO-MLPNNs | | | |
| Training Accuracy | | | Test Accuracy | | | G-mean | | F-Measure | |
| 94.50% | | | 93.10% | | | 92.19% | | 87.97% | |
| N | L | Training Accuracy | Testing Accuracy | Sensitivity | Specificity | Precession | Negative Prediction | F-Measure | G-mean |
| 100 | 10 | 92.89% | 90.03% | 81.24% | 93.51% | 84.73% | 91.59% | 82.95% | 87.16% |
| 150 | | 94.46% | 93.01% | 89.74% | 94.70% | 86.26% | 95.81% | 87.97% | 92.19% |
| 200 | | 94.24% | 92.64% | 88.95% | 95.12% | 87.23% | 95.34% | 88.08% | 91.98% |
| 100 | 20 | 92.64% | 88.34% | 79.01% | 93.07% | 82.65% | 90.71% | 80.79% | 85.75% |
| 150 | | 93.81% | 91.35% | 87.48% | 93.29% | 83.48% | 94.86% | 85.43% | 90.34% |
| 200 | | 93.75% | 91.65% | 82.72% | 96.23% | 90.84% | 92.13% | 86.59% | 89.22% |
| 100 | 35 | 92.72% | 89.05% | 80.08% | 93.19% | 82.79% | 91.67% | 81.41% | 86.39% |
| 150 | | 93.78% | 90.64% | 83.23% | 94.85% | 87.09% | 92.57% | 85.12% | 88.85% |
| 200 | | 93.93% | 91.62% | 87.30% | 94.17% | 85.01% | 94.39% | 86.14% | 90.67% |

Table 4.2: BBO-MLPNNs Models Results. (B)

| Accuracy | | 96.70% | | | | | | Best-fold BBO-MLPNNs | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 95.20% | 93.30% | 9 | 0 | 19 | 2 | 81.80% | 100.00% | 100.00% | 90.50% |
| 150 | | 97.40% | 96.70% | 8 | 1 | 21 | 0 | 100.00% | 95.50% | 88.90% | 100.00% |
| 200 | | 97.80% | 96.70% | 8 | 1 | 21 | 0 | 100.00% | 95.50% | 88.90% | 100.00% |
| 100 | 20 | 91.90% | 90.00% | 8 | 1 | 19 | 2 | 80.00% | 95.00% | 88.90% | 90.50% |
| 150 | | 95.60% | 93.30% | 8 | 1 | 20 | 1 | 88.90% | 95.20% | 88.90% | 95.20% |
| 200 | | 95.20% | 93.30% | 7 | 2 | 21 | 0 | 100.00% | 91.30% | 77.80% | 100.00% |
| 100 | 35 | 94.10% | 93.30% | 7 | 2 | 21 | 0 | 100.00% | 91.30% | 77.80% | 100.00% |
| 150 | | 96.70% | 96.70% | 9 | 0 | 21 | 1 | 90.00% | 100.00% | 100.00% | 95.50% |
| 200 | | 94.50% | 93.30% | 7 | 2 | 21 | 0 | 100.00% | 91.30% | 77.80% | 100.00% |

Table 4.2: BBO-MLPNNs Models Results. (C)

| Accuracy | | 86.20% | | | | | | Worst-fold BBO-MLPNNs | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 90.50% | 82.80% | 7 | 1 | 17 | 4 | 63.60% | 94.40% | 87.50% | 81.00% |
| 150 | | 90.50% | 86.20% | 7 | 1 | 18 | 3 | 70.00% | 94.70% | 87.50% | 85.70% |
| 200 | | 90.50% | 86.20% | 7 | 1 | 18 | 3 | 70.00% | 94.70% | 87.50% | 85.70% |
| 100 | 20 | 91.60% | 79.30% | 6 | 2 | 17 | 4 | 60.00% | 89.50% | 75.00% | 81.00% |
| 150 | | 92.70% | 82.80% | 6 | 2 | 18 | 3 | 66.70% | 90.00% | 75.00% | 85.70% |
| 200 | | 92.30% | 86.20% | 7 | 1 | 18 | 3 | 70.00% | 94.70% | 87.50% | 85.70% |
| 100 | 35 | 93.80% | 82.80% | 6 | 2 | 18 | 3 | 66.70% | 90.00% | 75.00% | 85.70% |
| 150 | | 90.10% | 75.90% | 6 | 2 | 17 | 4 | 60.00% | 89.50% | 75.00% | 81.00% |
| 200 | | 93.80% | 79.30% | 6 | 2 | 17 | 4 | 60.00% | 89.50% | 75.00% | 81.00% |

Table 4.2 represents the BBO-MLPNNs Models results. The model achieved the best performance with N=150 and L=10, where the performance parameters of the average folds represented as 94.5%, 93.1%, 92.19%, 87.79% for training accuracy, test accuracy, G-mean, F-measure respectively. While the testing accuracy for the best fold is96.7%, and for the worst fold is 86.2%.

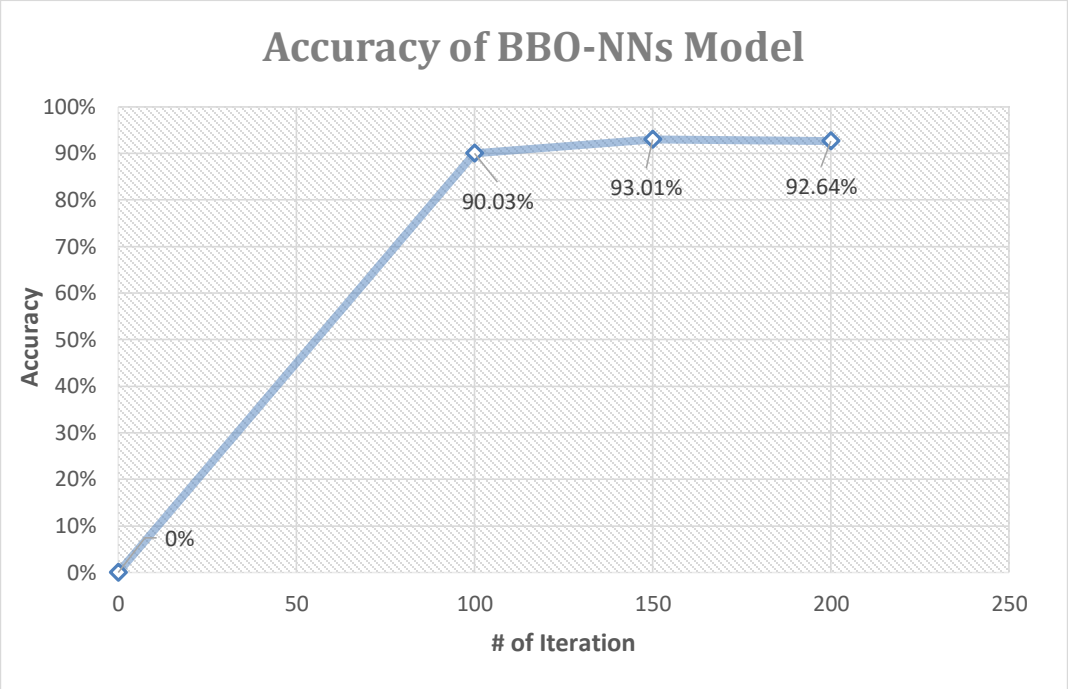


Figure 4. 4: The accuracy of BBO-MLPNNs model related to the number of iteration.

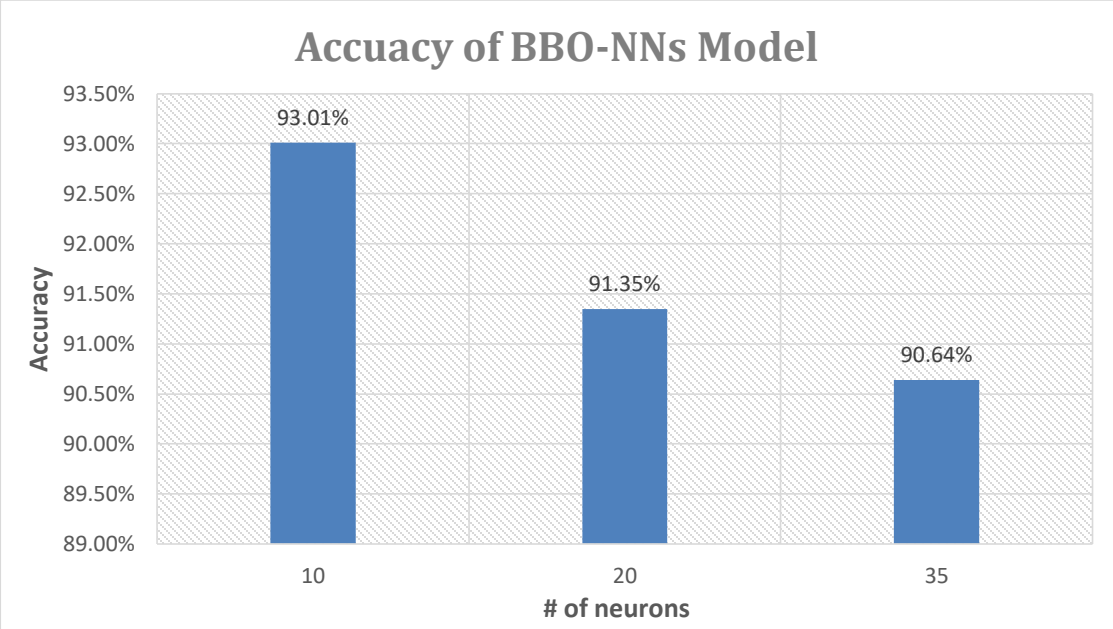


Figure 4. 5: The accuracy of BBO-MLPNNs model related to the number of neurons.

Figure 4.4 shows the classification accuracy of the BBO-MLPNNs model on L=10 and N values are [100 150 200]. It shows that the accuracy was increased until 150 iterations, the model

produces an accuracy of 90.03% at N=100, 92.64% at N=200, and the highest classification accuracy at N=150 which is 93.01%.Figure 4.5 shows a comparison between the classification accuracy of the BBO-MLPNNs model on N=150 and L values are [10 20 30]. The model achieves the best accuracy of 93.01% using L=10, compared with 91.35% using L=20 and 90.64% using L=35.

### 4.3.3 GAsBBO-MLPNNs Experiments on Z-Alizadeh Sani Dataset

The GAsBBO-MLPNNs algorithm was used to perform an experiment on the Z-Alizadeh Sani dataset. To improve the performance of the algorithm three parameters were adjusted: the first parameter was the number of neurons in the hidden layer (L). The second parameter was the number of Iteration (N).The third parameter was the population size (P).

The goal of the experiments performed using the GAsBBO-MLPNNs model was to find the best L, and N, and P that will be used to build GAsBBO-MLPNNs model for Heat disease detection and prediction solution. The combination of these parameters is listed in table 4.3. The experiment that was performed to achieve this goal is:

**Test**: several ten-fold cross-validation models of GAsBBO-MLPNNs were built to find the best L, and N, and P that achieved the optimized model. Table 4.3 includes the result of these models. It shows that the best L value was 10, N value was 100, and P value was 60.Figure 4.6 shows the relationship between the accuracy and the number of iterations of the GAsBBO-MLPNNs model, and figure 4.7shows the relationship between the accuracy and the number of neurons where the best L was 10.

**Summary:** the GAsBBO-MLPNNs was the best model with L=10, N=100, and P=60. Its assessment was94.5percent in the overall accuracy and 95.6, 89.94, 96.4, 94.8, 84.3, 98.6 percent in G-mean, F-measure, Sensitivity, Specificity, Precession, NP.

Table 4. 3: GAsBBO-MLPNNs Models Results (A).

| GAsBBO-MLPNNs result with $L = 10, N = 100, P = 60$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Average result GAs-BBO-MLPNNs | | | |
| **Training Accuracy** | | | **Test Accuracy** | | | **G-mean** | | **F-Measure** | |
| **95.50%** | | | **94.50%** | | | **95.60%** | | **89.94%** | |
| N | L | Training Accuracy | Testing Accuracy | Sensitivity | Specificity | Precession | Negative Prediction | F-Measure | G-mean |
| 50 | 10 | 95.00% | 93.80% | 92.5% | 93.5% | 86.5% | 96.80% | 89.40% | 93.00% |
| 100 | | 95.50% | 94.50% | 96.4% | 94.8% | 84.3% | 98.60% | 89.94% | 95.60% |
| 150 | | 94.70% | 93.10% | 91.0% | 94.3% | 86.7% | 95.90% | 88.80% | 92.70% |
| 50 | 20 | 94.40% | 92.80% | 93.5% | 93.8% | 82.1% | 97.20% | 87.43% | 93.60% |
| 100 | | 94.80% | 92.80% | 90.6% | 91.7% | 85.6% | 95.90% | 88.03% | 91.20% |
| 150 | | 94.40% | 92.50% | 92.0% | 92.3% | 83.4% | 96.80% | 87.49% | 92.10% |
| 50 | 35 | 91.90% | 89.50% | 86.1% | 91.5% | 77.5% | 94.50% | 81.57% | 88.80% |
| 100 | | 92.60% | 90.50% | 91.6% | 91.6% | 76.6% | 96.30% | 83.43% | 91.60% |
| 150 | | 93.50% | 91.50% | 89.3% | 91.4% | 82.1% | 95.90% | 85.55% | 90.40% |

Table 4.3: GAsBBO-MLPNNs Models Results (B).

| **Accuracy** | | **96.8%** | | | | | Best-fold GAsBBO-MLPNNs | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N | L | Training Accuracy | Test Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precession | Negative Prediction |
| 50 | 10 | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 97.5% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 150 | | 98.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 50 | 20 | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 94.1% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 150 | | 94.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 50 | 35 | 93.1% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 100 | | 94.6% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 150 | | 92.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table 4.3: GAsBBO-MLPNNs Models Results (C).

| **Accuracy** | | **90.3%** | | | | | Worst -fold GAsBBO-MLPNNs | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N | L | Training Accuracy | Test Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precession | Negative Prediction |
| 50 | 10 | 92.6% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 100 | | 92.1% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 150 | | 93.6% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 50 | 20 | 88.7% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 100 | | 89.2% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 150 | | 89.7% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 50 | 35 | 86.7% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 100 | | 90.1% | 87.1% | 5 | 4 | 22 | 0 | 100.0% | 84.6% | 55.6% | 100.0% |
| 150 | | 91.6% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |

Table 4.3 represents the GAsBBO-MLPNNs Models results. The model achieved the best performance with N=100 and L=10, where the performance parameters of the average folds represented as 95.5%, 94.5%, 95.60%, 89.96% for training accuracy, test accuracy, G-mean, F-measure respectively. While the testing accuracy for the best fold is96.8%, and for the worst fold is 90.3%.

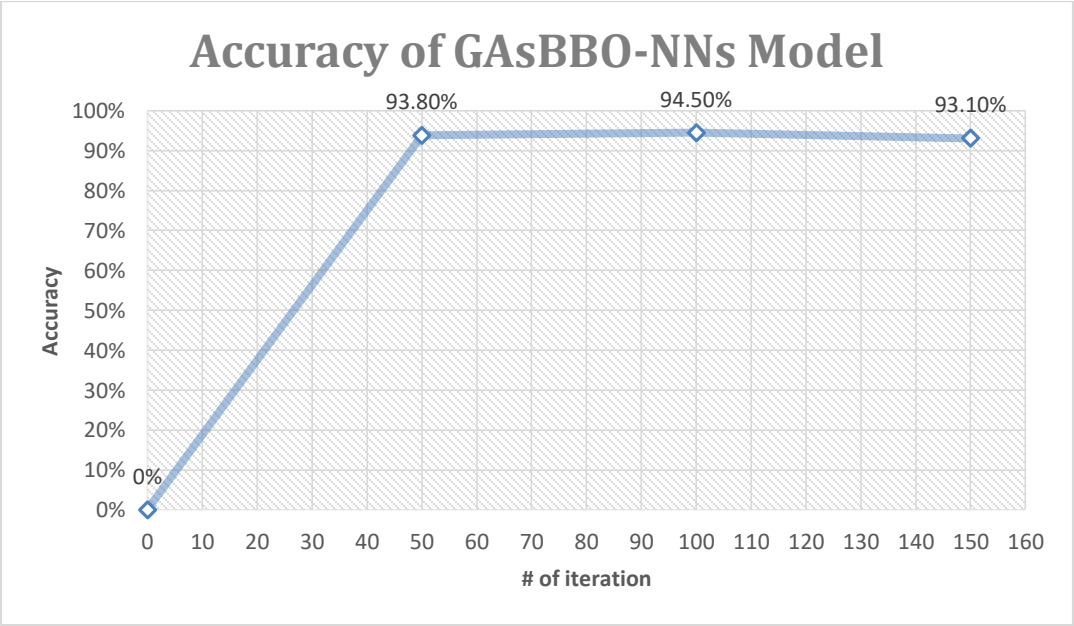


Figure 4. 6: The accuracy of the GAsBBO-MLPNNs model related to the number of iterations.

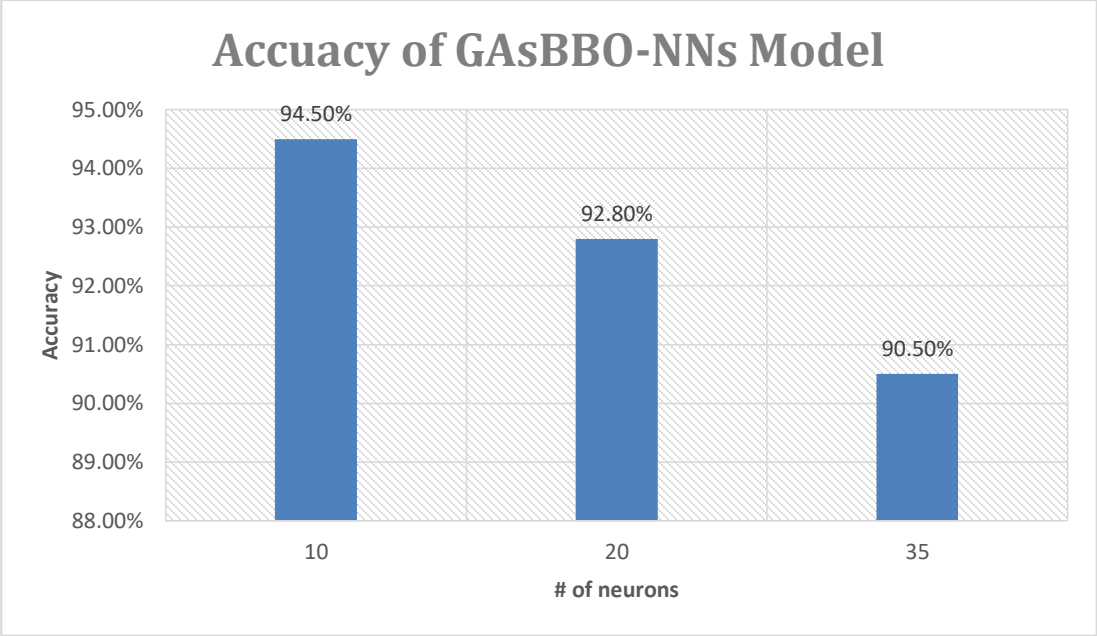


Figure 4. 7: The accuracy of GAsBBO-MLPNNs model related to the number of neurons.

Figure 4.6 shows the classification accuracy of the GAsBBO-MLPNNs model on L=10 and N values are [50 100 150].It shows that the accuracy was increased until 100 iterations, the model produces an accuracy of 90.80% on N=50, 93.10% on N=150, and the highest classification accuracy of 94.50% using N=100.Figure 4.7 shows a comparison between the classification accuracy of the GAsBBO-MLPNNs model on N=100 and L values are [10 20 30]. The model achieves the best accuracy of94.50% using L=10, compared with 92.80% using L=20 and 90.50% using L=35.

## 4.4 Discussion of the Results

GAsBBO-MLPNNs, BBO-MLPNNs, and PSO-MLPNNs model was applied on the Z-Alizadeh Sani data set, where different parameters related to these algorithms were optimized. Table 4.4 shows The List of experiments that were performed on the Z-Alizadeh Sani dataset with the optimized parameters. The experiments show that the GAsBBO-MLPNNs model has a better

performance than BBO-MLPNNs and PSO–MLPNNs with respect to the overall accuracy, G-mean, and F-measure.

Table 4. 4: The List of experiments that were performed on the Z-Alizadeh Sani dataset with the optimized parameters.

| Dataset | Algorithms | The Optimized Parameters, | | | The overall accuracy | F-score |
|---|---|---|---|---|---|---|
| | | # of Hidden Neurons | GAs parameter | # of Iteration | | |
| Z-Alizadeh Sani dataset | GAsBBO-MLPNNs Ten-Fold | 10 | GAs Iteration=60 | 50 | 93.80% | 89.40% |
| | | | | 100 | 94.50% | 89.94% |
| | | | | 150 | 93.10% | 88.80% |
| | | 20 | | 50 | 92.80% | 87.43% |
| | | | | 100 | 92.80% | 88.03% |
| | | | | 150 | 92.50% | 87.49% |
| | | 35 | | 50 | 89.50% | 81.57% |
| | | | | 100 | 90.50% | 83.43% |
| | | | | 150 | 91.50% | 85.55% |
| | BBO-MLPNNs Ten-Fold | 10 | | 100 | 90.03% | 82.95% |
| | | | | 150 | 93.01% | 87.97% |
| | | | | 200 | 92.64% | 88.08% |
| | | 20 | | 100 | 88.34% | 80.79% |
| | | | | 150 | 91.35% | 85.43% |
| | | | | 200 | 91.65% | 86.59% |
| | | 35 | | 100 | 89.05% | 81.41% |
| | | | | 150 | 90.64% | 85.12% |
| | | | | 200 | 91.62% | 86.14% |
| | PSO-MLPNNs Ten-Fold | 10 | | 100 | 85.10% | 74.32% |
| | | | | 150 | 86.40% | 77.39% |
| | | | | 200 | 88.80% | 80.79% |
| | | 20 | | 100 | 82.80% | 71.55% |
| | | | | 150 | 85.80% | 75.17% |
| | | | | 200 | 88.40% | 80.13% |
| | | 35 | | 100 | 85.50% | 77.43% |
| | | | | 150 | 87.40% | 79.39% |
| | | | | 200 | 88.70% | 81.35% |

Table 4. 5: Comparison between our model and the previous work in [18] [28].

| Model | | Accuracy | Sensitivity | Specificity |
|---|---|---|---|---|
| GAs-NNs | | 93.85% | 97% | 92% |
| SMO classifiers | | 92.09% | **97.22%** | 79.31% |
| The proposed | BBO-MLPNNs | 93.01% | 89.74% | 94.70% |
| | GAsBBO-MLPNNs | **94.50%** | 96.40% | **94.80%** |

Several works are published on the Z-Alizadeh Sani dataset, the two closest to this work are used to evaluate this work which referred by [18] [28]. As mentioned in the related works section, the proposed referred by [18] data mining method for diagnosis of coronary artery disease (using SMO algorithm), they create three features to improve the diagnosis accuracy, the proposed referred by [28] hybrid system using GAs and NNs to predict the cardiovascular disease. Figure 4.8 and Table 4.5 show the result of our work compared with both works in [18] [28].

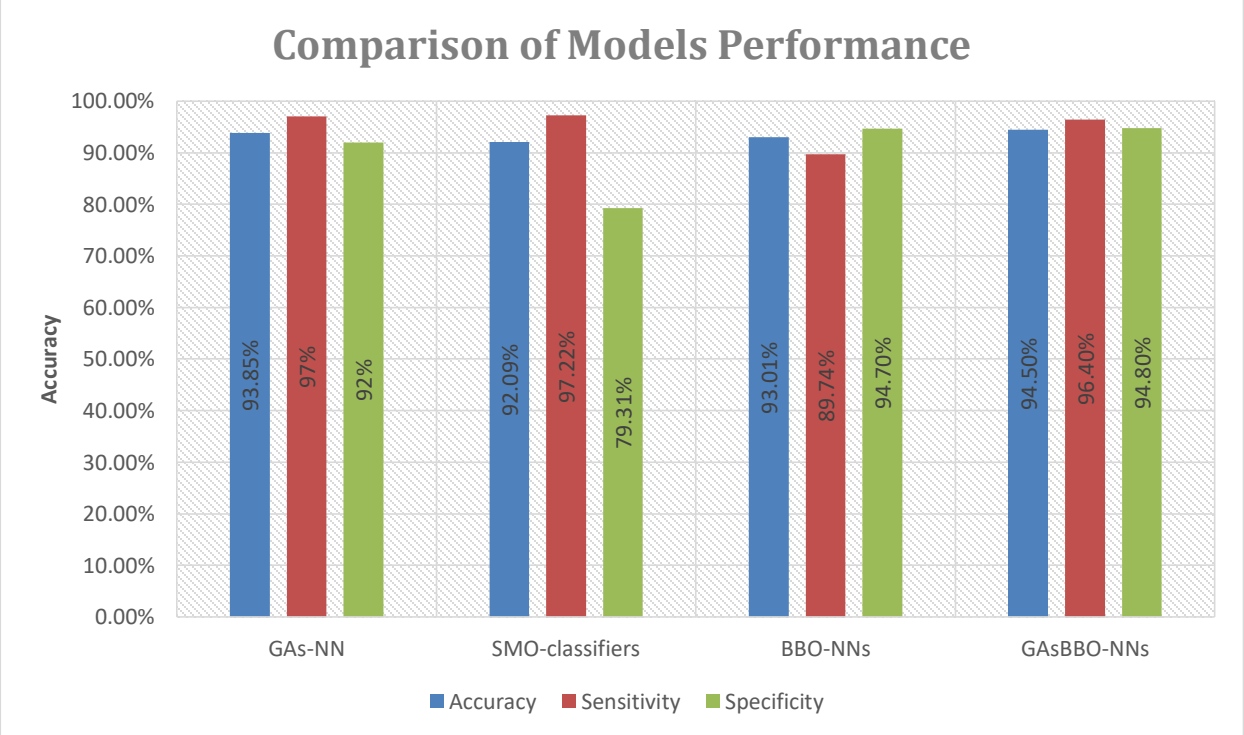


Figure 4. 8: comparison of the performance of the proposed and the previous work

The table shows that the proposed GAsBBO-MLPNNs produce a result of94.5%, 96.4%, 94.8% in Accuracy, Sensitivity, and Specificity respectively. BBO-MLPNNs model produces a result as 93.01%, 89.78%, 94.70% in Accuracy, Sensitivity, and Specificity respectively. GAsBBO-MLPNNs Outperform the GAs-NNs and SMO classifiers in terms of accuracy. Where the accuracy of our ten-fold GAsBBO-MLPNNs was 94.50% vs. 93.85% for GAs-NNs and 92.09% for SMO classifier models. Even the GAs-NNs and SMO classifiers got a better result in sensitivity; our model got a bitter result in terms of accuracy and specificity (94.50% and 94.80%).

## 4.5 Neuro-Fuzzy System Experiment

The proposed was evaluated by applying it to the Z-Alizadeh Sani data set that contains 303 records. The Neuro-Fuzzy Designer Tool was used to create Rule base system on the dataset. It uses neural networks and fuzzy logic to build a Decision Support Model. A MATLAB was used to perform our experiment. The best model with certain N and MF were proposed to find the optimal solution. Table 4.6 describes the parameter used in the experiments and their ranges. The following section describes the experiments that performed to find the best models.

### 4.5.1 Neuro-Fuzzy System Experiments on Z-Alizadeh Sani Dataset

The ANFIS tool was used to perform experiments on the Z-Alizadeh Sani dataset. Two parameters were adjusted to improve the performance of the method: Number of iteration (N) and membership function (MF).The objective of the experiments conducted using the ANFIS tool to find the best MF and N that used to build the Decision Support Model. The experiment that was performed to achieve this goal is:

**Tests:** several models using Triangular, Trapezoidal and Gaussian membership functions were tested in ANFIS method to build the Decision Support Model. Parameters (shown in table 4.6)

used to generate a Fuzzy Inference System (FIS) by implementing Grid partition on the data, the table shows that the best MF was the Triangular and the best N value was 10. It is clear that the RMSE was stabilized after this number of N.

Table 4. 6: The list of tests was performed on Z-Alizadeh Sani dataset to build the Decision Support Model

| Membership Function (MF) | # of iteration (N) | RMSE | MSE | Accuracy |
|---|---|---|---|---|
| Triangular (trimf) | 5 | 0.23482 | 0.5514 | 94.486% |
| | 10 | 0.22143 | 0.4903 | 95.097% |
| | 15 | 0.22143 | 0.4903 | 95.097% |
| | 20 | 0.22143 | 0.4903 | 95.097% |
| Trapezoidal (trapmf) | 5 | 0.24675 | 0.6088 | 93.911% |
| | 10 | 0.24322 | 0.5915 | 94.084% |
| | 15 | 0.2409 | 0.5803 | 94.197% |
| | 20 | 0.2409 | 0.5803 | 94.197% |
| Gaussian (gaussmf) | 5 | 0.2269 | 0.5148 | 94.852% |
| | 10 | 0.22538 | 0.5079 | 94.920% |
| | 15 | 0.2244 | 0.5035 | 94.964% |
| | 20 | 22386 | 0.5011 | 94.989% |

Table 4.6 shows the list of tests were performed to build the Decision Support Model. The minimum RMSE achieved with Triangular membership function and 10 neurons which is 0.22143, so triangular membership function was used to build the fuzzy sets and generate the rules for the Decision Support Model. The system achieved a prediction accuracy of 95.097%.

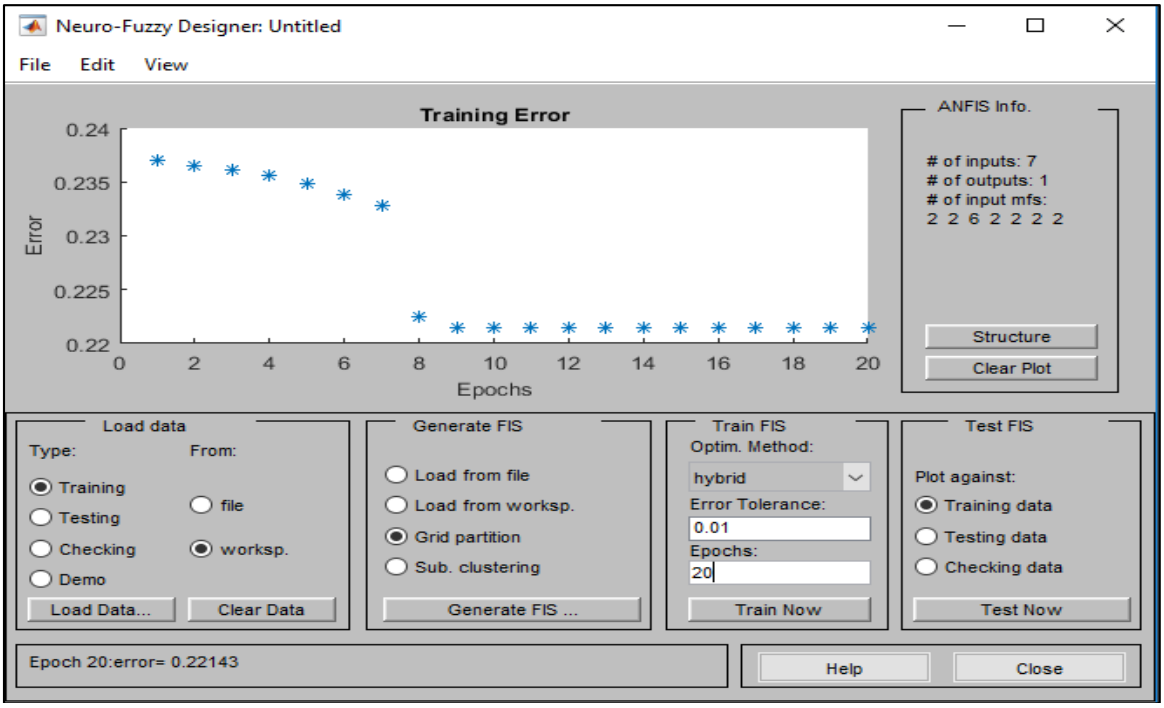


Figure 4. 9: The Neuro-Fuzzy designer for Heart Disease rule base system

Figure 4.9 shows the neuro-fuzzy designer for Heart Disease rule base system. The designer shows the training parameters: number of epochs, error tolerance, and ANFIS information. In addition to the relation between the training error and the number of epochs, where the error decreasing with the number of epochs selected. The system uses 384 rules, each input variable uses 2 membership functions except the age uses 6 membership functions. The total number of rules generated by the neuro-fuzzy system is calculated using the formula 4.4.1.

$$\text{The total number of rules} = \prod_{i=1}^{n} MFs\ of\ input(i) \qquad 4.4.1$$

Thus, the Entire number of rules for the proposed system = 2*2*6*2*2*2*2=384

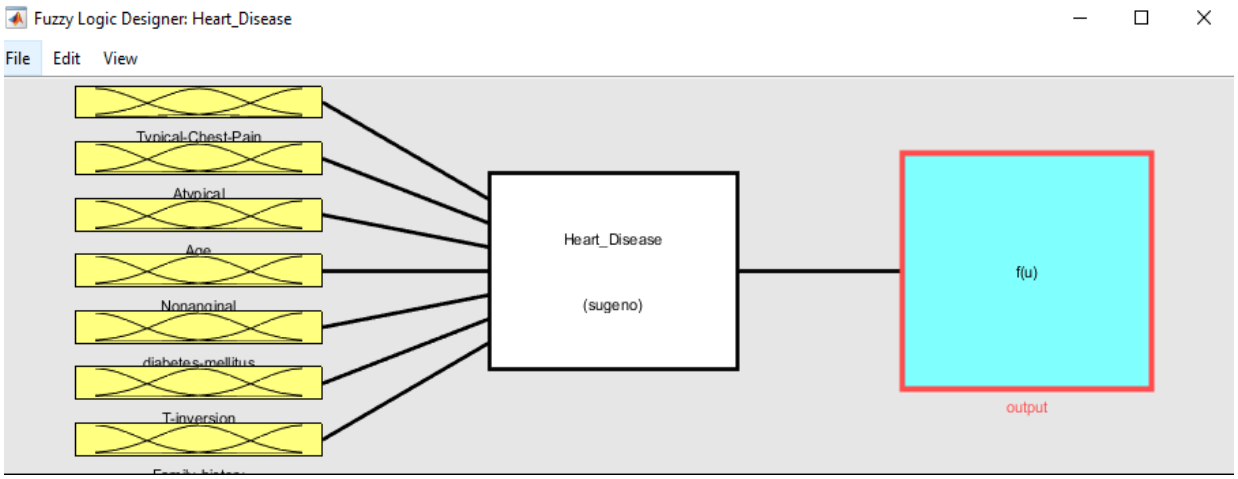


Figure 4. 10: The fuzzy logic structure for the Heart Disease rule base system.

Figure 4.10 shows the fuzzy logic structure for the Heart Disease rule base system. There are 7 input variables which are atypical chest pain, Atypical, Age, Nonanginal, DM, T-inversion, FH and one output which show the level of the risk on the patient. The name of the input variable and the membership function for each one is at the left of the figure. The type of the inference system is shown in the middle box which is Sugeno.

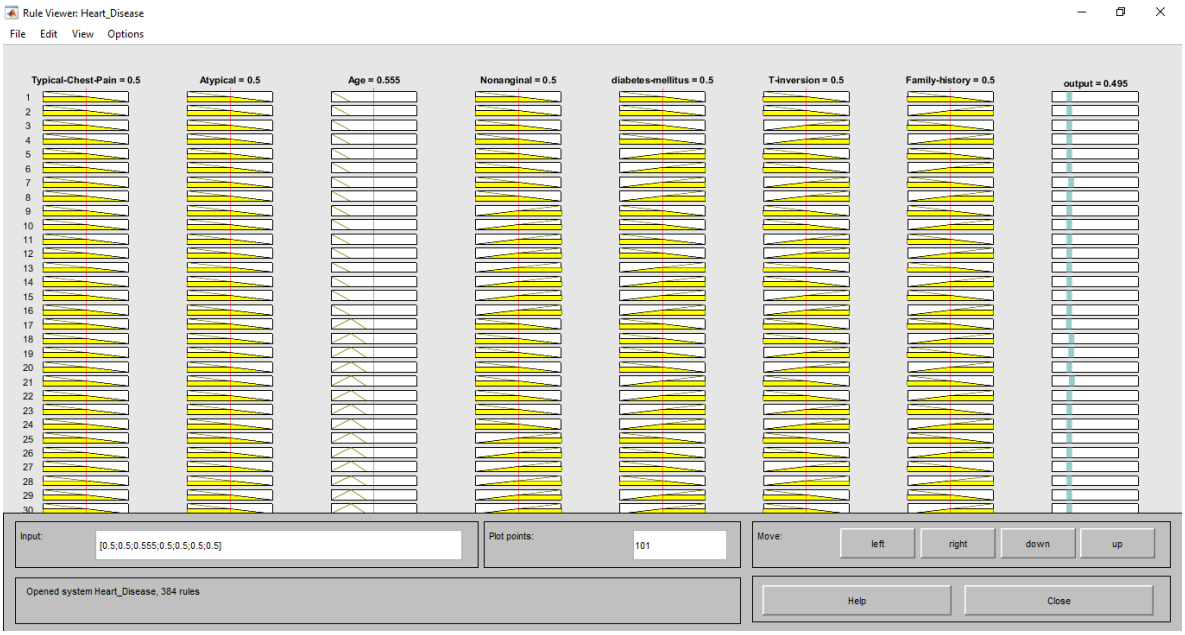


Figure 4. 11: There preventative rules in the neuro-fuzzy expert system.

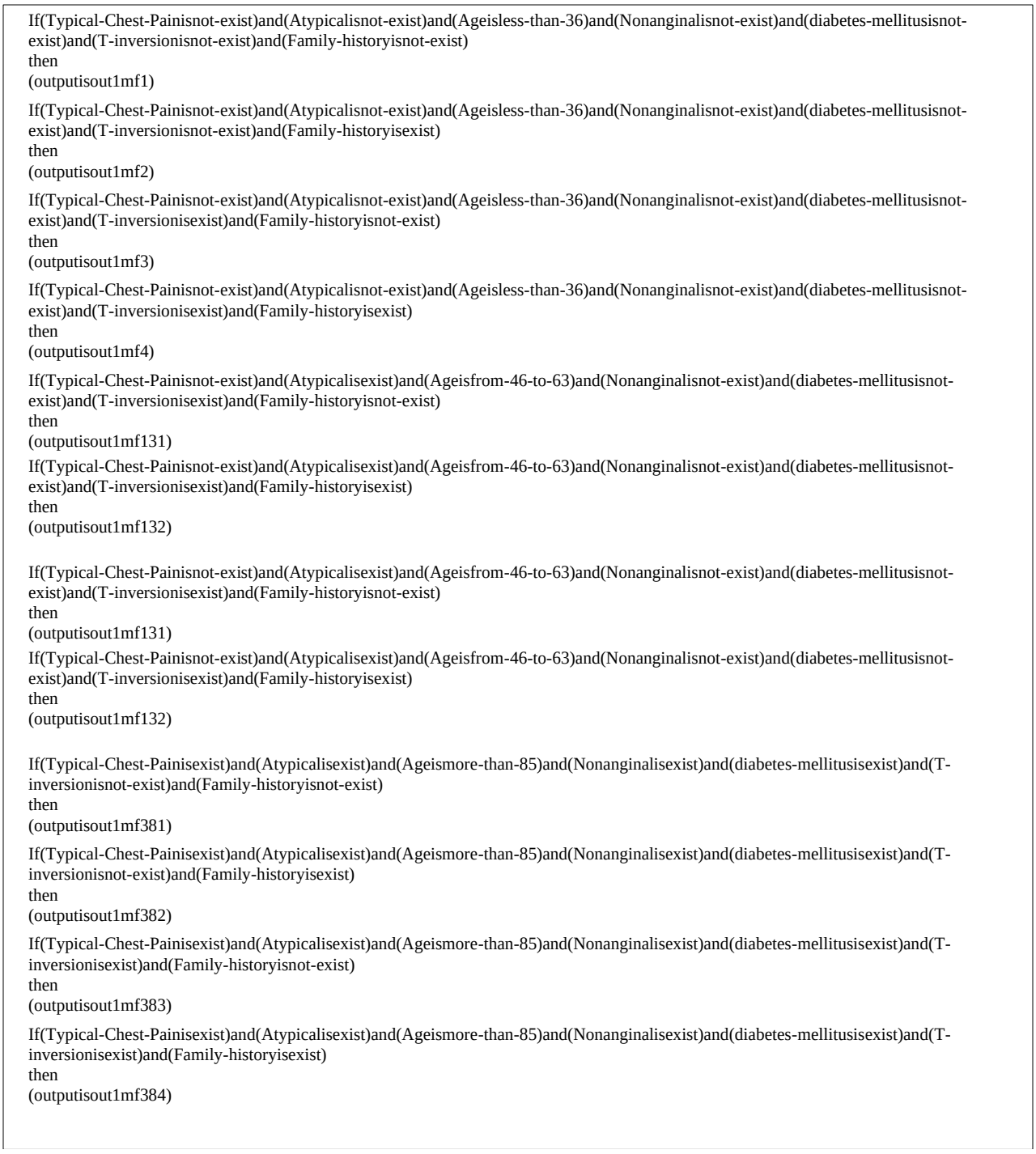


Figure 4. 12: screenshot of the system if-then rules that generated using the Neuro-Fuzzy.

Figure 4.11 shows the representative rules in the Neuro-Fuzzy expert system. For the decision support model, there are 384 Fuzzy if-then rules that give the relation between the input-output

parameters. Figure 4.12 shows a screenshot of some the rules that generated using the Neuro-Fuzzy System experiments used to build the intelligent system. The rule editor enables adding, deleting and updating the rules.

## 4.6 Limitation

The major challenge was faced is collecting local dataset to test our model, It is tried to collect dataset from the public sector which presented by the ministry of health, it is requested a dataset from the 'Palestine Medical Complex' and they refused to give data because of its sensitivity, despite a pledge to use it in scientific research only. After a while, useless and incomplete datasets that could not be used to test our model were given. After that, it is requested dataset from 'An-Najah National University Hospital' and four official letters to different parties in the hospital were sent upon their request, also the thesis proposal was requested to ensure that the data will be used for scientific research only. An official letter from Deanship of Student Affairs was requested for the same previous reason, and it is told verbally that dataset will not be given because of the competition between the two universities.

# Chapter 5

# Conclusion and Future work

## 5.1 Conclusion and Future Works

Expert systems in the medical field are very useful in real life and strongly support decision support systems since they allow decisions based on the human experience of a specialist in a certain area. Many times, the diagnosis is confused; the medical expert system will help in the rapid diagnosis, so that treatment can begin immediately, and avoid severe effects. Neural networks introduce the advantages of classification and diagnoses in the medical practice they contain valuable information. The medical data can be used to train neural networks and create expert systems; these enrich the diagnosis of the physicians and give him a new perspective.

The term “heart disease” is often used to refer to cardiovascular disease. Cardiovascular disease caused by Blocking or narrowing of the vessels that can lead toa heart attack or chest pain. Heart disease is the leading cause of death globally. However, saving lives can be achieved by the early and accurate diagnosis of the various types of heart diseases and provide the appropriate treatment. The aim of this work was to develop an Intelligent Medical Diagnosis and Decision Support Model that help in detecting the disease in early-stage.

In this thesis, a hybrid system that uses a Neuro-Fuzzy model is proposed to implement an Intelligent Medical Diagnosis for diagnosing heart disease. Also, the performance of the system was improved by optimizing the neural network parameter using optimization algorithms. A hybrid system that combines Genetic Algorithm (GAs) and Biogeography-Based Optimization (BBO) with neural networks (NNs) [GAsBBO-MLPNNs], BBO and particle swarm optimization (PSO) methods combined with the neural network to improve the performance of the system.

The proposed method produces better performance than previous works in terms of accuracy and Specificity, where the detecting of CAD and Normal class was improved. The GAsBBO-MLPNNs method produces the best result on the Z-Alizadeh Sani dataset with $L = 10$, $N =$

100, $P = 60$, and it achieved93.85%,95.6%, 89.94%, 96.4%, 94.8%in accuracy, G-mean, F-measure, Sensitivity, Specificity respectively. Intelligent Medical Diagnosis System using neuro fuzzy model with N= 10 using triangular membership function achieved a perdition accuracy of 95.097% on Z-Alizadeh Sani dataset,

In future work, datasets from other sources will be used to test the performance of the proposed system and expand the scope of the proposed from heart diseases to other diseases such as Lung Cancer and Alzheimer's disease. Different features will be applied such as extraction and reduction methods to improve the performance of the system. Finally, it is aimed to develop a friendly user interface for the system to facilitate use it by the specialist.

# Bibliography

[1] Mintz, Y., & Brodie, R. (2019). Introduction to artificial intelligence in medicine. Minimally Invasive Therapy & Allied Technologies, 28(2), 73-81.

[2]Patterson, D. W. (1990). Introduction to artificial intelligence and expert systems. Prentice-hall of India.

[3] Tran, B. X., Latkin, C. A., Vu, G. T., Nguyen, H. L. T., Nghiem, S., Tan, M. X., ... & Ho, R. (2019). The Current Research Landscape of the Application of Artificial Intelligence in Managing Cerebrovascular and Heart Diseases: A Bibliometric and Content Analysis. International journal of environmental research and public health, 16(15), 2699.

[4] http://www.medbroadcast.com/channel/heart/understanding-heart-disease/types-of-heart-disease. Accessed 22/7/2019

[5] Buja LM, Willerson JT. The role of coronary artery lesions in ischemic heart disease: insights from recent clinicopathologic, coronary arteriographic, and experimental studies. Hum Pathol., 18:451-61, 1987.

[6] Nasrabadi, A., &Haddadnia, J. (2016). Predicting Heart Attacks in Patients Using Artificial Intelligence Methods. Modern Applied Science, 10(3), p66.

[7] Feshki, M. G., &Shijani, O. S. (2016, April). Improving the heart disease diagnosis by evolutionary algorithm of PSO and Feed Forward Neural Network. In Artificial Intelligence and Robotics (IRANOPEN), 2016 (pp. 48-53). IEEE.

[8] https://www.cdc.gov/heartdisease/facts.htm Accessed 22/7/2019

[9] Amisha, P. M., Pathania, M., &Rathaur, V. K. (2019). Overview of artificial intelligence in medicine. Journal of family medicine and primary care, 8(7), 2328.

[10] Kidwai, M. S., & Khan, A. M. (2014). An Efficient Algorithm For The Diagnosis Of Alzheimer Disease By Using MATLAB. International Journal of Emerging Trends and Technology in Computer Science, 3(2), 297_298.

[11] Kahramanli, H. (2017). Training Product-Unit Neural Networks with Cuckoo Optimization Algorithm for Classification. International Journal of Intelligent Systems and Applications in Engineering, 5(4), 252-255.

[12] Cheng, R., He, C., Jin, Y., & Yao, X. (2018). Model-based evolutionary algorithms: a short survey. Complex & Intelligent Systems, 4(4), 283-292.

[13] Akkar, H. A., & Mahdi, F. R. (2016). Evolutionary algorithms for neural networks binary and real data classification. Int J Sci Technol Res, 5(7), 55-60

[14] Simon, D. (2008). Biogeography-based optimization. IEEE transactions on evolutionary computation, 12(6), 702-713.

[15] Gen, M., & Lin, L. (2007). Genetic algorithms. Wiley Encyclopedia of Computer Science and Engineering, 1-15.

[16] Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. Chemometrics and Intelligent Laboratory Systems, 149, 153-165.

[17] https://archive.ics.uci.edu/ml/datasets/Z-Alizadeh+Sani Accessed 2/2/2019

[18] Alizadehsani, R., Habibi, J., Hosseini, M. J., Mashayekhi, H., Boghrati, R., Ghandeharioun, A., ... & Sani, Z. A. (2013). A data mining approach for diagnosis of coronary artery disease. Computer methods and programs in biomedicine, 111(1), 52-61.

[19] Alizadehsani, R., Hosseini, M. J., Khosravi, A., Khozeimeh, F., Roshanzamir, M., Sarrafzadegan, N., &Nahavandi, S. (2018). Non-invasive detection of coronary artery disease in high-risk patients based on the stenosis prediction of separate coronary arteries. Computer methods and programs in biomedicine, 162, 119-127.

[20] Samuel, O. W., Asogbon, G. M., Sangaiah, A. K., Fang, P., & Li, G. (2017). An integrated decision support system based on ANN and Fuzzy_AHP for heart failure risk prediction. Expert Systems with Applications, 68, 163-172.

[21] Nazari, S., Fallah, M., Kazemipoor, H., &Salehipour, A. (2018). A fuzzy inference-fuzzy analytic hierarchy process-based clinical decision support system for diagnosis of heart diseases. Expert Systems with Applications, 95, 261-271.

[22] Chen, J., Li, K., Rong, H., Bilal, K., Yang, N., & Li, K. (2018). A disease diagnosis and treatment recommendation system based on big data mining and cloud computing. Information Sciences, 435, 124-149.

[23] Malmir, B., Amini, M., & Chang, S. I. (2017). A medical decision support system for disease diagnosis under uncertainty. Expert Systems with Applications, 88, 95-108.

[24] Paul, A. K., Shill, P. C., Rabin, M. R. I., &Murase, K. (2017). Adaptive weighted fuzzy rule-based system for the risk level assessment of heart disease. Applied Intelligence, 1-18.

[25] Sahoo, S., Kanungo, B., Behera, S., &Sabut, S. (2017). Multiresolution wavelet transform based feature extraction and ECG classification to detect cardiac abnormalities. Measurement, 108, 55-66.

[26] Shadmand, S., &Mashoufi, B. (2016). A new personalized ECG signal classification algorithm using block-based neural network and particle swarm optimization. Biomedical Signal Processing and Control, 25, 12-23.

[27] Herry, C. L., Frasch, M., Seely, A. J., & Wu, H. T. (2017). Heart beat classification from single-lead ECG using the synchrosqueezing transform. Physiological measurement, 38(2), 171.

[28] Arabasadi, Z., Alizadehsani, R., Roshanzamir, M., Moosaei, H., &Yarifard, A. A. (2017). Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm. Computer methods and programs in biomedicine, 141, 19-26.

[29] Acharya, U. R., Fujita, H., Lih, O. S., Adam, M., Tan, J. H., & Chua, C. K. (2017). Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network. Knowledge-Based Systems, 132, 62-71.

[30] Shinde, A., Kale, S., Samant, R., Naik, A., &Ghorpade, S. (2017). Heart Disease Prediction System using Multilayered Feed Forward Neural Network and Back Propagation Neural Network. International Journal of Computer Applications, 166(7), 32-36.

[31] Alizadehsani, R., Hosseini, M. J., Boghrati, R., Ghandeharioun, A., Khozeimeh, F., & Sani, Z. A. (2012). Exerting Cost-Sensitive and Feature Creation Algorithms for Coronary Artery Disease Diagnosis. International Journal of Knowledge Discovery in Bioinformatics (IJKDB), 3(1), 59-79.

[32] Cüvitoğlu, A., &Işik, Z. (2018, May). Classification of CAD dataset by using principal component analysis and machine learning approaches. In 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE) (pp. 340-343). IEEE.

[33] Alizadehsani, R., Roshanzamir, M., Abdar, M., Beykikhoshk, A., Zangooei, M. H., Khosravi, A., ... & Acharya, U. R. (2019). Model uncertainty quantification for diagnosis of each main coronary artery stenosis. Soft Computing, 1-12.

[34] Abushariah, M. A., Alqudah, A. A., Adwan, O. Y., & Yousef, R. M. (2014). Automatic heart disease diagnosis system based on artificial neural network (ANN) and adaptive neuro-fuzzy inference systems (ANFIS) approaches. Journal of software engineering and applications, 7(12), 1055.

[35] Verma, A., Agarwal, G., & Bhardwaj, S. (2018, November). SMS Advisory System for Medical Assistance by using Neuro Fuzzy System. In 2018 3rd International Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH) (pp. 1-4). IEEE.

[36] Chen, Y. W., & Lin, C. J. (2006). Combining SVMs with various feature selection strategies. In Feature extraction (pp. 315-324). Springer, Berlin, Heidelberg..

[37] Aggarwal, C. C. (2015). Data mining: the textbook. Springer.

[38] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), 115-133.

[39] Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), 2627-2636.

[40] Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1), 66-73.

[41] Cantú-Paz, E. (1998). *A survey of parallel genetic algorithms. Calculators parallels, reseaux et systems repartis*, 10(2), 141-171.

[42] Shi, Y. (2001, May). Particle swarm optimization: developments, applications, and resources. In Proceedings of the 2001 congress on evolutionary computation *(IEEE Cat. No. 01TH8546)* (Vol. 1, pp. 81-86). IEEE.

[43] Zhang, J. R., Zhang, J., Lok, T. M., &Lyu, M. R. (2007). *A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training*. Applied mathematics and computation, 185(2), 1026-1037.

[44] Ding, S., Su, C., & Yu, J. (2011). *An optimizing BP neural network algorithm based on genetic algorithm*. Artificial intelligence review, 36(2), 153-162.

[45] Zheng, Y., Lu, X., Zhang, M., & Chen, S. (2019). *Biogeography-based optimization: algorithms and applications.*

[46][Nwankpa, C., Ijomah, W., Gachagan, A., & *Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning*. arXiv preprint arXiv:1811.03378.]

[47] Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665-685.

[48] Vieira, J., Dias, F. M., &Mota, A. (2004, March). *Neuro-fuzzy systems: a survey. In 5th WSEAS NNA international conference on neural networks and applications*, Udine, Italia (pp. 87-92).

[49]Fullér, R. (1995). Neural fuzzy systems.

[50] S. L. Phung, A. Bouzerdoum and G. H. Nguyen, "*Learning Pattern Classification Tasks with Imbalanced Data Sets*," in Pattern Recognition, InTech, 2009

[51]Akosa, J. (2017, April). *Predictive accuracy: a misleading performance measure for highly imbalanced data*. In Proceedings of the SAS Global Forum (pp. 2-5).

[52]Refaeilzadeh, P., Tang, L., & Liu, H. (2009). *Cross-validation*. Encyclopedia of database systems, 532-538.

# Appendix

The appendix includes the result of optimization experiments that performed on Z-Alizadeh Sani dataset.

## Appendix A

This section contains the result of all experiments performed on the Z-Alizadeh Sani dataset to optimize some parameters of PSO-MLPNNs, BBO-MLPNNs, and GAsBBO-MLPNNs methods.

Table A. 1: result of GAsBBO-MLPNNs Experiments fold #1.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 93.5% | | | | | | Fold 1 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 91.6% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | | 93.6% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 150 | | 95.1% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 50 | 20 | 94.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 100 | | 94.1% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 150 | | 93.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 50 | 35 | 90.6% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | | 92.1% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 150 | | 93.1% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table A. 2: result of GAsBBO-MLPNNs Experiments fold #2.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 93.1% | | | | | | Fold 2 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 90.6% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 100 | | 93.6% | 93.1% | 7 | 1 | 20 | 1 | 87.5% | 95.2% | 87.5% | 95.2% |
| 150 | | 94.1% | 93.1% | 8 | 0 | 19 | 2 | 80.0% | 100.0% | 100.0% | 90.5% |
| 50 | 20 | 90.1% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 100 | | 95.1% | 93.1% | 8 | 0 | 19 | 2 | 80.0% | 100.0% | 100.0% | 90.5% |
| 150 | | 94.3% | 93.1% | 8 | 0 | 19 | 2 | 80.0% | 100.0% | 100.0% | 90.5% |
| 50 | 35 | 90.6% | 82.8% | 6 | 2 | 18 | 3 | 66.7% | 90.0% | 75.0% | 85.7% |
| 100 | | 89.7% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 150 | | 92.6% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |

Table A. 3: result of GAsBBO-MLPNNs Experiments fold #3.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 96.8% | | | | | | Fold 3 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 97.5% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 150 | | 98.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 50 | 20 | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 94.1% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 150 | | 94.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 50 | 35 | 93.1% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 100 | | 94.6% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 150 | | 92.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table A. 4: result of GAsBBO-MLPNNs Experiments fold #4.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 93.5% | | | | | | Fold 4 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 96.1% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 100 | | 95.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 150 | | 92.1% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 50 | 20 | 94.1% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 100 | | 94.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 150 | | 95.1% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 50 | 35 | 92.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 100 | | 91.1% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 150 | | 93.6% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |

Table A. 5: result of GAsBBO-MLPNNs Experiments fold #5.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 90.3% | | | | | | Fold 5 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 92.6% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 100 | | 92.1% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 150 | | 93.6% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 50 | 20 | 88.7% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 100 | | 89.2% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 150 | | 89.7% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 50 | 35 | 86.7% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 100 | | 90.1% | 87.1% | 5 | 4 | 22 | 0 | 100.0% | 84.6% | 55.6% | 100.0% |
| 150 | | 91.6% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |

Table A. 6: result of GAsBBO-MLPNNs Experiments fold #6.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 93.5% | | | | | | Fold 6 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 96.1% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 100 | | 95.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 150 | | 92.1% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 50 | 20 | 94.1% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 100 | | 94.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 150 | | 95.1% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 50 | 35 | 92.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 100 | | 91.1% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 150 | | 93.6% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |

Table A. 7: result of GAsBBO-MLPNNs Experiments fold #7.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 96.8% | | | | | | Fold 7 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 97.5% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 150 | | 94.1% | 93.3% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 50 | 20 | 97.5% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 98.0% | 96.8% | 9 | 0 | 21 | 1 | 90.0% | 100.0% | 100.0% | 95.5% |
| 150 | | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 50 | 35 | 91.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 100 | | 93.1% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 150 | | 93.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table A. 8: result of GAsBBO-MLPNNs Experiments fold #8.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 93.5% | | | | | | Fold 8 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 94.6% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 100 | | 95.1% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 150 | | 95.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 50 | 20 | 92.6% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 100 | | 94.1% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 150 | | 90.6% | 90.3% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 50 | 35 | 93.6% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 100 | | 94.1% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 150 | | 94.6% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |

Table A. 9: result of GAsBBO-MLPNNs Experiments fold #9.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 96.8% | | | | | | Fold 9 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 97.5% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 150 | | 98.0% | 96.8% | 9 | 0 | 21 | 1 | 90.0% | 100.0% | 100.0% | 95.5% |
| 50 | 20 | 97.5% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 150 | | 95.6% | 93.3% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 50 | 35 | 93.6% | 93.3% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 100 | | 97.5% | 96.8% | 9 | 0 | 21 | 1 | 90.0% | 100.0% | 100.0% | 95.5% |
| 150 | | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |

Table A. 10: result of GAsBBO-MLPNNs Experiments fold #10.

| GAsBBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 96.8% | | | | | | Fold 10 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 50 | 10 | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 100 | | 97.5% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 150 | | 94.6% | 93.3% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 50 | 20 | 97.5% | 96.8% | 9 | 0 | 21 | 1 | 90.0% | 100.0% | 100.0% | 95.5% |
| 100 | | 97.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 150 | | 98.0% | 96.8% | 8 | 1 | 22 | 0 | 100.0% | 95.7% | 88.9% | 100.0% |
| 50 | 35 | 93.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 100 | | 92.1% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 150 | | 93.1% | 90.3% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |

Table A. 11: result of BBO-MLPNNs Experiments fold #1.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 93.10% | | | | | | Fold 1 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 90.10% | 89.70% | 7 | 2 | 19 | 1 | 87.5% | 90.5% | 77.8% | 95.0% |
| 150 | | 94.20% | 93.10% | 7 | 1 | 20 | 1 | 87.5% | 95.2% | 87.5% | 95.2% |
| 200 | | 90.90% | 89.70% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 100 | 20 | 93.40% | 93.10% | 7 | 1 | 20 | 1 | 87.5% | 95.2% | 87.5% | 95.2% |
| 150 | | 94.20% | 93.10% | 7 | 1 | 20 | 1 | 87.5% | 95.2% | 87.5% | 95.2% |
| 200 | | 92.70% | 89.70% | 8 | 0 | 18 | 3 | 72.7% | 100.0% | 100.0% | 85.7% |
| 100 | 35 | 91.20% | 89.70% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 150 | | 92.00% | 89.70% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 200 | | 90.10% | 89.70% | 8 | 0 | 18 | 3 | 72.7% | 100.0% | 100.0% | 85.7% |

Table A. 12: result of BBO-MLPNNs Experiments fold #2.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 93.1% | | | | | Fold 2 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 94.5% | 93.1% | 7 | 1 | 20 | 1 | 87.5% | 95.2% | 87.5% | 95.2% |
| 150 | | 94.2% | 93.1% | 7 | 1 | 20 | 1 | 87.5% | 95.2% | 87.5% | 95.2% |
| 200 | | 97.1% | 96.6% | 8 | 0 | 20 | 1 | 88.9% | 100.0% | 100.0% | 95.2% |
| 100 | 20 | 90.5% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 150 | | 92.3% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 200 | | 91.2% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 100 | 35 | 93.1% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 150 | | 93.1% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 200 | | 93.4% | 93.1% | 7 | 1 | 20 | 1 | 87.5% | 95.2% | 87.5% | 95.2% |

Table A. 13: result of BBO-MLPNNs Experiments fold #3.

| Ten-fold BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 93.5% | | | | | Fold 3 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 93.4% | 83.9% | 7 | 3 | 19 | 2 | 77.8% | 86.4% | 70.0% | 90.5% |
| 150 | | 94.5% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 200 | | 93.1% | 90.3% | 7 | 1 | 21 | 1 | 87.5% | 95.5% | 87.5% | 95.5% |
| 100 | 20 | 91.2% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 150 | | 91.5% | 87.1% | 9 | 3 | 21 | 1 | 90.0% | 87.5% | 75.0% | 95.5% |
| 200 | | 92.7% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | 35 | 91.5% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 150 | | 91.9% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 200 | | 93.8% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |

Table A. 14: result of BBO-MLPNNs Experiments fold #4.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 93.5% | | | | | Fold 4 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 91.9% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 150 | | 94.2% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 200 | | 92.0% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |
| 100 | 20 | 94.5% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 150 | | 93.8% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 200 | | 94.2% | 93.5% | 8 | 1 | 22 | 1 | 88.9% | 95.7% | 88.9% | 95.7% |
| 100 | 35 | 91.2% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 150 | | 94.9% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 200 | | 93.8% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |

Table A. 15: result of BBO-MLPNNs Experiments fold #5.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 87.1% | | | | | Fold 5 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 90.1% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 150 | | 88.7% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 200 | | 92.0% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 100 | 20 | 92.3% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 150 | | 91.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 200 | | 93.1% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | 35 | 89.8% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 150 | | 90.1% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 200 | | 91.2% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table A. 16: result of BBO-MLPNNs Experiments fold #6.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 96.7% | | | | | Fold 6 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 93.8% | 93.3% | 8 | 0 | 20 | 2 | 80.0% | 100.0% | 100.0% | 90.9% |
| 150 | | 97.8% | 96.7% | 8 | 0 | 21 | 1 | 88.9% | 100.0% | 100.0% | 95.5% |
| 200 | | 97.4% | 96.7% | 7 | 1 | 22 | 0 | 100.0% | 95.7% | 87.5% | 100.0% |
| 100 | 20 | 93.4% | 90.0% | 8 | 0 | 19 | 3 | 72.7% | 100.0% | 100.0% | 86.4% |
| 150 | | 96.7% | 96.7% | 8 | 0 | 21 | 1 | 88.9% | 100.0% | 100.0% | 95.5% |
| 200 | | 98.2% | 96.7% | 8 | 0 | 21 | 1 | 88.9% | 100.0% | 100.0% | 95.5% |
| 100 | 35 | 94.9% | 93.3% | 8 | 0 | 20 | 2 | 80.0% | 100.0% | 100.0% | 90.9% |
| 150 | | 97.8% | 96.7% | 7 | 1 | 22 | 0 | 100.0% | 95.7% | 87.5% | 100.0% |
| 200 | | 98.5% | 96.7% | 8 | 0 | 21 | 1 | 88.9% | 100.0% | 100.0% | 95.5% |

Table A. 17: result of BBO-MLPNNs Experiments fold #7.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 96.7% | | | | | Fold 7 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 95.6% | 93.3% | 8 | 2 | 20 | 2 | 80.0% | 90.9% | 80.0% | 90.9% |
| 150 | | 97.5% | 96.7% | 8 | 0 | 21 | 1 | 88.9% | 100.0% | 100.0% | 95.5% |
| 200 | | 97.1% | 96.7% | 8 | 0 | 21 | 1 | 88.9% | 100.0% | 100.0% | 95.5% |
| 100 | 20 | 94.9% | 93.3% | 7 | 1 | 21 | 1 | 87.5% | 95.5% | 87.5% | 95.5% |
| 150 | | 98.2% | 96.7% | 7 | 1 | 22 | 0 | 100.0% | 95.7% | 87.5% | 100.0% |
| 200 | | 97.1% | 96.7% | 8 | 0 | 21 | 1 | 88.9% | 100.0% | 100.0% | 95.5% |
| 100 | 35 | 94.2% | 93.3% | 8 | 0 | 20 | 2 | 80.0% | 100.0% | 100.0% | 90.9% |
| 150 | | 95.6% | 93.3% | 8 | 0 | 20 | 2 | 80.0% | 100.0% | 100.0% | 90.9% |
| 200 | | 94.9% | 93.3% | 7 | 1 | 21 | 1 | 87.5% | 95.5% | 87.5% | 95.5% |

Table A. 18: result of BBO-MLPNNs Experiments fold #8.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 93.5% | | | | | Fold 8 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 93.8% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 150 | | 95.6% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 200 | | 94.5% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 100 | 20 | 92.7% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 150 | | 91.5% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 200 | | 90.8% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | 35 | 93.4% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 150 | | 95.6% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 200 | | 95.3% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |

Table A. 19: result of BBO-MLPNNs Experiments fold #9.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 96.7% | | | | | Fold 9 | | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 95.2% | 93.3% | 9 | 0 | 19 | 2 | 81.8% | 100.0% | 100.0% | 90.5% |
| 150 | | 97.4% | 96.7% | 8 | 1 | 21 | 0 | 100.0% | 95.5% | 88.9% | 100.0% |
| 200 | | 97.8% | 96.7% | 8 | 1 | 21 | 0 | 100.0% | 95.5% | 88.9% | 100.0% |
| 100 | 20 | 91.9% | 90.0% | 8 | 1 | 19 | 2 | 80.0% | 95.0% | 88.9% | 90.5% |
| 150 | | 95.6% | 93.3% | 8 | 1 | 20 | 1 | 88.9% | 95.2% | 88.9% | 95.2% |
| 200 | | 95.2% | 93.3% | 7 | 2 | 21 | 0 | 100.0% | 91.3% | 77.8% | 100.0% |
| 100 | 35 | 94.1% | 93.3% | 7 | 2 | 21 | 0 | 100.0% | 91.3% | 77.8% | 100.0% |
| 150 | | 96.7% | 96.7% | 9 | 0 | 21 | 1 | 90.0% | 100.0% | 100.0% | 95.5% |
| 200 | | 94.5% | 93.3% | 7 | 2 | 21 | 0 | 100.0% | 91.3% | 77.8% | 100.0% |

Table A. 20: result of BBO-MLPNNs Experiments fold #10.

| BBO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 86.2% | | | | | | Fold 10 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 90.5% | 82.8% | 7 | 1 | 17 | 4 | 63.6% | 94.4% | 87.5% | 81.0% |
| 150 | | 90.5% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 200 | | 90.5% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 100 | 20 | 91.6% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 150 | | 92.7% | 82.8% | 6 | 2 | 18 | 3 | 66.7% | 90.0% | 75.0% | 85.7% |
| 200 | | 92.3% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 100 | 35 | 93.8% | 82.8% | 6 | 2 | 18 | 3 | 66.7% | 90.0% | 75.0% | 85.7% |
| 150 | | 90.1% | 75.9% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 200 | | 93.8% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |

Table A. 21: result of PSO-MLPNNs Experiments fold #1.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 87.1% | | | | | | Fold 1 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 92.3% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 150 | | 93.8% | 83.9% | 7 | 2 | 19 | 3 | 70.0% | 90.5% | 77.8% | 86.4% |
| 200 | | 91.2% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 100 | 20 | 93.0% | 80.6% | 7 | 2 | 18 | 4 | 63.6% | 90.0% | 77.8% | 81.8% |
| 150 | | 91.2% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 200 | | 92.6% | 83.9% | 7 | 2 | 19 | 3 | 70.0% | 90.5% | 77.8% | 86.4% |
| 100 | 35 | 93.0% | 80.6% | 8 | 1 | 17 | 5 | 61.5% | 94.4% | 88.9% | 77.3% |
| 150 | | 89.0% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 200 | | 94.1% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |

Table A. 22: result of PSO-MLPNNs Experiments fold #2.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 89.7% | | | | | | Fold 2 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 89.4% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 150 | | 93.1% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 200 | | 93.8% | 89.7% | 7 | 1 | 19 | 2 | 77.8% | 95.0% | 87.5% | 90.5% |
| 100 | 20 | 92.0% | 82.8% | 7 | 1 | 17 | 4 | 63.6% | 94.4% | 87.5% | 81.0% |
| 150 | | 92.7% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 200 | | 91.6% | 89.7% | 6 | 2 | 20 | 1 | 85.7% | 90.9% | 75.0% | 95.2% |
| 100 | 35 | 91.2% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |
| 150 | | 92.3% | 86.2% | 6 | 2 | 19 | 2 | 75.0% | 90.5% | 75.0% | 90.5% |
| 200 | | 91.6% | 86.2% | 7 | 1 | 18 | 3 | 70.0% | 94.7% | 87.5% | 85.7% |

Table A. 23: result of PSO-MLPNNs Experiments fold #3.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 83.9% | | | | | | Fold 3 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 91.2% | 80.6% | 5 | 4 | 20 | 2 | 71.4% | 83.3% | 55.6% | 90.9% |
| 150 | | 90.4% | 83.9% | 7 | 2 | 19 | 3 | 70.0% | 90.5% | 77.8% | 86.4% |
| 200 | | 91.2% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 100 | 20 | 94.5% | 80.6% | 6 | 3 | 19 | 3 | 66.7% | 86.4% | 66.7% | 86.4% |
| 150 | | 91.5% | 83.9% | 6 | 3 | 20 | 2 | 75.0% | 87.0% | 66.7% | 90.9% |
| 200 | | 92.6% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 100 | 35 | 92.3% | 80.6% | 5 | 4 | 20 | 2 | 71.4% | 83.3% | 55.6% | 90.9% |
| 150 | | 91.9% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 200 | | 91.5% | 90.3% | 6 | 3 | 22 | 0 | 100.0% | 88.0% | 66.7% | 100.0% |

Table A. 24: result of PSO-MLPNNs Experiments fold #4.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 93.5% | | | | | | Fold 4 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 90.5% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 150 | | 93.4% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 200 | | 93.8% | 93.5% | 7 | 2 | 22 | 0 | 100.0% | 91.7% | 77.8% | 100.0% |
| 100 | 20 | 93.4% | 80.6% | 8 | 1 | 17 | 5 | 61.5% | 94.4% | 88.9% | 77.3% |
| 150 | | 93.8% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 200 | | 93.4% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | 35 | 92.3% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 150 | | 92.6% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 200 | | 90.8% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table A. 25: result of PSO-MLPNNs Experiments fold #5.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 90.3% | | | | | | Fold 5 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 90.4% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 150 | | 91.5% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 200 | | 92.3% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 100 | 20 | 93.4% | 87.1% | 7 | 2 | 20 | 2 | 77.8% | 90.9% | 77.8% | 90.9% |
| 150 | | 90.4% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 200 | | 91.5% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 100 | 35 | 91.5% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 150 | | 92.6% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 200 | | 91.5% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |

Table A. 26: result of PSO-MLPNNs Experiments fold #6.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 93.3% | | | | | | Fold 6 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 90.1% | 90.0% | 7 | 1 | 20 | 2 | 77.8% | 95.2% | 87.5% | 90.9% |
| 150 | | 91.2% | 90.0% | 7 | 1 | 20 | 2 | 77.8% | 95.2% | 87.5% | 90.9% |
| 200 | | 93.8% | 93.3% | 7 | 1 | 21 | 1 | 87.5% | 95.5% | 87.5% | 95.5% |
| 100 | 20 | 93.4% | 93.3% | 7 | 1 | 21 | 1 | 87.5% | 95.5% | 87.5% | 95.5% |
| 150 | | 94.1% | 93.3% | 7 | 1 | 21 | 1 | 87.5% | 95.5% | 87.5% | 95.5% |
| 200 | | 93.7% | 93.3% | 8 | 0 | 20 | 2 | 80.0% | 100.0% | 100.0% | 90.9% |
| 100 | 35 | 90.8% | 90.3% | 8 | 0 | 20 | 2 | 80.0% | 100.0% | 100.0% | 90.9% |
| 150 | | 90.0% | 90.0% | 8 | 0 | 19 | 1 | 88.9% | 100.0% | 100.0% | 95.0% |
| 200 | | 94.5% | 93.3% | 8 | 0 | 20 | 2 | 80.0% | 100.0% | 100.0% | 90.9% |

Table A. 27: result of PSO-MLPNNs Experiments fold #7.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 86.7% | | | | | | Fold 7 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 92.7% | 80.0% | 6 | 3 | 18 | 3 | 66.7% | 85.7% | 66.7% | 85.7% |
| 150 | | 90.5% | 83.3% | 7 | 2 | 18 | 3 | 70.0% | 90.0% | 77.8% | 85.7% |
| 200 | | 92.7% | 86.7% | 7 | 2 | 19 | 2 | 77.8% | 90.5% | 77.8% | 90.5% |
| 100 | 20 | 92.3% | 73.3% | 3 | 6 | 19 | 2 | 60.0% | 76.0% | 33.3% | 90.5% |
| 150 | | 93.0% | 73.3% | 3 | 6 | 19 | 2 | 60.0% | 76.0% | 33.3% | 90.5% |
| 200 | | 93.0% | 86.7% | 7 | 2 | 19 | 2 | 77.8% | 90.5% | 77.8% | 90.5% |
| 100 | 35 | 91.6% | 80.0% | 7 | 2 | 17 | 4 | 63.6% | 89.5% | 77.8% | 81.0% |
| 150 | | 93.8% | 83.3% | 7 | 2 | 18 | 3 | 70.0% | 90.0% | 77.8% | 85.7% |
| 200 | | 93.0% | 83.3% | 7 | 2 | 18 | 3 | 70.0% | 90.0% | 77.8% | 85.7% |

Table A. 28: result of PSO-MLPNNs Experiments fold #8.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 90.3% | | | | | | Fold 8 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 88.2% | 87.1% | 6 | 3 | 21 | 1 | 85.7% | 87.5% | 66.7% | 95.5% |
| 150 | | 89.3% | 87.1% | 8 | 1 | 19 | 3 | 72.7% | 95.0% | 88.9% | 86.4% |
| 200 | | 90.4% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | 20 | 90.4% | 87.1% | 8 | 1 | 19 | 3 | 72.7% | 95.0% | 88.9% | 86.4% |
| 150 | | 90.4% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 200 | | 92.3% | 90.3% | 8 | 1 | 20 | 2 | 80.0% | 95.2% | 88.9% | 90.9% |
| 100 | 35 | 92.3% | 90.3% | 7 | 2 | 21 | 1 | 87.5% | 91.3% | 77.8% | 95.5% |
| 150 | | 94.1% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |
| 200 | | 94.5% | 93.5% | 8 | 1 | 21 | 1 | 88.9% | 95.5% | 88.9% | 95.5% |

Table A. 29: result of PSO-MLPNNs Experiments fold #9.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 90.0% | | | | | | Fold 9 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precision | Negative Prediction |
| 100 | 10 | 91.2% | 90.0% | 7 | 2 | 20 | 1 | 87.5% | 90.9% | 77.8% | 95.2% |
| 150 | | 92.3% | 90.0% | 7 | 2 | 20 | 1 | 87.5% | 90.9% | 77.8% | 95.2% |
| 200 | | 93.0% | 90.0% | 8 | 1 | 19 | 2 | 80.0% | 95.0% | 88.9% | 90.5% |
| 100 | 20 | 89.4% | 86.7% | 6 | 3 | 20 | 1 | 85.7% | 87.0% | 66.7% | 95.2% |
| 150 | | 94.5% | 93.3% | 8 | 1 | 20 | 1 | 88.9% | 95.2% | 88.9% | 95.2% |
| 200 | | 93.4% | 93.3% | 7 | 2 | 20 | 1 | 87.5% | 90.9% | 77.8% | 95.2% |
| 100 | 35 | 93.8% | 93.3% | 7 | 2 | 21 | 0 | 100.0% | 91.3% | 77.8% | 100.0% |
| 150 | | 94.1% | 93.3% | 7 | 2 | 21 | 0 | 100.0% | 91.3% | 77.8% | 100.0% |
| 200 | | 96.7% | 96.7% | 9 | 0 | 20 | 1 | 90.0% | 100.0% | 100.0% | 95.2% |

Table A. 30: result of PSO-MLPNNs Experiments fold #10.

| PSO-MLPNNs | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | 82.8% | | | | | | Fold 10 | | | |
| N | L | Training Accuracy | Testing Accuracy | TP | FP | TN | FN | Sensitivity | Specificity | Precession | Negative Prediction |
| 100 | 10 | 92.0% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 150 | | 92.0% | 79.3% | 5 | 3 | 18 | 3 | 62.5% | 85.7% | 62.5% | 85.7% |
| 200 | | 94.2% | 82.8% | 6 | 2 | 18 | 3 | 66.7% | 90.0% | 75.0% | 85.7% |
| 100 | 20 | 92.7% | 75.9% | 5 | 3 | 17 | 4 | 55.6% | 85.0% | 62.5% | 81.0% |
| 150 | | 94.2% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 200 | | 93.4% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 100 | 35 | 89.8% | 75.9% | 6 | 2 | 16 | 5 | 54.5% | 88.9% | 75.0% | 76.2% |
| 150 | | 92.0% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |
| 200 | | 92.7% | 79.3% | 6 | 2 | 17 | 4 | 60.0% | 89.5% | 75.0% | 81.0% |

النموذج الهجين يجمع بين الخوارزمية الجينية (GAs) مع التحسين القائم على الجغرافيا الحيوية (BBO) والشبكات العصبية] (NNs) [GAsBBO- MLPNNs]،التحسين القائم على الجغرافيا الحيوية (BBO)وأساليب تحسين سرب الجسيمات (PSO) مع الشبكات العصبية (NNs) لتحسين أداء النظام. فكرة هذه الأطروحة ترتكز على استخدام تنقيات الذكاء الإصطناعي لتقديم نظام تشخيص طبي لأمراض القلب. حقق النهج المقترح أداء أفضل من الطرق السابقة؛ حيث كانت نتائج معايير أداء النهج الهجين [GAsBBO-MLPNNs] ممثلة بالنسب 94.5%، 95.6%، 89.94% والتي تعبر عن الدقة ، المتوسط ، والمقياس على التوالي.حقق نظام نموذج Neuro-Fuzzy بالاعتماد على وظيفة عضوية من نوع triangular تشخيص طبي بدقة تصل إلى 95.097٪.

## الملخص

يعد جسم الإنسان مصدراً حيوياً للبيانات، الأمر الذي يجعلها مهمة لصحته. التصوير الطبي أحد العمليات التي تنتج أنواعاً مختلفة من البيانات لجسم الإنسان. في حين أن بعض هذه البيانات على شكل صور، البعض الآخر على شكل إشارات. تستخدم البيانات يوميا لتشخيص أنواع مختلفة من الأمراض. هناك العديد من الإشارات الحيوية المختلفة التي يمكن جمعهامن جسم الإنسان واستخدامها في تشخيص الأمراض المختلفة ؛ ومثالا على بعض هذه الإشارات: المخطط الكهربائي للقلب (ECG) ، المخطط الكهربائيللدماغ (EEG) و المخطط الكهربائي للعضلات (EMG). إن تصميم وتنفيذ البرامج الذكية التي تحاكي الذكاء البشري إشارة الى التكامل بين العلوم المختلفة ومجالات المعرفة. أحد المجالات المهمة هو التحسين الذي يسمح بتقديم المساعدة المناسبة للأطباء في صنع القرار. ان تطوير التنقنيات المرتبطة بالذكاء الاصطناعي المستخدمة في الطب يقلل من التكاليف والوقت والأخطاء الطبية. ان التكامل بين الذكاء الاصطناعي والطب هو أمر حيوي ، وبالرغم من بذل جهود كبيرة في هذا المجال إلا أنه بحاجة إلى بذل المزيد . إن استخدام هذه البيانات من قبل أنظمة دعم القرارت الطبية والتشخيصية الذكية يمكّن من تحسين الرعاية الصحية. يمكن اعتبار التشخيص الطبي كمعيار للعديد من أنظمة الذكاء الاصطناعي مثل: الأنظمة الخبيرة (التشخيص المعتمد على القواعد والاحتمالات)، المنطق الضبابي (التشخيص المعتمد على التصنيف)، الشبكات العصبية (NNs) (التشخيص المعتمد على التدريب والتعرّف)، تنقيب البيانات (التشخيص من خلال التعرف على الأنماط).

تم في هذه الأطروحة تقديم تقنية تستخدم المتغيرات الفسيولوجية لتشخيص أمراض القلب، حيث استخدمنا نظامًا هجينًا يدمج الأنظمة الخبيرة مع الشبكات العصبية (NNs)لإنتاج نظام تشخيص طبي ذكي يساعد في اتخاذ القرار في التطبيقات الطبية. كما قمنا بتحسين النظام عن طريق تحسين معاملات الشبكة العصبية (NNs) بإستخدام خوارزميات التحسين.