T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Yapay Alg Algoritması ile Yapay Sinir Ağlarının Ağırlıklarının Optimizasyonu

SAEED SHAKIR MAHMOOD

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Aralık-2018
KONYA

TEZ KABUL VE ONAYI

SAEED SHAKIR MAHMOOD tarafından hazırlanan "**The Weights Optimization of Artificial Neural Network with Artificial Algae Algorithm**" adlı tez çalışması 28/12/2016 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği** Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak Kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Doç.Dr. Barış KOÇER	.....
Danışman Doç. Dr. Gülay TEZEL	.....
Üye Doç.Dr. Humar KAHRAMANLI	.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Mustafa Yılmaz
FBE Müdürü

Bu tez çalışması ..... tarafından ..... nolu proje ile desteklenmiştir.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

SAEED SHAKIR MAHMOOD

Tarih: 28.12.2018

ÖZET

YÜKSEK LİSANS TEZİ

Yapay Alg Algoritması ile Yapay Sinir Ağlarının Ağırlıklarının Optimizasyonu

SAEED SHAKIR MAHMOOD

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç.Dr.GULAY TEZEL

2018, 53 Sayfa
Jüri
Doç.Dr. Gülay TEZEL
Doç.Dr. Barış KOÇER
Doç.Dr. Humar KAHRAMANLI

Son yıllarda, Yapay sinir Ağı (YSA), yapay zeka uygulamaları içinde en çok tercih edilen teknik ve bunun sebebi dinamik olarak sınıflandırma veya tahmin problemlerini çözebilme yeteneğidir. YSA üzerine yapılan pek çok çalışma, YSA'nın daha iyi öğrenmesini ve sınıflandırma veya tahmin için verilen dataya göre daha doğru davranmasını amaçlamaktadır. YSA güçlü bir teknik olmasına rağmen, ağın eğitimi zor bir görev olabilir ve daha karmaşık problemler, ağın eğitiminin daha zor hale gelmesine neden olur. Bu kısıtların üstesinden gelmek için yeni yaklaşımlar geliştirilmektedir. Bu yaklaşımlar, YSA modelleme sürecinin başarısını etkileyen en önemli faktörler olduğu için çoğunlukla eğitme algoritması ve parametre optimizisyonuna dayanmaktadır. Bu amaçla kullanılan birçok metaheuristik optimizasyon algoritması vardır. Parçacık Sürü Optimizasyonu (PSO), bunların en yaygın algoritmasıdır. Yapay Alg Algoritması (YAA) yeni geliştirilen ve farklı tipte problemlerde yüksek başarı gösteren bir optimizasyon algoritmasıdır. Algoritma, Evrimsel Süreç, Adaptasyon Süreci ve mikro algın hareketine dayanır. Bu çalışmada YAA, geri yayılım yerine YSA eğitim süreci olarak önerilmiştir. YSA-PSO, algoritmaların performanslarını karşılaştırmak için önerilen YSA-YAA ile aynı koşullarda geliştirilmiştir. UCI KDD Makine Öğrenme Deposu'ndan elde edilen iki veri seti bu amaçla kullanılmıştır. Ayrıca hem sınıf hedef değerlerinin ve hem de farklı nitelik kombinasyonlarının etkisini gözlemek amacıyla da YSA-PSO ve YSA-YAA ile denemeler yapılmıştır. Sınıf hedef değerleri (0, 1), (0.1, 0.9), (0.2, 0.8) ve (0.2, 0.6) olarak seçilmiştir. Sonuçlar göstermiştir ki, YSA-PSO, YSA-YAA'dan biraz daha iyi performansa sahiptir. Ancak YSA-YAA'nın da YSA-PSO'ya benzer bir doğruluk gösteren çok etkili bir rakip olduğu görülmüştür. Her iki veri kümesi için de (0.2,0.6) hedef değerleri için başarılı sonuçlar elde edilmiştir.

Anahtar Kelimeler: Yapay Alg Algoritması, Yapay Sinir Ağları, Parçacık Sürü Optimizasyonu, Sınıflandırma.

ABSTRACT

MSc THESIS

The Weights Optimization of Artificial Neural Network with Artificial Algae Algorithm

SAEED SHAKIR MAHMOOD

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY THE DEGREE OF MASTER OF SCIENCE OF PHILOSOPHY IN COMPUTER ENGINEERING

Advisor: Assoc.Prof.Dr. GULAY TEZEL
2018, 53 Pages
Jury
Assoc.Prof.Dr. Gülay TEZEL
Assoc.Prof.Dr. Barış KOÇER
Assoc.Prof.Dr. Humar KAHRAMANLI

Recently, Artificial Neural Network (ANN) was the most preferred technique for artificial intelligence applications, and the reason is the dynamic ability to solve classification or prediction problems. Most studies on ANN aim to develop a model that learn perfectly and to behave correctly according to the given data. Although ANN is a powerful technique, training a network can be a difficult task, and the more complex problems are, the more difficult the training of the network becomes. The most important studies on training algorithm and parameter optimization as these are the most important factors which affect the success of the process of ANN modeling. There are several metaheuristic optimization algorithms used to train ANN. Particle Swarm Optimization (PSO) is one of the most common algorithm of these. Artificial Algae Algorithm (AAA) is an optimization algorithm which has been newly developed and showed high success in different types of problems. The algorithm is based on the Evolutionary Process, Adaptation Process and the movement of microalgae. In this study, AAA was suggested as an ANN training algorithm instead of backpropagation. ANN-PSO is improved in the same conditions with the proposed ANN-AAA to compare the performances of algorithms. Two benchmark datasets obtained from UCI KDD Machine Learning Repository were used for this aim. Also the experiments with MLP-PSO and MLP-AAA were performed to investigate both the effects of class target values and different attribute combinations. The class target values were selected as (0, 1), (0.1, 0.9), (0.2, 0.8) and (0.2, 0.6). The results showed that ANN-PSO performed slightly better than ANN-AAA. But performance of ANN-AAA is very effective and competitor to ANN-PSO which showed good accuracy. The successful accuracy values were obtained for both datasets with the (0.2, 0.6) target values.

Keywords: Artificial Algae Algorithm, Artificial Neural Networks, Particle Swarm Optimization, classification.

ÖNSÖZ

Huge thanks to Assoc. Prof. Dr. Gülay TEZEL for her support to finish this research, another huge thanks I give to my parents for superior help and support and I appreciate all their prayers to god for me, I give to my wife Dr. Salwa Al-Nu'aymi and my daughter Raseel S. Mahmood my love and thanks for their patience, support and for being with me all the time.

SAEED SHAKIR MAHMOOD
KONYA-2018

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Simgeler

c1, c2:	Learning factors
pop:	Particle size
gBest:	global best position of swarm
İ:	Particle index
J:	particle size index
k:	Iteration sequence
n:	Number of iterations
N:	Number of particles
Pbest:	Local best position of particle
r1, r2:	Normal distributed random number
t:	Decreasing coefficient
xij:	Particle position
vij:	Particle velocity
Vmax:	Maximum particle velocity
w:	Weight of inertia
Ap:	Adaptation parameter in AAA
D:	Problem dimension
e:	Energy loss parameter in AAA
f:	Purpose function
G:	Size of algae colonies in AAA
g:	Inequality constraint function
h:	Equality constraint function
K:	Half saturation constant of artificial algal column
N:	Number of algal colonies in the population
S:	Nutrient concentration of artificial algal colon
X:	Design vector
a:	Significance level
Δ:	Shear force coefficient at AAA
τ:	Friction surface areas of algae in AAA
μ:	Specific growth rate
μ_max:	Maximum specific growth rate

Kısaltmalar

ANN:	Artificial neural network
MLP:	Multi-layer perceptron
AAA:	Artificial algae algorithm
PSO:	Particle swarm optimization

1. Introduction

In recent years, the amount of data collected from the medical industry has been on the increase. This data was very beneficial on the diagnosis of a specific disease for specific patients in medical treatment. However, the human analysis still will not be able to understand this data due to its size and its complexity. For that reason, computational tools are required to analyse and understand this data. These are the main reasons to choose machine learning as the best way to understand the data and classify it according to its characteristics and features.

Classification is an important technique that depends on the features of the data that is going to be classified; this technique is also considered as data mining. One of the most famous classification techniques is the Artificial Neural Network (ANN). ANN acts like a human brain with its great ability to understand the data according to its features. ANN is just a series of artificial neurons that are connected to each other in a network form. Backpropagation is considered the most common algorithm to train an ANN that focuses on changing the connection weight between neurons to provide the best knowledge. Backpropagation is used to compute the gradient error in the training step ANN.

Recently, there has been a considerable research effort to apply evolutionary calculation techniques (EC) to the development of one or more aspects of ANN such as selection of training algorithms, network topology and, the transfer function (Agam Gupta et al. 2015). The most effective aspect of these efforts is to increase the performance of ANN is the optimizing of weights using metaheuristic optimization algorithms.

Especially, Biological Inspired Algorithms (BIAs) have a good approval by the Artificial Intelligence (A.I) society because they are robust optimization tools and can solve highly complex optimization problems. BIAs have the ability to search big multimodal and non-continuous explore spaces and have the ability to discover the best solution, can make the results close to the optimum value. (Yao , 1999).

For example, Conforth and Meng. suggested a method that combines Ant Colony Optimization (ACO) to find a particular architecture (the connections) for ANN and Particle Swarm Optimization (PSO) to improve the synaptic weights (Conforth and Meng, 2008).

In a different study, researchers performed a modified PSO mixed with Simulated Annealing (SA) to obtain a combination of interlocking weights and thresholds (Da and Xiurun, 2005).

The researchers used a new evolutionary system for evolving ANNs based on Evolutionary Programming (EP) called EPNet. This algorithm had the ability to generate a very compact ANN that was able to make a good results on several on medical diagnosis, and classification and prediction problems (Yao and Liu, 1997). Abdul-Kader et al. applied the algorithm of differential evolution (DE) on ANN to solve the problem of weather prediction (Abdul-Kader, 2009).PSO algorithm was used to adjust the synaptic weight of a feedforward ANN to predict the daily relationship of rainfall-runoff status in Malaysia (Kuok et al., 2010).

To solve classification problems and to adjust the connection weights, Garro et al. compared the back-propagation method to basic PSO (Garro et al., 2011a). However, Garro et al. presented that differential evaluation (DE) might be at some point better than PSO in collecting the most improved set of weights for a feedforward ANN. In addition they explained that PSO could be better to get the best set of weights for a feedforward ANN if the correct parameters were used (Garro et al., 2011b).

In similar works, the three major elements of an ANN have improved at the same time: architecture, transfer functions, and synaptic weights (Garro et al., 2009).The authors suggested for the same major elements using a Differential Evolution (DE) algorithm (Garro et al., 2010).

Artificial Algae Algorithm (AAA) is a new and successful bio-inspired metaheuristic optimization algorithm that based on algae cells behaviour for life according to the availability of light on the surface water (Uymaz et al., 2015). Because the achievement of this algorithm in the high dimensional problems, there is increasing interest to this algorithm.

AAA was preferred to optimize the weight of feedforward ANN. This algorithm is used to train of feedforward ANN by getting as best set of weights as possibly can, and its performance is going to be compared with PSO performance on the train a feedforward ANN. The first chapter focuses on the introduction of the thesis which explains the idea and the aim of this thesis; the second chapter focuses on the material and the method of this thesis, which explains how the data was collected, and the

general look of ANN, PSO, and AAA. The third chapter gives a full idea about how PSO and AAA trains an ANN, and the process of those algorithms in the training phase. The fourth chapter gives an open look at the results of this work which shows the performance of both PSO and AAA in training the ANN and compares the results between them. The last chapter, results, and recommendation gives an overview of the results and summarizes them.

1.1 Purpose and Importance of Thesis

The objective medical diagnosis for all patients using biomedical signals or analysis parameters can be difficult for the experts. Computational tools such as artificial intelligence methods are mostly the most important helper for the doctors to interpret complex medical data. ANN is the most preferred AI method in similar studies.

Because of the importance of understanding the risk of diabetes and hepatitis, it is clearly shown that the correct diagnosis leads to a full understanding of the situation of these diseases. Using techniques like ANN might help doctors to fix huge problems like locating the disease and keeping everything under control. The purpose of this study was to present a hybrid algorithm based on a novel optimization algorithm called AAA to find optimum weights of ANN using benchmark classification problems. Therefore, ANN-PSO hybrid algorithm, which was commonly preferred, was implemented to compare with ANN-AAA. Moreover, two benchmark medical datasets were selected to evaluate their performances. Furthermore, this study aimed to explore the behaviour of two bio-inspired algorithms (AAA and PSO) using different values of their parameters for different output values of problems.

During the experimentation phase, the best parameter's values for these algorithms were determined to obtain the best results. In addition, the best arrangement was used to create a set of a statistically good experience for each selected classification problem. Moreover, the results gained with the suggested methodology regarding the number of hidden layer's neurons.

1.2. Literature review

Artificial Neural Networks (ANNs) can offer an inclusive framing for performing non-linear mappings from input variables to output variables, and they can be counted as an expansion of the many conventional mapping techniques (Alba and Marti, 2006).

A hybrid neural network that contains both Artificial Neural Network and Fuzzy Neural network (FNN) is used to classify two datasets from the University of California at Irvine (UCI) machine learning repository. The classification accuracies were 84.24% for Pima Indians Diabetes and 86.8% Cleveland heart disease (Kahramanli and Allahverdi, 2008).

Applying error back-propagation to a medical classification problem by comparing the performance of two neural networks, the results were gathered from conventional linear discriminant analysis or the technique of classification tree and regression tree. Neural networks showed a unique ability to detect features hidden in the inputs of the data (Reibnegger et al., 1991).

Differential Evolution (DE) algorithm was used to design an ANN automatically due to its efficiency and adaptability to solve nonlinear optimization problems. The main goal was to look for the best topology, transfer function, and synaptic weights to make a solution to multiple classification problems (Garro et al., 2010).

A different combination of ANNs and evolutionary algorithms (EAs) was used, including using EAs for ANNs' evolving in weights connection, architecture, learning rule, and an input feature. Also, different search operators were used with various EAs, and possible future research direction was pointed out (Yao , 1999).

Two algorithms Particle Swarm Optimization (PSO) and Differential Evolution (DE) used to train ANN where the applicability of PSO and DE on finding the best weights for ANN were explained and how PSO and DE are usable for training ANN to solve different nonlinear problems were studied (Garro et al., 2011b).

Swarm Intelligence explained as a form of intelligent behaviour exhibited by systems can be non-trivial. It was also explained how Cellular Robotic Systems

consisted of collections of autonomous, non-intelligent, non-synchronized robots which considered as an engineering problem and also explained the clarity that the term "SWARM" could associate with robotic concepts (Beni et al., 1993).

An intuition-based optimization model was developed after the behaviour of birds' swarms which was called (PSO), which seems to be effective in a good wide range of functions due to its simplicity. Its adjustment was explained as it goes towards the global best and public best (best particle and fitness) according to particles behaviour which made this method conceptually similar to the crossover in the genetic algorithm (Eberhart and Kennedy, 1995).

A new parameter included to (PSO) was called inertia weight, where the simulation had been performed to clarify the impact of this parameter on PSO. This showed that PSO had better performance when the inertia weight is in the range of (0.9, 1.2). Inertia weight could bring an amazing improvement in PSO performance (Shi and Eberhart, 1998).

A hybrid PSOs, based on PSO, and a Genetic Algorithm (GA) were developed which could combine velocity and position update rules using the ideas of breeding and subpopulations. This work was better than the standard PSO and the speed of convergence was much faster (Løvbjerg et al., 2001).

A new optimization algorithm was proposed which was called as Second Generation Particle Swarm Optimization or (SGPSO), which was better than the standard Particle Swarm Optimization (PSO) and PSO with Time-Varying Acceleration Coefficient (PSO-TVAC) in both accuracy and passing local optimum solution. This result was obtained from the tests done on several functions (Chen, 2008).

Random Dynamic Neighbourhoods in PSO was applied to train an ANN to have a good understanding on its performance on real-world dataset classification from seismological data which compared with the standard PSO on the same subject. Random Dynamic PSO showed a good performance compared to PSO. The paper also noted that the work needed more improvement due to the much of time taken for the algorithm to train an ANN (Mohais et al., 2007).

PSO algorithm was applied to train Artificial Neural Network (ANN) for the diagnosis of epilepsy, and the results were compared to ANN trained with backpropagation. The results of this work showed that PSO could be used to make

good training and make a perfect neural network to classify epilepsy datasets (Yalcin et al., 2015).

PSO with the Gaussian mutation was presented, which was compared with the standard PSO and the standard Genetic Algorithm (GA). The results showed that PSO with Gaussian mutation showed a magnificent performance compared to the standard PSO and the standard GA and was able to make better results than those of standard PSO and standard GA (Higashi and Iba, 2003).

Inspired by the living behaviour of microalgae and photosynthetic species, a new algorithm called Artificial Algae Algorithm (AAA) was presented. This algorithm was based on the process of evolution, adaptation, and the movement of microalgae. AAA was compared to other algorithms like Artificial Bee Colony (ABC), Bee Algorithm (BA), Deferential Evaluation (DE), Ant Colony Optimization for the continuous domain (ACO_R), and Harmony Search (HS_{POP}). AAA had made a good performance compared to some algorithms which AAA could perform better than other optimizations if AAA had the correct values for its parameters (Uymaz et al., 2015).

A methodology based on the Artificial Bee Colony (ABC) was used to train the Artificial Neural Network (ANN). This methodology allowed not only the optimization of the synaptic weights, the ANN architecture, and transfer function for each neuron but also optimizes ANN design without reducing to the design. The aim of this work was to maximize the accuracy of an ANN and minimize the number of connections of an ANN (Garro et al., 2011c).

A method that combines two algorithms, PSO and ACO, used in the form of Swarm Intelligence Based Reinforcement Learning (SWIRL) to train the ANN was presented. The idea was PSO adjusts the synaptic weights of an ANN and ACO optimizes ANN's topology. SWIRL showed a good performance which considered competitive with neuroevolutionary techniques (Conforth and Meng, 2008).

A new evolutionary (ANN) trained by Improved Particle Swarm Optimization (IPSO) was presented. IPSO had improved the performance of PSO. It was understood that IPSO could be used to address the design problem of feedforward ANN. The performance of the method in this work was considered as competitive to other the performance of other algorithms and there were some ideas about a way to improve this work (Yu et al., 2007).

A PSO algorithm based classification technique for classification rule mining was presented. This work was compared with Ant-Miner and Organizational CoEvolutionary algorithm for Classification (OCEC) in public domain datasets, where the proposed work showed better accuracy and compared to Ant-Miner and OCEC (Wang et al., 2007).

Two algorithms were used to train the ANN, PSO and BP. In this work, each of the two algorithms and how they train an ANN were explained, and also how possibly that Bio-inspired algorithms could be better than BP. The results showed that it's possible that PSO could be better than BP when the right parameters chosen (Garro et al., 2011a).

A hybrid methodology was proposed based on the capability of Self-Regulated PSO (SRPSO) to improve a Feed Forward Neural Network called Extreme Learning Machine (ELM). In this work, five medical benchmark datasets were taken from the University of California Irvine (UCI) Machine Learning Repository and used on SRPSO trained ELM and on several other classification techniques for comparison. The results showed that the proposed method had better performance than the other classification methods in general (Subbulakshmi and Deepa, 2015).

PSO algorithm was used to optimize the weight and architecture of an ANN where several medical datasets were used for this purpose. In this work, the ability of PSO was good enough for a good optimization to the weights and architecture of an ANN (Carvalho and Ludermir, 2007).

A new technique to train a feedforward ANN was presented. This technique was based on a Particle with Ability of Local search Swarm Optimization (PALSO) algorithm. This technique was compared with quasi-Newton and Conventional Neural Network, which the proposed technique showed a great performance with high accuracy and great training results (Ninomiya and Zhang, 2008).

Four algorithms used to train Multilayer Perceptron Neural Network (MLPNN) on diabetes dataset. The algorithms were BP, Delta-Bar-Delta (DBD), Extended Delta-Bar-Delta (DDBD), and Quick Propagation (QP). Also, cross-validation was used in this work. As a result, QP showed the best performance and had good effective training for MLPNN, and it had the ability to diagnosis diabetes sickness (Guler and Ubeyli, 2006).

A PSO-based classifier according to predefined data separation method was proposed for the classification of a breast cancer dataset. This method achieved 100% accuracy in both training and testing which was considered a good classification performance (Tewolde and Hanna, 2007).

Gu et al. explained that standard PSO could be trapped into local minima or lose its diversity. And also they showed that improved PSO interrupted the best position by a random function to increase the diversity of population without changing the best position of all particles. This improvement provided that PSO produced better convergence performance and made more efficient and effective in training ANNs (Gu et al., 2009).

The performance of a Simulated Annealing (SA) technique modified by Particle Swarm Optimization (PSO) for ANN training was compared with an Improved PSO based on ANN. The results showed that SA-PSO had a great performance in training an ANN compared to that of IPSO based ANN (Da and Xiurun, 2005).

The study emphased on evolving ANN's behaviors using Fogel's evolutionary programming (EP) using medical diagnosis problems, Australian credit card assessment problem and Mackey-Glass time series prediction problems. The results showed that improved algorithm had a good ability in classification, diagnosis, and prediction (Yao and Liu, 1997).

Ensemble Neural Network as the core of the proposed system was introduced to diagnose heart disease. The obtained result was 97.4% accuracy for the classification of the valvular heart disease dataset, 100% sensitivity and 96% specificity (Das et al., 2009).

The authors introduced Linear Discriminant Analysis (LDA) and Morlet Wavelet Support Vector Machine Classifier (LDA-MWSVM) for Diabetes automatic diagnosis system to find the most accurate method among the proposed methods. The results showed that LDA and LDA-MWSVM classifier learning technique was good enough for diabetes diagnosis with a good classification accuracy of 89.74% (Calisir and Esin, 2011).

A new learning system based on Generalized Discriminant Analysis (GDA) and Least Square Support Vector Machine (LS-SVM) was proposed. This learning system and the standard LS-SVM were used to diagnose diabetes disease, and the results were

gathered from them showed that GDA-LS-SVM had better accuracy (82.05%) than LS-SVM (78.21%) (Polat et al., 2008).

Principal Components Analysis (PCA) and Adaptive Neuro-Fuzzy Inference System (ANFIS) were used for the diagnosis of that disease using the diabetes dataset taken from the University of California Irvine UCI Machine Learning Repository. The method's classification accuracy was about 89.47% which was considered very promising compared to several classification techniques (Polat and Gunes, 2007).

Fuzzy threshold entropy based on feature relevance measurement was presented and used to classify five benchmarks medical datasets which were taken from the University of California Irvine UCI Machine Learning Repository. This method was explained as it was capable of making good classification results with a lower amount of features (Jaganathan and Kuppuchamy, 2013).

ANN generated by Genetic Programming (GP) was developed to work with graph structures. The results showed that this method had better results than some of the already existed techniques (Rivero and Periscal, 2009).

Three neural networks Multilayer Perceptron (MLP), Radial Basis Function Network (RBF) and Feedforward Neural Network were used for the weather prediction. The results showed that the Feedforward Neural Network was the most accurate among the three (Abdul-Kader, 2009).

Particle Swarm Optimization Neural Network (PSONN) was applied for a rainfall-runoff relationship. The results showed that PSONN made good accurate results and proved its ability in runoff prediction with good accuracy (Kuok et al., 2010).

A methodology based on PSO, Second Generation PSO (SGPSO), and New Model of PSO (NMPSO) was presented to design Artificial Neural Network (ANN) automatically. The fitness functions were based on Mean Square Error (MSE) and Classification Error (CER). The presented methods' performance was compared to the proposed method showed promising performance in designing ANN compared to ANN trained with BP (Garro et al., 2009).

2. MATERIALS AND METHODS

In this chapter, we explain the main focus of this work, where the data was collected, and which algorithms were used with ANN to improve its performance.

2.1. Data Resource.

In this study, two classification datasets were taken from the UCI KDD Data respiratory on 22nd March 2017. These datasets were used to compare and to show the performance of developed hybrid algorithms. The description of these datasets was given in Table 2.1 (http://archive.ics.uci.edu/ml/index.php).

Table 2.1. Description of datasets.

Dataset	Number of instances	Number of features	Number of classes
Pima Indians Diabetes	768	8	2
Hepatitis	155	19	2

2.1.1. Pima Indians Diabetes.

The Pima Indian diabetes database from UCI (https://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/), given by Vincent Sigillito, is a collection of medical diagnostic reports of 768 samples. It was taken from the National Institute of Diabetes and Digestive and Kidney Diseases (NIKDD). The dataset of Pima Indians Diabetes contains 768 sample described by 8 features to predict the presence or absence of diabetes. The features are in the following order:

- Number of pregnancies
- Plasma glucose concentration
- Diastolic blood pressure
- Triceps skin fold thickness
- Serum insulin
- Body mass index
- Diabetes pedigree function
- Age in years

2.1.2. Hepatitis

The Hepatitis dataset was collected from the Carnegie- Mellon University (https://archive.ics.uci.edu/ml/datasets/hepatitis). The classes of this dataset are “live” or “die.” It contains 155 samples described by 19 features. The features are in the following order:

- Age.
- Sex.
- Steroid.
- Antivirals.
- Fatigue.
- Malaise.
- Anorexia.
- Big liver.
- Liver film.
- Palpable spleen.
- Spiders.
- Ascites.
- Varices.
- Bilirubin.
- Alk phosphate.
- SGOT.
- Albumin.
- Protime.
- Histology.

2.2. Artificial Neural Networks (ANN)

Artificial Neural Network inspiration is based on the human brain's neuron structure. Just like the human brain, it depends on experience to learn. This mechanism is, in general, is similar to the way that a human learns which makes this technique very important and advanced in the artificial intelligence world. In the artificial neural network, its training can be compared to different forms of impressions that a human has like forgetting, learning, reacting, and behaving. Taking into account that different types of networks are based on the layers activity, a simple type network has referred

the network that contains the hidden layers are free to form their own representation of the input. The activation of the hidden layers is decided by the weights between the hidden layer and the input layer. The weights adjustment makes the hidden layer to select what is going to be represented. Another form of architectures are single layer architecture and multilayer architecture. In single layer architecture, all layers are connected to each other which network has consisted of inputs and outputs. In multilayer architecture, all units are in different layers input, hidden, and output layer. (Engelbrecht, 2007).

One type of multilayer architecture is Multilayer Perceptron (MLP). This network has one path for the signal to flow on through the network. Each neuron uses a non-linear activation function apart from the input neurons. (Engelbrecht, 2007).

2.2.1. Structure of Artificial Neural Networks

The neural network consists of three layers: the input layer, hidden layers, and the output layer.

- The input layer: the beginning of ANN which represents the provided raw data to the network.
- The hidden layers: (One or multi layers) set between the input and the output layer where these layers work on the basis of the input data.
- The output layer: the last layer of ANN which its output represents the output of ANN.

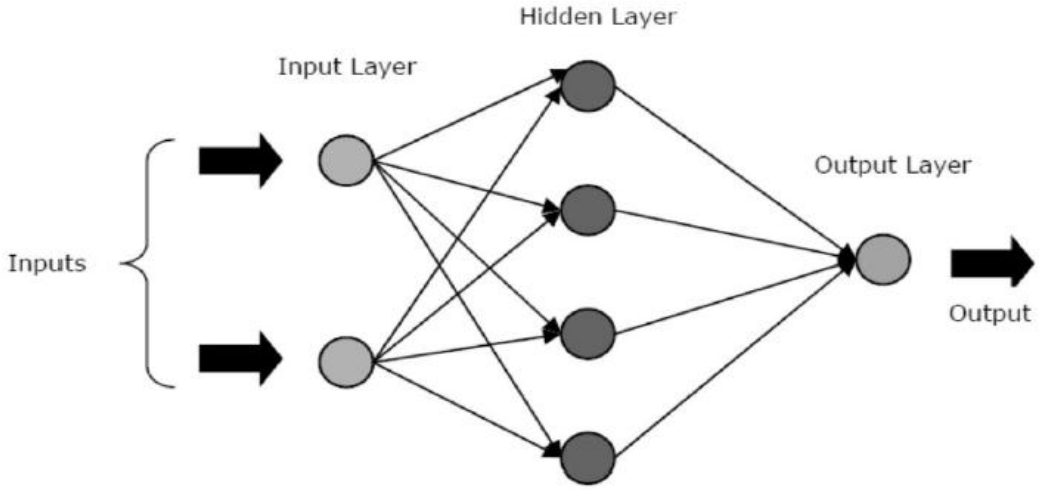

Figure 2.1 General neural network architecture

The first layer in the artificial neural network is used for receiving information (data), signals or features. In general, the inputs of ANN are usually normalized inside the limit value depending on the activation functions. The main reason for the normalization is to get better results in the form of numerical precision for the mathematical operations.
Weights are the form of parameters that connect input, hidden and output layers of ANN and modified for the data that feed into the network. This modification considered as "learning" for the network. The weights value, either positive or negative, will affect the ability of ANN (Engelbrecht, 2007).

The final layer in ANN, which is accountable for display and generating the final network outputs, can result in the processing to complete the use of the neurons in the past layers. The main architectures of artificial neural networks, considered according to neuron arranging, how they are interconnected and how its layers are composed (Engelbrecht, 2007).

2.3. Multilayer Neural Network (MLP)

The main structure of this type of network consists of input, output, and hidden layers. The hidden layers set in between input and output layers which usually considered as external input and in the form of network output for the input layer in some cases. After the signal goes into the input layer, the output of this layer goes to the

second layer (considering the hidden layer is only one layer as an example). The output of the hidden layers is the input of the output layer (Engelbrecht, 2007).
The structure of the multilayer feed forward network is in figure 2.1.

The characterization of feed-forward networks:

a. Typically, the general architecture which is the flow of the activation is from input to the output layer through the hidden layers.
b. Input-output mapping is the mathematical implementation.
c. Backpropagation algorithm is the most common supervised training.
d. Have proven useful in many forms of applications as approximates of nonlinear functions and as pattern classificatory.

2.4. Metaheuristic Inspired algorithms

In this section, we explain how PSO and AAA work, from where their inspiration, their characteristics, their parameters, and their abilities, and how the used to train of MLP.

2.4.1. Particle Swarm Optimization (PSO)

One of the population-based heuristic methods is the Particle Swarm Optimization (PSO) which was first developed by Kennedy and Eberhard (1995) by the behavior of birds or fish swarms in finding food Inspired. Each bird in the bird's wheel, which forms the basis of the method, refers to an individual solution and is called a "particle." All of the particles can be optimized by the fitness function (or quality) values evaluated and the flight of particles (velocities) that direct the search (research).

In computer science, particle swarm optimization (PSO) is a computational Technique that optimizes a problem using iteratively trying to make a better candidate solution with a precise of deference to a definite measure of quality. PSO can solve a problem using particles which are candidate solutions in the form of the -the change on the position (according to the velocity of the particle) of the particles within the search-space range is a form of a solution. The best position of a specific set of positions considered as its local best where each particle’s movement is influenced by it and also

guided toward the best positions within the range in the search-space. This makes the swarm proceed toward the best solutions (Eberhart and Kennedy, 1995).

PSO algorithm initialization is some random particles which are considered as the first and the best solution. After the initialization, PSO starts to search for the optimal solution using particle position update. ***pbest*** (Best solution for each particle in a generation) and ***gbest*** (Global best solution) are PSO's special particles. By updating them PSO can search for its best solution. Figure 2.2 shows the updating procedure of a particle.

The PSO algorithm initially generates a random set of particles (solutions) and then continues to develop through the generations continues. In each iteration, each particle (solution) is evaluated by two best values developed by interchanging and one step further to the point of global best solution approach is provided in Figure 2.2.

Figure 2.2 shows that the variable V_{ij}^{k} is the velocity of the particle and the variable X_{ij}^{k} is the position of the particle where *i* is the particle within the range of population *N* and *j* is the position of a particle within the range of dimension (swarm size) *d* at iteration *k* (Yalcin et al., 2015).

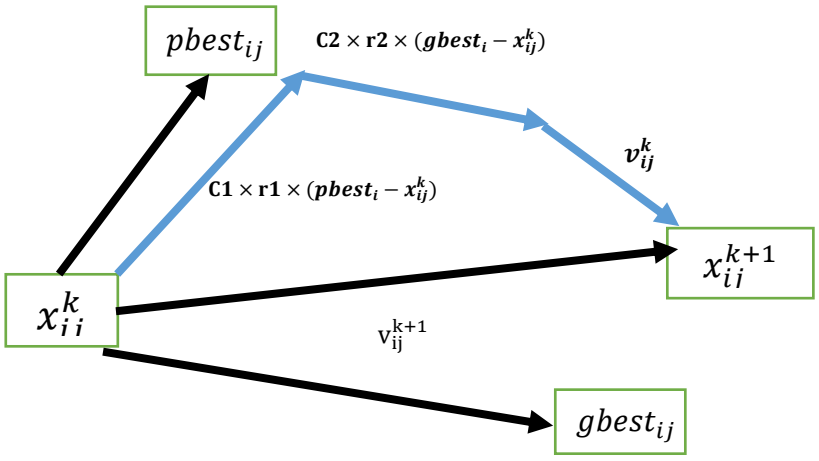

Figure 2.2. The velocity and position updating of a particle.

The particles in the swarm and their velocities are denoted by d-dimensional arrays as shown in Eqs. (2.1) and (2.2), respectively. *N* indicates the total number of particles in the lot.

$$\mathrm{Xi} = (X_{i1}\ X_{i2}\ X_{i3}\ \dots\ X_{id}) \qquad i = 1, 2, 3, \dots, N \tag{2.1}$$

$$\mathrm{Vi} = (V_{i1}\ V_{i2}\ V_{i3}\ \dots\ V_{id}) \qquad i = 1, 2, 3, \dots, N \tag{2.2}$$

Accordingly, the positions (2.3) and velocities (2.4) of N particles traveling in the d-dimensional search space are expressed by the following matrices:

$$\mathrm{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{1d} \\ x_{21} & x_{22} & x_{23} & x_{2d} \\ \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & x_{N3} & x_{Nd} \end{bmatrix} \tag{2.3}$$

$$\mathrm{V} = \begin{bmatrix} v_{11} & v_{12} & v_{13} & v_{1d} \\ v_{21} & v_{22} & v_{23} & v_{2d} \\ \dots & \dots & \dots & \dots \\ v_{N1} & v_{N2} & v_{N3} & v_{Nd} \end{bmatrix} \tag{2.4}$$

$$v^{k+1} = v_{ij}^{k} + c1\ r1\ \times \left(pbest_{ij} - x_{ij}^{k}\right) + c2\ r2\ \times \left(gbest_{i} - x_{ij}^{k}\right) \tag{2.5}$$

$$x^{k+1} = x^{k} + v^{k+1} \tag{2.6}$$

The velocity and position of each particle are updated with equations (2.5) and (2.6). In these equations; k number of iterations, r1 and r2 are normal random numbers in the range [0, 1], c1 and c2 are learning constants, v_{ij} velocity, and x_{ij} position value.

$$pbest_{ij} = (pbest_{i1} pbest_{i2} pbest_{i3} \dots pbest_{id}) i = 1,2,3, \dots N \tag{2.7}$$

$$\mathrm{Pbest} = \begin{bmatrix} \mathrm{Pbest_{11}} & \mathrm{Pbest_{12}} & \mathrm{Pbest_{13}} & \dots & \mathrm{Pbest_{1d}} \\ \mathrm{Pbest_{21}} & \mathrm{Pbest_{22}} & \mathrm{Pbest_{23}} & \dots & \mathrm{Pbest_{2d}} \\ \dots & \dots & \dots & \dots & \dots \\ \mathrm{Pbest_{N1}} & \mathrm{Pbest_{N2}} & \mathrm{Pbest_{N3}} & \dots & \mathrm{Pbest_{Nd}} \end{bmatrix} \tag{2.8}$$

$$gbest = (gbest_1 gbest_2 gbest_3 \dots gbest_d) i = 1,2,3, \dots N \quad (2.9)$$

The local best position is represented by a *d*-dimensional array for each particle (2.7), and is represented by an $N \times d$ matrix (2.8) for all particles. The global best position is represented by a d-dimensional array (2.9). The following pseudo code was taken from (Yalcin et al., 2015).

The algorithm pseudo code is the following:

Initialize the particle with random values

Do

For** each particle **do

Calculate fitness value of the particle

If fi**tness value of the current particle < fitness value of the pbest particle **then

Update the pbest particle

End if

End for

Gbest = the particle whose fitness value is equal to min (fitness values of all particles)

For** each particle **do

Calculate the particle velocity

Update velocity and position of the current particle according to equations (2.5) and (2.6)

End for

***While** stop criterion (maximum generation number or target fitness value of the gbest particle) is provided*

2.4.1.2. Particle Swarm Optimization Parameter Check

The parameters in our implementation were swarm size, number of iterations, acceleration coefficients and inertia weight. These parameters might have a good strong influence or no influence depending on the problem.

Swarm size (N) is considered as the number of particles in the swarm which the big sized holds more space in search space and increases algorithm's exploration abilities. This increases the possibility to gain a good solution with less iteration in a good amount of time within small swarm size. Technically, a large swarm size might improve

the computational time per iteration. However, there is not enough theoretical analysis about defining a good swarm size. In some of the algorithm versions, swarm size (depending on its dimension) calculation can be done automatically using the following equation:

$$N = 10 + |2\sqrt{D}| \qquad (2.10)$$

where D is the dimension of the problem and N is the swarm size for the optimization. Unfortunately, this formula does not give the optimal swarm size and usually gives results far from it. Therefore, manual setting is considered a good solution according to the problem.

Number of iterations (MaxIte): according to the complexity of the optimization problem, the maximum iteration number can be chosen. The larger the number, the higher is the possibility of gaining optimal solution. This also means the possibility of converging the optimal solution due to the additional computation. In this implementation maximum iteration is used as a stopping criterion. And the reason is that the optimal value isn't known. So, the maximum iteration number is set on a high value to improve the possibility to obtain a good solution and avoid premature convergence.

Acceleration coefficients (c1, c2): Cognitive and social components' stochastic influence depends on c1 and c2, which cognitively are responsible for particles move towards the best personal positions while socially help particles to move to the global best position for all particles. c1 affects the personal best positions meanwhile c2 affects the global best. This means if c1 is higher than c2, then the particles are going to be affected by the personal best position; otherwise, all particles are going to be affected by the global best as long as they are not equal.

Acceleration coefficients are usually set according to the following equation:

$$c1 + c2 = 4 \qquad (2.11)$$

The set c1 = c2 = 2 is considered as a good choice, and the reason is that particles are attracted towards the average of their personal best positions and global best position.

2.4.2. Artificial Algae Algorithm (AAA)

Artificial Algae Algorithm (AAA) is a metaheuristic algorithm inspired by microalgae's living behavior. Artificial Algae can look for any solution for a specific problem due to its characteristics similarity with real algae. AAA uses environment adaptation and helical swimming to move to the light source just like real algae because of mitotic division which provides good capability.

AAA based on three important phases as a major generation. These phases are Helical Movement, Evolutionary Process, and Adaptation phase. The population of AAA is based on the sets of algal cells that held together in the form of the cluster called colony, which lives or dies according to the living situation just like one cell. Unsuitable conditions like sheer force make the colony dispensed and create new colony from each dispersed part which a colony is situated at the ideal point otherwise its considered as ideal algal cells (Uymaz et al., 2015).

$$Population\ of\ Algal\ Colony = \begin{bmatrix} X_1^1 & -- & X_1^D \\ | & & | \\ X_N^1 & -- & X_N^D \end{bmatrix} \tag{2.13}$$

$$i^{th}\ algal\ colony = \left[X_i^1, X_i^2 \ldots\ldots\ldots X_i^D\right] \tag{2.14}$$

Where X_i^j algal cell in j^{th} dimension of i^{th} algal colony.

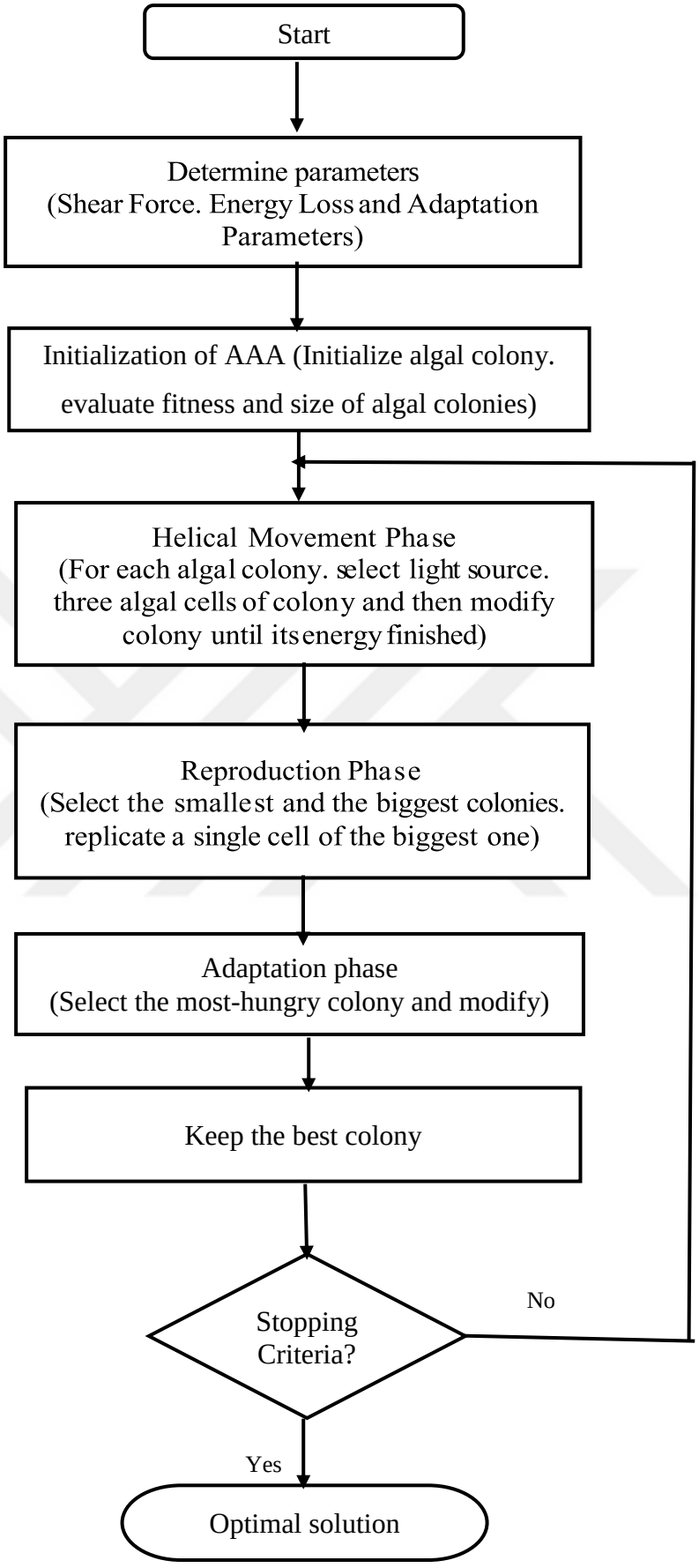

Figure 2.3. General flowchart of the AAA.

2.4.2.1. Helical movement

Obviously, because the surface of the water is lighter than other places in the water, Algae cells and colonies basically try to be as close as possible to the surface of the water. Algae cells and colonies movement is limited because of the gravity and viscous tow, but they swim helically to restrict both gravity force and viscous tow which is different from one algae cell to another. The helical movements' frequency generally increases according to surface of the growing for algae cells which depends on local search capacity. The closer the algae cell to the surface, the more energy it has due to algae's proportional movement towards its energy. This gives the algae greater chance to move within the liquid. On the other hand, it takes them a long time to travel to light if the light coverage on the surface of the water is low. This means there is more global search capacity. In AAA, sheer force is considered as the viscous tow and 0 for motion restriction, where sheer force is related to the size of the algae cell. The shape of algae cell is spherical and the surface area is hemispheric (Eq. (2.15) and (2.16)). (Uymaz et al., 2015).

$$\tau\,(X_i) = 2\pi r^2 \tag{2.15}$$

$$\tau\,(X_i) = 2\pi \left(\sqrt[3]{\frac{3G^i}{4\pi}} \right)^2 \tag{2.16}$$

Where the friction surface is τ (xi). Helical movement dimensions of the algae cell are randomly determined. One of them is for linear motion in eq. (2.17) and angular movement is provided by the other two dimensions which are Eq. (2.18) and (2.19). Eq. (2.17) used for single-dimensional problems and algae cell/colony moves in one direction. In two-dimensional problems, the movement of algae is sine Eq. (2.17) and Eq. (2.18). In the case of three or more dimensions, the movement of algae is helical and Eq. (2.17) - (2.18). Friction surface and distance to the light source select the step size of movement: (Uymaz et al., 2015).

$$X_{im}^{t+1} = X_{im}^{t} + \left(X_{jm}^{t} - X_{im}^{t}\right)(\Delta - \tau^{t}(X_i))P \tag{2.17}$$

$$X_{ik}^{t+1} = X_{ik}^{t} + \left(X_{jk}^{t} - X_{ik}^{t}\right)(\Delta - \tau^{t}(X_i))Cos\,\alpha \tag{2.18}$$

$$X_{il}^{t+1} = X_{il}^{t} + \left(X_{jl}^{t} - X_{il}^{t}\right)(\Delta - \tau^{t}(X_i))Sin\,\beta \tag{2.19}$$

Where X_{ik}^{t} , X_{il}^{t} , and X_{im}^{t} are x, y and z coordinates of i^{th} algal cell at time t; $\alpha, \beta, \in [0,2\pi]; P \in [-1,1]$; Δ is shear force; $\tau^{t}\ (X_i)$ is the friction surface area of i^{th} algal cell.

2.4.2.2. Evolutionary process

Under enough nutrient conditions, if the algal colony extradites sufficient light, it emerges and increases itself to generate two new algal cells in time t, analogous to the real mitotic division. However if the algal colonies do not receive sufficient light, it will survive for a while till finally dies. The algal colony's growth kinetics was computed with the following given Monod model (Eq. 2.20) (Uymaz et al., 2015).

$$\mu = \frac{\mu_{max} S}{K_s + 5} \tag{2.20}$$

Where, the specific growth rate is represented as μ, the maximum specific growth rate is μ_{max}, S is the nutrient concentration, that the fitness value is $\left(f^{i}(X_1)\right)$ in time t in the model and K is the substrate half saturation constant of the algal colony. μ_{max} Was presumed as 1. K was computed as the growth rate at half nutrient conditions of algal colony in time t. The size of i^{th} algal colony in time $t + 1$ in Monod equation is given in the following equation (Uymaz et al., 2015).

$$G_i^{t+1} = \mu_i^t\ G_i^t \quad i = 1, 2, \ldots\ldots\ldots, N. \tag{2.21}$$

Where, the specific growth rate is represented as, the maximum specific growth rates is, nutrient concentration is S, that the fitness value is in time t in the model and K is the substrate half saturation constant of the algal colony. Was presumed as 1, K was computed as the growth rate at half nutrient conditions of algal colony in time t. The size of algal colony in time in Monod equation is given in the following equations :(Eq. (2.22)-(2.24)).

$$biggest^t = \max G_i^t \qquad i = 1,2, \ldots\ldots\ldots, N; \tag{2.22}$$

$$smallest^t = \min G_i^t \qquad i = 1,2, \ldots\ldots, N; \tag{2.23}$$

$$smallest_m^t = biggest\ _m^t \qquad m = 1,2, \ldots\ldots, N \tag{2.24}$$

Where the problem dimension is D, the biggest and the smallest algal colonies are called is biggest and the smallest, respectively. In AAA, algal colonies are arranged according to their sizes in time t. In any randomly chosen dimension, the smallest algal colony in algal cell died and algal cell of the biggest colony regenerate itself (Uymaz et al., 2015).

2.4.2.3. Adaptation

Algal colony cannot grow enough in an environment that helps to make conform itself to the environment which results to the dominant type modification. The biggest algal colony in the environment identified as the most grown algal colony. This process finishes with the modification in starvation level in the algorithm. The initial starvation value is zero for any artificial algae. Starvation value is increased with time t, when the algal cell receives scanty light. The artificial alga having the highest starvation value (Eq. (2.25)) has adapted with Eq. (2.26).

$$Starving^t = \max A_i^t \qquad i = 1,2,\dots\dots\dots..N; \tag{2.25}$$

$$Starving^{t+1} = Starving^t + (biggest^t - Starving^t) * rand. \tag{2.26}$$

Where the starvation value is A_i^t of i^{th} algal colony in time t, the algal colony is starving with the highest starvation value in time t. Adaptation process is determined by the adaptation parameter (Ap) which discovers the probability to be applied in time t or not. Constant on the interval [0, 1] is (Ap).

2.5. Performance Criteria

In this study, MSE, Specificity, Sensitivity, and accuracy were preferred to compare the performance of both PSO and AAA with MLP.

The mean squared error (MSE) (Eq. (2.27) (to make an unobserved quantity estimate) measures the mean squares of errors or deviations -that is, the difference between the target and output of the classifier.

$$MSE = \frac{1}{output\ size} \sum_{i=1}^{output\ size} (output_i - Targets_i)^2 \quad (2.27)$$

The predictor's MSE computed as (2.27) as long as ***Targets*** is a vector of *n* predictions and ***output*** is the vector of observed values of the predicted value.

To determine the accuracy of the classification process, sensitivity and specificity have to be calculated. These concepts the equations (2.28) and (2.29) are used in the calculations. In these equations, definitions of parameters:

- TP (True Positive): Patient diagnosed as a patient,
- TN (True Negative): the patient is not ill, the diagnosis is made,
- FP (False Positive): Diagnosis of a patient who is not a patient,
- FN (False Negative): The patient is not a patient but a diagnosed condition counts.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (2.28)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (2.29)$$

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (2.30)$$

Sensitivity refers to patient-related data; the specificity belongs to the non-sick person estimate of the data.

3. APPLICATION

In this chapter, we show the usage of Multi-Layer Perceptron (MLP) learning by Particle Swarm Optimization (PSO) and Multi-Layer Perceptron (MLP) learning by artificial algae algorithm (AAA) and how test results affected by the training that is done by Particle Swarm Optimization (PSO) or Artificial algae algorithm (AAA).

3.1. Multi-Layer Perceptron with Particle Swarm Optimization (MLP-PSO)

One of the most famous heuristic optimization methods is PSO which was developed by (Eberhart and Kennedy, 1995), In the training of artificial neural networks with the particle swarm optimization (improve the weight and bias of neural networks by particle swarm optimization algorithm) we can obtain the best results and less errors and more accuracy after comparing the output of artificial neural networks for each iteration in PSO to the targets of a supervised data, The steps of the algorithm are as follows:

1) According to the d-dimension space, determination of position and velocity initialization can be done.
2) Training the PSO- MLP by using the particles' positions and determine MSE (particle fitness) for each particle.
3) The current position and fitness achieved by particle p that set as its best history amount also called the personal best (pbest). The pbest with best value in all particles are set as global best (gbest).
4) Change the velocity of the particle according to Equation (2.5).
5) Update particle position by adding the calculated velocity value to the current position value according to Equation (2.6).
6) Use the new sets of positions to generate new learning error.
7) Compare the MSE of each particle with its pbest MSE then update the pbest, if the current MSE is lower than the pbest MSE.
8) Finding the minimum calculated MSE in the swarm then comparing it by the global best MSE then updating gbest, if the minimal MSE is lower than gbest MSE.
9) The optimization output is based on gbest position value. The iteration loop continues until reaching the MSE of the gbest lower than the desired threshold or a

maximum iteration number. The gbest weights are used as the training results when the iteration is finished.

The algorithm pseudo code of MLP-PSO is the following:

For *each particle* ***do***
Initialize the particle with random values
Start training and present the training of the MLP
End for
Do
For *each particle* ***do***
Calculate fitness value of the particle
If fi*tness value of the current particle < fitness value of the pbest particle* ***then***
Update the pbest particle
End if
End for
Gbest = the particle whose fitness value is equal to min (fitness values of all particles)
For *each particle* ***do***
Update velocity and position of the current particle
End for
While *stop criterion (maximum generation number or target fitness value of the gbest particle) is provided*
Terminate the training and start testing
Present test data and compute results.

Figure 3.1 shows the MLP training using PSO which explains the weights and biases optimization on MLP by PSO.

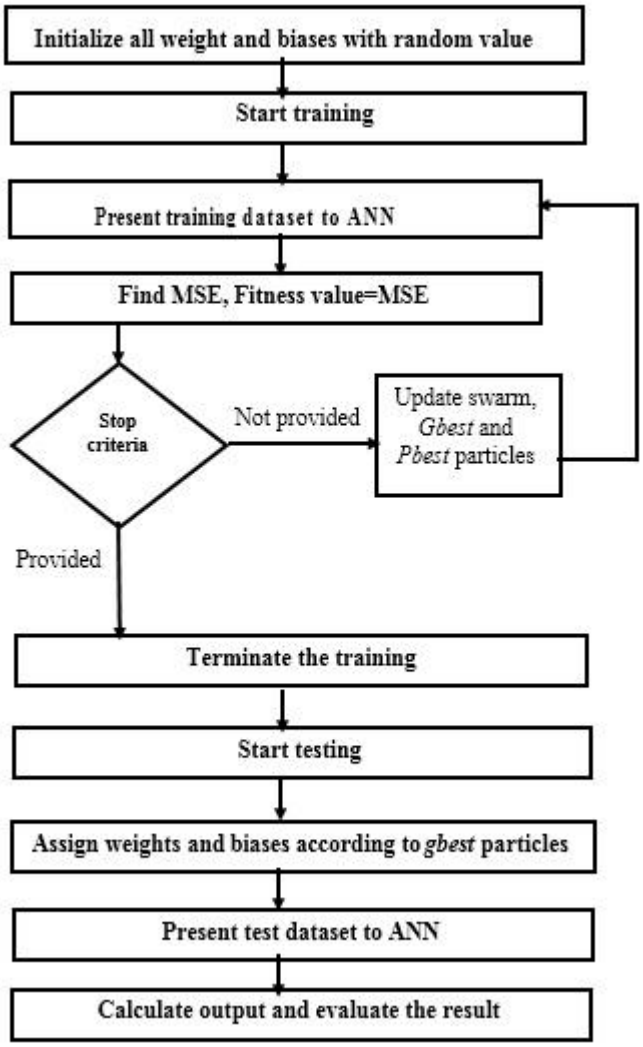

Figure 3.1. Flowchart for the training and testing of the MLP-PSO.

3.2 Multi-layer perceptron with Artificial Algae Algorithm (MLP-AAA)

Artificial algae algorithm (AAA), (Uymaz et al., 2015) developed as bio-inspired optimization algorithms, has been inserting by inspiration of living behavior of microalgae. AAA was presented recently, which has been successfully applied to solve various continuous optimization problems. Therefore, in this study, the AAA was adopted to optimize the training of MLP. Figure 3.2 shows the flowchart of MLP training using the AAA algorithm.

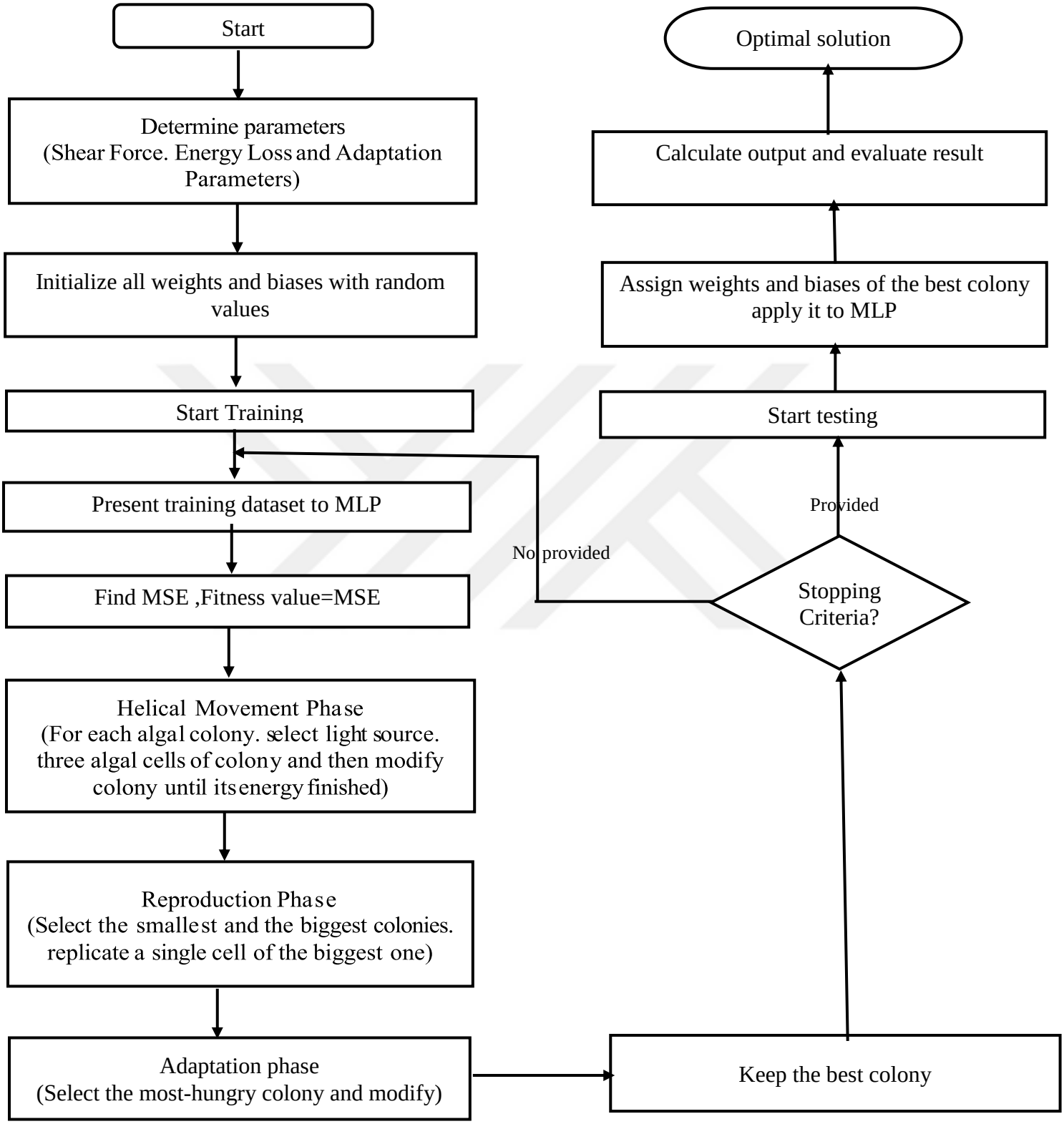

Figure 3.2 AAA based MLP training.

3.2.1. MLP using AAA Step's

1) Initialize and discrete population fitness and size of each colony.
2) Training the AAA- MLP by using the algae's position and determine MSE (fitness) for each colony.
3) The current positions and fitness achieved by algal colonies are set as its best history amount.
4) Calculate the MSE of each colony.
5) Selecting 3 algal cells (k, l, and m) randomly and modifying them using the following equations (2.17), (2.18) and (2.19).
6) Checking the new solution. If the new solution better to move the algae's to a new position or keep the old solution if not.
7) Selecting the smallest and the biggest colonies and making a random algae cell to from the smallest to be replicated from the biggest.
8) Selecting the hungriest colony and modifying the colony.
9) Terminating the training and start testing.
10) Presenting test data to MLP and computing results.

The algorithm pseudo code of MLP-AAA is the following:

For *each colony* ***do***
Initialize the colony with random values
Start training and present the training of the MLP
End for
Do
For *each colony* ***do***
Calculate fitness value of the colony
If *the new fitness value of the current colony < old fitness value of the current colony* ***then***
Update the fitness colony
End if
End for
For *each colony* ***do***
Selecting 3 algal cells (k, l and m) randomly and modifying them using the following equations.

$$X_{im}^{t+1} = X_{im}^{t} + \left(X_{jm}^{t} - X_{im}^{t}\right)(\Delta - \tau^{t}(X_i))P$$

$$X_{ik}^{t+1} = X_{ik}^{t} + \left(X_{jk}^{t} - X_{ik}^{t}\right)(\Delta - \tau^{t}(X_i))Cos\,\alpha$$

$$X_{il}^{t+1} = X_{il}^{t} + \left(X_{jl}^{t} - X_{il}^{t}\right)(\Delta - \tau^{t}(X_i))Sin\,\beta$$

Selecting the smallest and the biggest colonies and making a random algae cell to from the smallest to be replicated from the biggest.

$$biggest^{t} = \max G_i^{t} \qquad i = 1,2,\ldots\ldots\ldots,N;$$

$$smallest^{t} = \min G_i^{t} \qquad i = 1,2,\ldots\ldots.,N;$$

$$smallest_m^{t} = biggest\,_m^{t} \qquad i = 1,2,\ldots\ldots.,N;$$

Selecting the hungriest colony and modifying the colony.

$$Starving^{t} = \max A_i^{t} \qquad i = 1,2,\ldots\ldots\ldots..N;$$

$$Starving^{t+1} = Starving^{t} + (biggest^{t} - Starving^{t}) * rand.$$

End for

While *stop criterion is provided*

Terminate the training and start testing

Present test data and compute results

4. RESULTS

This chapter presents the simulation results for MLP-PSO and MLP-AAA on two research datasets, hepatitis and Diabetes Datasets. We already discussed the properties and structure of these datasets. We implemented the MLP using PSO and AAA algorithms to train and test these research datasets in order to measure the different performance parameters. In section 4.1, the comparative results are discussed. In section 4.2 recommendation based on current results are discussed.

The number of neurons in the hidden layer was set as 5 neurons for both MLP-PSO and MLP-AAA. Sigmoid activation function was used in all neurons.

Therefore, the effect of target output values for MLP-PSO and MLP-AAA was investigated. When neural networks supported by sigmoid activation function are used to solve classification problems, neural networks can never converge with the best range of weights if the targets were 0 & 1 because the weights are going to become extreme values until the training stops which make the output of neural network out of reach. For that reason, the 0 in targets usually changed to 0.1 and 1 changed to 0.9 (Engelbrecht, 2007). Because of this, different target values were selected as (0, 1), (0.1, 0.9), (0.2, 0.8) and (0.2, 0.6).

4.1. The Results of MLP-PSO

PSO algorithm has superior ability to search solution space. In MLP-PSO, the weights between the layers of the MLP are the particles of the PSO, which are optimized to make MLP has less error and higher accuracy. In this section, we collect the error, accuracy, specificity, and sensitivity of MLP-PSO on diabetes and hepatitis and compare it to other works.

4.1.1. The results of MLP-PSO for Diabetes Dataset

In this study, Diabetes dataset was used for the evaluation of MLP-PSO with 5 neurons in the hidden layer, 30 max run and 30 number of swarm size with 200 maximum iterations. C1 and C2 selected as 1.5 and 2.5. The lower and upper bounds (LB and UB) are determined as -2 and 2.

Firstly, MLP-PSO was carried out with all attribute of diabetes dataset with 5 neurons in the hidden layer for different target values both to investigate the effect of target values and to find the best structure.

By using all attribute of diabetes dataset for MLP-PSO, the results, as shown in Table 4.1, show that the selected target (0.2, 0.6) was the best solution.

Figure 4.1 indicates the average, the best and the worst training error values of 30 runs for MLP-PSO with 5 neurons in hidden layer, 30 particles, and 200 iterations for all the attributes of Diabetes dataset.

Table 4.1. Result of MLP-PSO using (All) attribute diabetes dataset with different of targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	All	All	All	All
TP	52	70	87	90
FP	38	20	3	0
TN	158	171	179	183
FN	25	12	4	0
Sensitivity	67.53	85.36	95.60	100
Specificity	80.61	89.52	98.35	100
Standard Deviation Training Accuracy	1.761706	1.416655	0.614222	0.257875
Min Training Accuracy	70.51	79.35	91.17	95.97
Max Training Accuracy	84.73	90.11	94.60	97.54
Mean Training Accuracy	83.78	89.27	94.22	97.37
Error is (Fitness/MSE)	0.158473	0.099236	0.057731	0.026970
Testing Accuracy	76.92	88.27	97.43	100

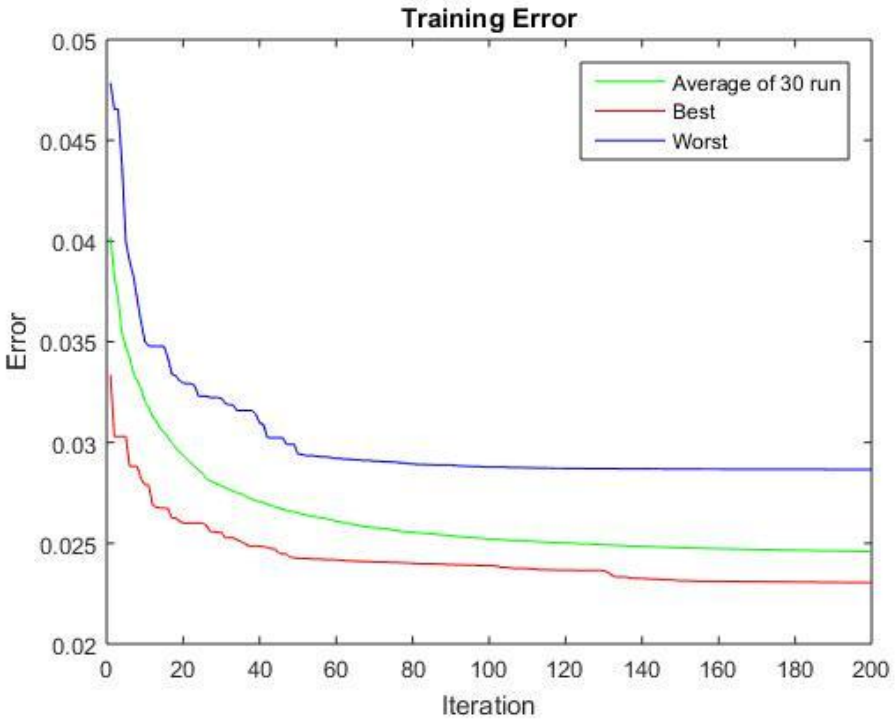

Figure 4.1. Training Error graph performance using MLP-PSO for Diabetes dataset.

The performance of MLP was tested by examining the effects of each attribute as indicated by (Jaganathan and Kuppuchamy, 2013). Initially all the attributes were tested as seen in Table 4.1, secondly, attributes 2 (plasma glucose concentration), 6 (body mass index) and 8 (age in years) were tested (Table 4.2) then attribute 1 (number of pregnancies) was added (Table 4.3), and finally attribute 7 (diabetes pedigree function) was added (Table 4.4).

Table 4.2. Result of MLP-PSO using attributes (2, 6, 8) diabetes dataset for different targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	2,6,8	2,6,8	2,6,8	2,6,8
TP	47	69	86	90
FP	43	21	4	0
TN	155	171	180	183
FN	28	12	3	0
Sensitivity	62.66	85.18	96.62	100
Specificity	78.28	89.06	97.82	100
Standard Deviation Training Accuracy	1.003917	0.563145	0.344917	0.224165
Min Training Accuracy	77.25	84.98	91.47	95.57
Max Training Accuracy	84.35	90.08	94.34	97.51
Mean Training Accuracy	83.80	89.83	94.16	97.40
Error is (Fitness/MSE)	0.157764	0.099181	0.056680	0.025963
Testing Accuracy	73.99	87.91	97.43	100

For all attribute combinations, Tables 4.1 to 4.4 indicated the selected target (0.2, 0.6) was the best solution.

Table 4.3. Results of MLP-PSO using attributes 1, 2, 6, 8 diabetes dataset for different targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	1, 2, 6, 8	1, 2, 6, 8	1, 2, 6, 8	1, 2, 6, 8
TP	47	60	87	90
FP	43	30	3	0
TN	151	171	179	181
FN	32	12	4	2
Sensitivity	59.49	83.33	95.60	97.82
Specificity	77.83	85.07	98.35	100
Standard Deviation Training Accuracy	1.524027	0.725300	0.454599	0.165285
Min Training Accuracy	71.94	85.25	91.30	96.38
Max Training Accuracy	84.52	90.19	94.42	97.53
Mean Training Accuracy	83.85	89.80	94.16	97.44
Error is (Fitness/MSE)	0.168792	0.098174	0.056746	0.028328
Testing Accuracy	72.52	84.61	97.43	99.26

Table 4.4. Results of MLP-PSO using attributes 1, 2, 6, 7, 8 diabetes dataset for different targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	1, 2, 6, 7, 8	1, 2, 6, 7, 8	1, 2, 6, 7, 8	1, 2, 6, 7, 8
TP	50	70	87	90
FP	40	20	3	0
TN	152	166	180	182
FN	31	17	3	1
Sensitivity	61.72	80.45	96.66	98.90
Specificity	79.16	89.24	98.3607	100
Standard Deviation Training Accuracy	1.581507	1.165112	0.541312	0.282017
Min Training Accuracy	75.56	81.19	90.97	95.87
Max Training Accuracy	85.36	90.40	94.68	97.65
Mean Training Accuracy	84.47	89.83	94.38	97.47
Error is (Fitness/MSE)	0.160435	0.105127	0.061748	0.025133
Testing Accuracy	73.99	86.44	97.80	99.63

4.1.2. The results of MLP-PSO for Hepatitis Dataset.

In this study, hepatitis dataset was used for the evaluation of MLP-PSO with 5 neurons in hidden layers, and 30 number of size swarm with 200 maximum iterations. C1 and C2 selected as 1.5 and 2.5. The lower and upper bounds (LB and UB) are

determined as -2 and 2. MLP-PSO was run 30 times using this dataset. The best, average and the worst training error values of 30 runs indicated in Figure 4.2.

By using all attribute of Hepatitis dataset for MLP-PSO, the results, as shown in Table 4.5, indicated that the selected target (0.2, 0.6) was the best solution.

The same work as diabetes dataset was performed with hepatitis. Feature selection was performed for hepatitis dataset as (Jaganathan and Kuppuchamy, 2013). Different attribute combinations were tested with MLP-PSO and the results were compared. First combination consists attributes 5, 6, 11, 12, 13, 14, 17, and 19. (Table 4.6) The second one is attributes 2, 5, 6, 10, 11, 12, 13, 14, 17, and 19 (Table 4.7). The third combination of hepatitis features is attributed (6, 11, 12, 13, 14, 17, and 19 (Table 4.8). Finally attributes 11, 12, 13, 14, 17, and 19 constitutes the last combination (Table 4.9) for all combinations, MLP-PSO illustrated the best performance for (0.2, 0.6) selected target values. In addition. MLP-PSO showed the highest performance with all attributes for hepatitis database.

Table 4.5. Results of MLP-PSO using (All) attributes of Hepatitis dataset for different targets.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	All	All	All	All
TP	36	38	40	40
FP	4	2	0	0
TN	7	8	8	9
FN	2	1	1	0
Sensitivity	94.73	97.43	97.56	100
Specificity	63.63	80	100	100
Standard Deviation Training Accuracy	2.391010	3.705282	1.149864	0.658525
Min Training Accuracy	75.47	62.95	88.70	93.34
Max Training Accuracy	95.52	95.18	97.28	98.91
Mean Training Accuracy	94.17	93.47	96.58	98.55
Error is (Fitness/MSE)	0.090320	0.072934	0.042848	0.018675
Testing Accuracy	87.75	93.87	97.95	100

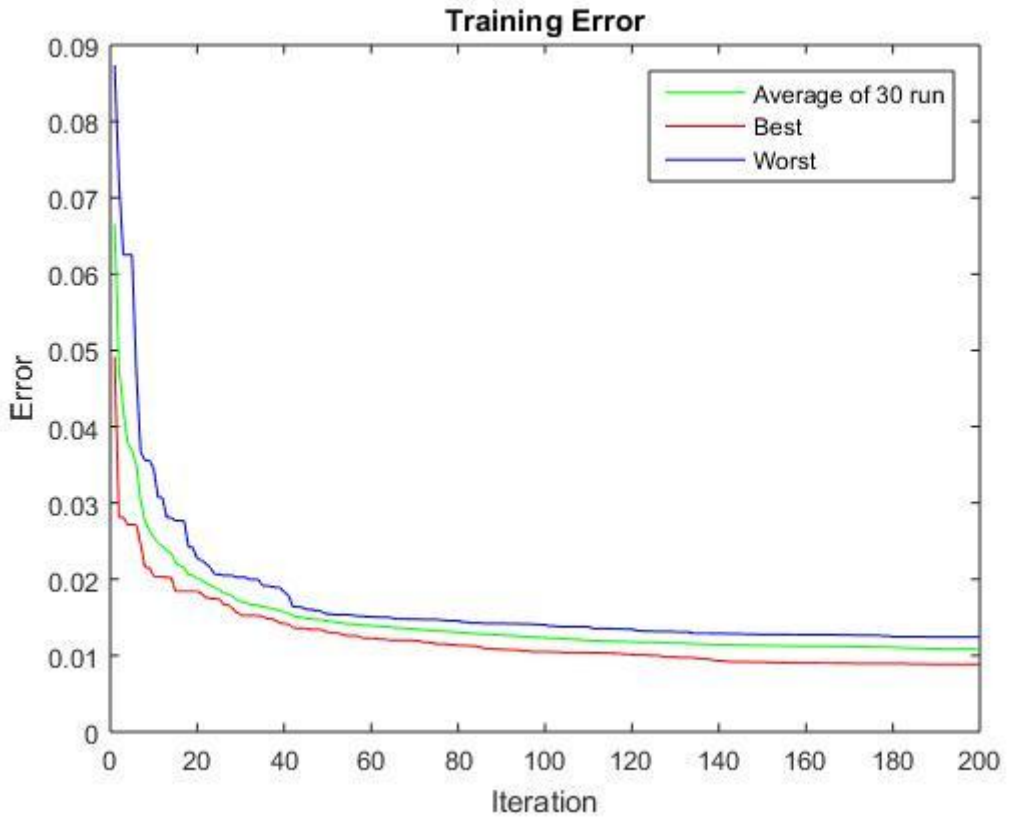

Figure 4.2. Training Error graph performance using MLP-PSO for Hepatitis dataset.

Table 4.6. Results of MLP-PSO using Hepatitis dataset attributes 5, 6, 11, 12, 13, 14 and 17 for different targets.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	(5, 6, 10, 11, 12, 13, 14, 17 and 19)	(5, 6, 10, 11, 12, 13, 14, 17 and 19)	(5, 6, 10, 11, 12, 13, 14, 17 and 19)	(5, 6, 10, 11, 12, 13, 14, 17 and 19)
TP	40	39	40	40
FP	0	1	0	0
TN	3	5	6	8
FN	6	4	3	1
Sensitivity	86.95	90.69	93.02	97.56
Specificity	100	83.33	100	100
Standard Deviation Training Accuracy	3.180322	1.829328	1.363597	0.831518
Min Training Accuracy	66.28	82.16	87.01	92.56
Max Training Accuracy	92.26	95.17	97.05	98.79
Mean Training Accuracy	90.54	94.08	96.33	98.33
Error is (Fitness/MSE)	0.096732	0.124655	0.052150	0.018177
Testing Accuracy	87.75	89.79	93.87	97.95

Table 4.7. Results of MLP-PSO using Hepatitis dataset attributes 2, 5, 6, 10, 11, 12, 13, 14, 17 and 19 for different targets.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	(2, 5, 6, 10, 11, 12, 13, 14, 17 and 19)	(2, 5, 6, 10, 11, 12, 13, 14, 17 and 19)	(2, 5, 6, 10, 11, 12, 13, 14, 17 and 19)	(2, 5, 6, 10, 11, 12, 13, 14, 17 and 19)
TP	39	39	40	40
FP	1	1	0	0
TN	4	5	6	8
FN	5	4	3	1
Sensitivity	88.63	90.69	93.02	97.56
Specificity	80	83.33	100	100
Standard Deviation Training Accuracy	3.344483	2.481696	0.975285	0.445343
Min Training Accuracy	68.33	74.15	88.62	95.63
Max Training Accuracy	94.92	91.96	96.76	98.72
Mean Training Accuracy	93.50	90.34	96.29	98.47
Error is (Fitness/MSE)	0.073987	0.076634	0.050962	0.022651
Testing Accuracy	87.75	89.79	93.87	97.95

Table 4.8. Results of MLP-PSO using Hepatitis dataset attributes 6, 11, 12, 13, 14, 17 and 19 for different targets.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	(6, 10, 11, 12, 13, 14, 17 and 19)	(6, 10, 11, 12, 13, 14, 17 and 19)	(6, 10, 11, 12, 13, 14, 17 and 19)	(6, 10, 11, 12, 13, 14, 17 and 19)
TP	37	38	38	40
FP	3	2	2	0
TN	5	6	9	8
FN	4	3	0	1
Sensitivity	90.24	92.68	100	97.56
Specificity	62.50	75	81.81	100
Standard Deviation Training Accuracy	4.546717	2.119955	0.555411	1.508646
Min Training Accuracy	58.99	78.23	94.45	85.09
Max Training Accuracy	93.15	94.83	98.73	97.13
Mean Training Accuracy	90.74	93.81	98.42	96.44
Error is (Fitness/MSE)	0.108835	0.073929	0.020758	0.039132
Testing Accuracy	85.71	89.79	95.91	97.95

Table 4.9. Results of MLP-PSO using Hepatitis dataset attributes 11, 12, 13, 14, 17 and 19 for different targets.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	(11, 12, 13, 14, 17 and 19)	(11, 12, 13, 14, 17 and 19)	(, 11, 12, 13, 14, 17 and 19)	(11, 12, 13, 14, 17 and 19)
TP	40	39	39	39
FP	0	1	1	1
TN	3	5	7	9
FN	6	4	2	0
Sensitivity	86.95	90.69	95.12	100
Specificity	100	83.33	87.50	90
Standard Deviation Training Accuracy	2.520248	1.800990	1.072093	0.420645
Min Training Accuracy	70.20	79.99	89.79	95.76
Max Training Accuracy	91.26	94.17	96.92	98.57
Mean Training Accuracy	90.05	93.22	96.32	98.32
Error is (Fitness/MSE)	0.084521	0.062116	0.047105	0.019908
Testing Accuracy	87.75	89.79	93.87	97.95

4.2. The Results of MLP-AAA

AAA algorithm, like PSO algorithm, has the ability to search solution space. The weights between the layers of the ANN are the algae's of the AAA. Therefore, the weight changed during MLP training updates, and helical movement phase in AAA the new weight values obtained by the collection are the result of the algal movement. They have expressed that their new positions will appear in algal positions change which refers to the change in weight. AAA's helical movement phase, adaption phase, and starving methods do different changes for the position of the algae which also make changes for the weights in MLP. The best weights and biases are considered as the optimum solution for AAA by calculating the least error of MLP which makes the output the closest to the targets.

The number of input neurons in the input layer and the output layer's neurons depends on the number of input and output attributes in datasets. But the number of hidden neurons was defined experimentally. In this study, 5 neurons in the hidden layer for one neuron in the output layer (the total number of iterations is 200), which were used for the obtained training and testing processes. The comparison of results was performed according to accuracy, find best fitness values, and calculate sensitivity and specificity, for maximum 200 iterations.

4.2.1. Results MLP-AAA with Diabetes Dataset

In this work, MLP-AAA was performed for diabetes dataset with 5 neurons in hidden layers, sheer force is 2, energy loss is 0.3, adaption is 0.2, 30 algae colony, 30 max run, and 200 max iterations.

Firstly MLP-AAA had carried out with all attributes of diabetes dataset with 5 neurons in hidden layers for different target values both to investigate the effect of target values and to find the best structure. The selected target values were (0, 1), (0.1, 0.9), (0.2, 0.8) and (0.2, 0.6).

For the same obvious reasons with MLP-PSO, the 0 in targets usually changed to 0.1 and 1 changed to 0.9.

By using all attributes of diabetes dataset for MLP-AAA, the results, as shown in Table 4.10, show that the selected target (0.2, 0.6) was the best solution.

Figure 4.3 shows the average, best and worst training error values of MLP-AAA for all Diabetes dataset for 30 runs,

To compare the performances of MLP-AAA and MLP-PSO. Same attribute combinations were applied for a diabetes database. And the results of all combinations for MLP-AAA showed in Figures 4.11- 4.13. In addition.

The best results of all data combinations were reached with the target values as (0.2, 0.6). However, the worst performance was obtained with (0 1) target values.

Table 4.10. Results of MLP-AAA using All attributes of diabetes dataset for different targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	All	All	All	All
TP	51	63	86	90
FP	39	27	4	0
TN	155	167	180	183
FN	28	16	3	0
Sensitivity	64.55	79.74	96.62	100
Specificity	79.89	86.08	97.82	100
Standard Deviation Training Accuracy	0.049406	0.052262	0.022813	0.011591
Min Training Accuracy	86.07	90.83	94.96	97.70
Max Training Accuracy	86.38	91.20	95.08	97.78
Mean Training Accuracy	86.34	91.16	95.06	97.77
Error is (Fitness/MSE)	0.163590	0.112026	0.054980	0.025334
Testing Accuracy	75.45	84.24	97.43	100

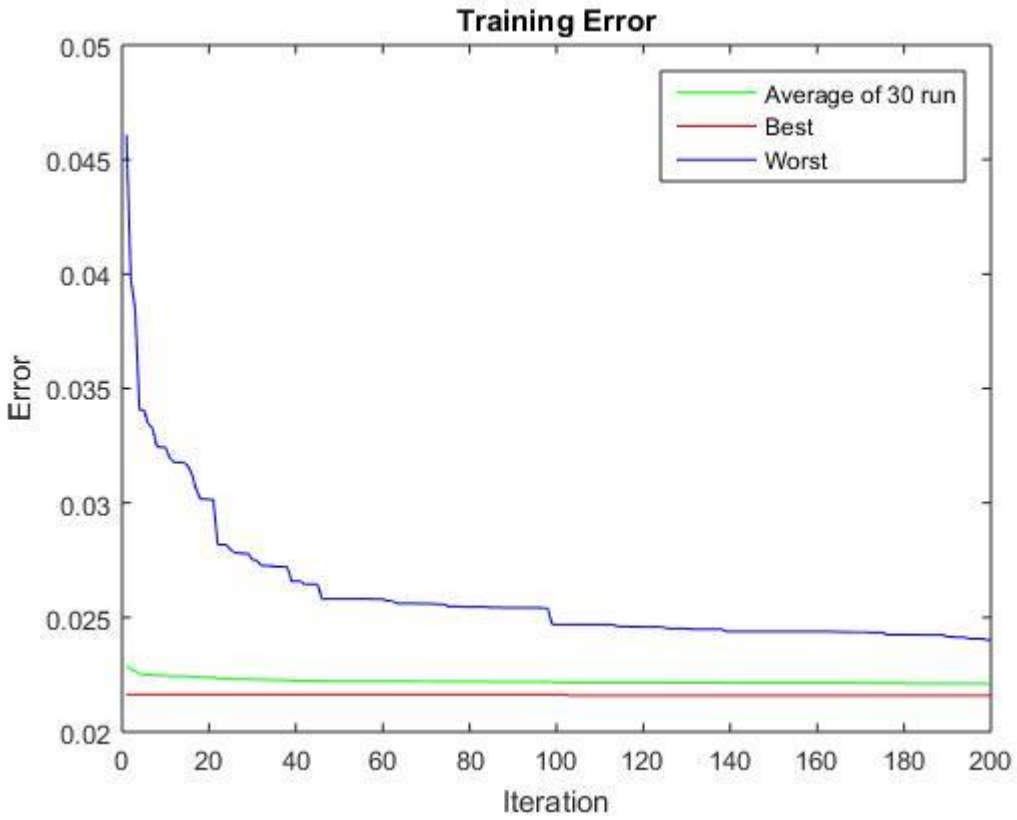

Figure 4.3. Training Error graph performance using MLP-AAA for diabetes dataset.

Table 4.11. Results of MLP-AAA using diabetes dataset attributes 2, 6, 8 for different targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	268	268	268	268
TP	51	67	90	90
FP	39	23	0	0
TN	154	171	180	183
FN	29	12	3	0
Sensitivity	63.75	84.81	96.7742	100
Specificity	79.79	88.14	100	100
Standard Deviation Training Accuracy	0.605668	0.485925	0.246911%	0.096907
Min Training Accuracy	80.16	85.96	92.62	96.82
Max Training Accuracy	84.51	90.08	94.38	97.52
Mean Training Accuracy	84.09	89.78	94.24	97.45
Error is (Fitness/MSE)	0.164793	0.097727	0.060659	0.026367
Testing Accuracy	75.09	87.17	98.90	100

Table 4.12. Results of MLP-AAA using diabetes dataset attributes 1, 2, 6, and 8 for different targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	1,2,6,8	1,2,6,8	1,2,6,8	1,2,6,8
TP	46	76	88	90
FP	44	14	2	0
TN	160	154	180	183
FN	23	29	3	0
Sensitivity	66.66	72.38	96.70	100
Specificity	78.43	91.66	98.90	100
Standard Deviation Training Accuracy	0.814108	0.628031	0.352020	0.142145
Min Training Accuracy	78.65	85.34	91.86	96.44
Max Training Accuracy	84.57	90.14	94.45	97.54
Mean Training Accuracy	84.07	89.78	94.22	97.45
Error is (Fitness/MSE)	0.158757	0.115757	0.063024	0.025401
Testing Accuracy	75.45	84.24	98.16	100

Table 4.13. Results of MLP-AAA using diabetes dataset attributes 1, 2, 6, 7, and 8 for different targets.

Data name	Diabetes	Diabetes	Diabetes	Diabetes
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	1,2,6,8	1,2,6,8	1,2,6,8	1,2,6,8
TP	56	71	85	90
FP	34	19	5	0
TN	146	165	181	183
FN	37	18	2	0
Sensitivity	60.21	79.77	97.70	100
Specificity	81.11	89.67	97.31	100
Standard Deviation Training Accuracy	1.198722	0.818753	0.436980	0.212574
Min Training Accuracy	74.8	84.28	91.77	96.05
Max Training Accuracy	85.32	90.53	94.70	97.65
Mean Training Accuracy	84.56	90.04	94.39	97.51
Error is (Fitness/MSE)	0.172380	0.109163	0.059044	0.025328
Testing Accuracy	73.99	86.44	97.43	100

4.2.2. Results MLP-AAA with Hepatitis Dataset

MLP-AAA was run 30 times by using Hepatitis dataset for 5 neurons in hidden layers, the other important parameters, sheer force, energy loss, adaption and the number of algae colony were selected as, 2, 0.3, 0.2 and 30, respectively.

By using all attribute of Hepatitis dataset for MLP-AAA, the best results were obtained for selected target values (0.2, 0.6) (Table 4.14).

Figure 4.4 shows the average, the best and worst training error values of MLP-AAA for 30 runs with all the attributes of hepatitis using 30 algae colonies and 200 iterations, In the same way, the attribute combinations of hepatitis dataset for MLP-AAA used as MLP-PSO to compare results. The results were given in Tables 4.15-4.18. This algorithm, the best results were also obtained using MLP-AAA with target values (0.2, 0.6) for all attribute combinations of hepatitis dataset.

Table 4.14. Result of MLP-AAA using (All) attribute of hepatitis dataset with different of targets.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	All	All	All	All
TP	39	38	39	40
FP	1	2	1	0
TN	2	6	8	9
FN	7	3	1	0
Sensitivity	84.78	92.68	97.50	100
Specificity	66.66	75	88.88	100
Standard Deviation Training Accuracy	5.578260	3.466637	2.297176	1.761903
Min Training Accuracy	57.35	74.67	81.90	87.21
Max Training Accuracy	93.26	95.68	97.73	97.47
Mean Training Accuracy	89.619128	93.35	96.27	96.21
Error is (Fitness/MSE)	0.190365	0.114210	0.059590	0.032946
Testing Accuracy	83.67	89.79	95.91	100

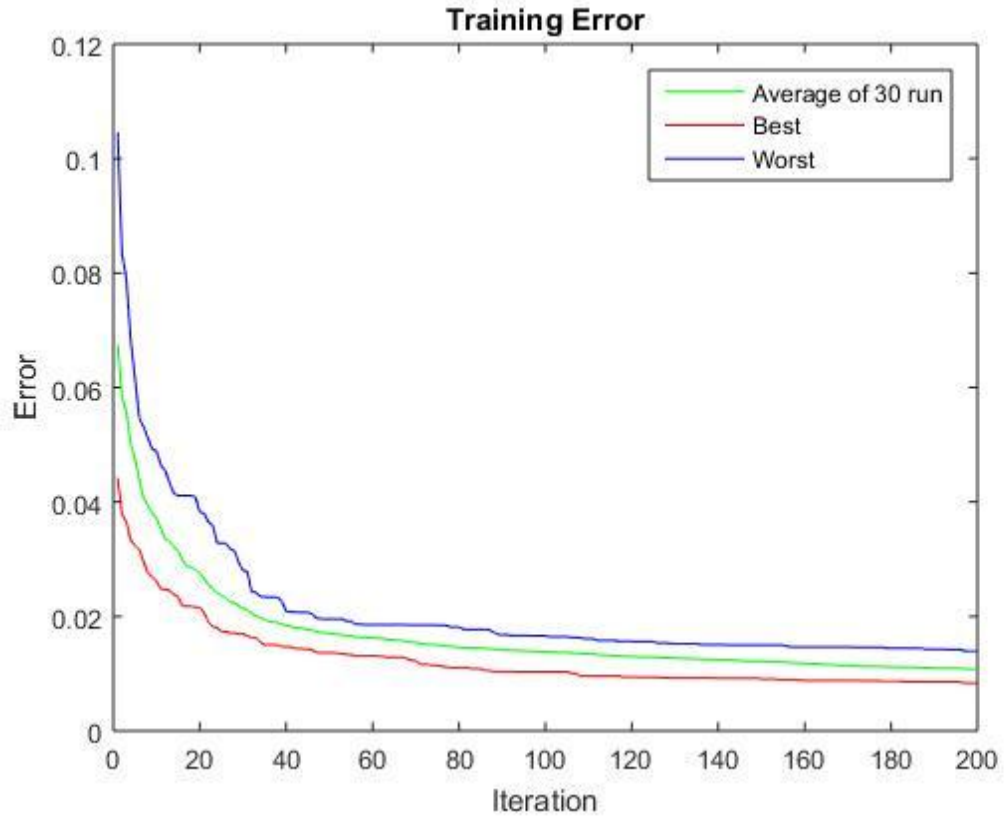

Figure 4.4. Training Error graph performance using MLP-AAA for Hepatitis dataset.

Table 4.15. Result of MLP-AAA using (5, 6, 11, 12, 13, 14, 17, and 19) attribute Hepatitis dataset.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	5, 6, 10, 11, 12, 13, 14, 17, 19	5, 6, 10, 11, 12, 13, 14, 17, 19	5, 6, 10, 11, 12, 13, 14, 17, 19	5, 6, 10, 11, 12, 13, 14, 17, 19
TP	37	39	40	39
FP	3	1	0	1
TN	4	5	5	7
FN	5	4	4	2
Sensitivity	88.09	90.69	90.90	95.12
Specificity	57.14	83.33	100	87.50
Standard Deviation Training Accuracy	3.491780	2.092846%	0.928703	0.535558
Min Training Accuracy	67.69	79.28	90.43	95.34
Max Training Accuracy	92.18	94.98	97.07	98.76
Mean Training Accuracy	90.04	93.63	96.45	98.44
Error is (Fitness/MSE)	0.109282	0.071401	0.040946	0.029508
Testing Accuracy	83.67	89.79	91.83	93.87

Table 4.16. Result of MLP-AAA using (2, 5, 6, 10, 11, 12, 13, 14, 17, 19) Hepatitis dataset.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	(2, 5, 6, 10, 11, 12, 13, 14, 17,)	(2, 5, 6, 10, 11, 12, 13, 14, 17,)	(2, 5, 6, 10, 11, 12, 13, 14, 17,)	(2, 5, 6, 10, 11, 12, 13, 14, 17,)
TP	36	36	40	40
FP	4	4	0	0
TN	4	6	8	8
FN	5	3	1	0
Sensitivity	87.80	92.30	97.56	100
Specificity	50	60	100	100
Standard Deviation Training Accuracy	3.736126	2.402537	1.373377	0.686229
Min Training Accuracy	65.65	77.43	87.26	93.51
Max Training Accuracy	92.41	94.96	97.25	98.79
Mean Training Accuracy	89.92	93.37	96.33	98.38
Error is (Fitness/MSE)	0.262057	0.125317	0.043209	0.014278
Testing Accuracy	81.63	85.71	97.95	100

Table 4.17. Result of MLP-AAA using (6, 11, 12, 13, 14, 17, and 19) attribute Hepatitis dataset.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	(6, 11, 12, 13, 14, 17,19)	(6, 11, 12, 13, 14, 17,19)	(6, 11, 12, 13, 14, 17,19)	(6, 11, 12, 13, 14, 17,19)
TP	38	40	39	39
FP	2	0	1	1
TN	4	5	7	8
FN	5	4	2	1
Sensitivity	88.37	90.90	95.12	97.50
Specificity	66.66	100	87.50	88.88
Standard Deviation Training Accuracy	0.027593	2.058521	1.072034	0.424080
Min Training Accuracy	71.80	81.05	90.55	95.55
Max Training Accuracy	92.71	95.19	97.30	98.81
Mean Training Accuracy	90.78	93.88	96.52	98.52
Error is (Fitness/MSE)	2.759291	0.127194	0.047394	0.05
Testing Accuracy	85.71	91.83	93.87	95.91

Table 4.18. Result of MLP-AAA using (11, 12, 13, 14, 17, and 19) attribute Hepatitis dataset.

Data name	Hepatitis	Hepatitis	Hepatitis	Hepatitis
Selected targets	(0,1)	(0.1,0.9)	(0.2,0.8)	(0.2,0.6)
Selected feature	(11, 12, 13, 14, 17,19)	(11, 12, 13, 14, 17,19)	(11, 12, 13, 14, 17,19)	(11, 12, 13, 14, 17,19)
TP	39	40	39	39
FP	1	0	1	1
TN	3	3	7	8
FN	6	6	2	1
Sensitivity	86.66	86.95	95.12	97.50
Specificity	75	100	87.50	88.88
Standard Deviation Training Accuracy	2.214617	1.680140	0.749745	0.313374
Min Training Accuracy	75.61	80.81	90.66	96.37
Max Training Accuracy	91.11	94.43	96.87	98.62
Mean Training Accuracy	89.74	93.54	96.36	98.40
Error is (Fitness/MSE)	0.092967	0.093754	0.037403	0.033921
Testing Accuracy	85.71	87.75	93.87	95.91

4.3. General Comparison of MLP-PSO and MLP-AAA

The results of diabetes and hepatitis datasets were compared with other researches in Table 4.19 and Table 4.20. It can be clearly seen that the results of this work are much better than those of the other researches. This study used the whole attributes with generally better accuracy than the other researches. And by comparing the results of (0, 1) targets with the results of the targets (0.1, 0.9), (0.2, 0.8) and (0.2, 0.6), it can be seen that the target (0.2, 0.6) made better results than other targets.

The results of diabetes dataset for (0 1) and (0.1,0.9) target values obtained with MLP-AAA are better than MLP-AAA as MLP-AAA is more successful than MLP-PSO for hepatitis dataset for all target values (Table 4.19 and Table 4.20). Furthermore, it can be said that MLP-PSO slightly better than MLP-AAA with very close results making MLP-AAA very competitive to MLP-PSO.

Table 4.19. Comparison results using Diabetes dataset with similar studies.

Methodology adopted	Selected features	Sensitivity	Specificity	Accuracy
PCA-ANFIS (10x FC) (Polat and Gunes, 2007)	All	85.71	92	89.47
LS-ELM (10x FC) (Polat et al., 2008)	All	73.91	80	78.21
GDA-LS-ELM (10x FC) (Polat et al., 2008)	All	83.33	82.05	79.16
MLNN with LM (10x FC) (H. Temurtas,2009)	1, 2, 6, 8	70	70.31	79.62
PNN (10x FC) (H. Temurtas,2009)	2, 6, 8	71	70.5	78.05
LDA-MWELM (Calisir and Esin, 2011)	1, 2, 6, 7, 8	83.33	93.75	89.74
Mean selection method (Jaganathan and Kuppuchamy, 2013)	2, 6, 8	71	78	76.04
Half selection method (Jaganathan and Kuppuchamy, 2013)	1, 2, 6, 8	69	79	75.91
Neural network for threshold selection (Jaganathan and Kuppuchamy, 2013)	2, 6, 8	71	78	76.04
PSO + ELM (Subbulakshmi and Deepa, 2015).	1, 2, 6, 8	85.26	94.10	91.27
SRLPSO + ELM (Subbulakshmi and Deepa, 2015).	2, 6, 8	91.47	96.26	93.09
Proposed MLP-PSO(0, 1)	All	67.53	80.61	76.92
Proposed MLP-PSO (0.1, 0.9)	All	80.61	89.52	88.27
Proposed MLP-PSO (0.2, 0.8)	All	98.35	95.60	97.43
Proposed MLP-PSO (0.2, 0.6)	All	100	100	100
Proposed MLP-AAA(0, 1)	All	64.55	79.89	75.45
Proposed MLP-AAA (0.1, 0.9)	All	79.74	86.08	84.24
Proposed MLP-AAA (0.2, 0.8)	All	96.62	97.82	97.43
Proposed MLP-AAA (0.2, 0.6)	All	100	100	100

Table 4.20. Comparison of Classification results using Hepatitis dataset with similar studies.

Methodology adopted	Selected features	Sensitivity	Specificity	Accuracy
Mean selection method(Jaganathan and Kuppuchamy, 2013)	5, 6, 11, 12, 13, 14, 17, 19	87	60	82.58
Half selection method(Jaganathan ve Kuppuchamy, 2013)	2, 5, 6, 10, 11, 12, 13, 14, 17, 19	90	66	85.16
Neural network for threshold selection(Jaganathan and Kuppuchamy, 2013)	2, 5, 6, 10, 11, 12, 13, 14, 17, 19	90	66	85.16
PSO + ELM(Subbulakshmi and Deepa, 2015)	6, 11, 12, 13, 14, 17, 19	93.65	95.71	97.43
SRLPSO + ELM (Subbulakshmi and Deepa, 2015)	11, 12, 13, 14, 17, 19	94.22	96.04	98.71
Proposed MLP-PSO(0.1)	All	62.66	78.28	73.99
Proposed MLP-PSO(0.1,0.9)	All	85.18	89.06	87.91
Proposed MLP-PSO(0.2,0.8)	All	96.62	97.82	94.16
Proposed MLP-PSO(0.2,0.6)	All	100	100	100
Proposed MLP-AAA(0.1)	All	84.78	66.66	83.67
Proposed MLP-AAA(0.1,0.9)	All	92.68	75	89.79
Proposed MLP-AAA(0.2,0.8)	All	97.50	88.88	95.91
Proposed MLP-AAA(0.2,0.6)	All	100	100	100

Table 4.21. The best results for both Diabetes and Hepatitis

Dataset	**Method**	**TP**	**FP**	**TN**	**FN**	**Sensitivity**	**Specificity**	**Mean Training Accuracy**	**Testing Accuracy**
Hepatitis	MLP-PSO	40	0	9	0	100	100	98.55	100
	MLP-AAA	40	0	9	0	100	100	96.21	100
Diabetes	MLP-PSO	90	0	183	0	100	100	97.37	100
	MLP-AAA	90	0	183	0	100	100	97.77	100

5. RESULTS AND RECOMMENDATIONS.

5.1. Results.

In this study, MLP-AAA was implemented for the classification problems. Furthermore, to see the performance of MLP-AAA MLP-PSO was implemented in the same conditions. The experiments with both algorithms were performed using diabetes and hepatitis datasets and also for different class target values. Thus the effects of target values were investigated. Also, the experiments were carried out with different attributes combinations studied in the literature. Moreover, the results of them were compared with literature. The best success rate was obtained as 100% for both MLP-AAA and MLP-PSO for all attributes of both datasets. In addition, the best performances of both algorithms were achieved with the target values (0.2 0.6).

The parameters that are used for both diabetes and hepatitis classification in MLP-PSO and MLP-AAA are shown in table 5.1 and Table 5.2, which had an accuracy of 100% for both diabetes and hepatitis datasets.

Table 5.1. MLP-PSO parameters.

Category	Parameter	Value
MLP	MLP type	Feed forward
	Number of hidden layer	1
	Maximum iteration	200
	Lower band(LB) and upper band(UB)	(-2,2)
	Nods number in inputs, hidden and output layers	8-5-1
	Maximum run	30
PSO	Population size	30
	Learning factor (c1,c2)	c1+c2=4
	Objective function	MSE

Table 5.2. MLP-AAA parameters.

Category	Parameter	Value
MLP	MLP type	Feed forward
	Number of hidden layer	1
	Maximum iteration	200
	Lower band(LB) and upper band(UB)	(-2,2)
	Nods number in inputs ,hidden and output layers	19-5-1
	Maximum run	30
AAA	Population size	30
	Sheer Force	2
	Energy Loss	(0.3)
	Adaptation	(0.2)
	Objective function	MSE

5.2. Suggestions

This work is mainly based on the PSO and AAA optimization algorithms to train ANN to solve classification and prediction problems. In future studies, the PSO and AAA algorithms can be modified to increase the performance and speed of MLP. The design of the classification process as an automated system has opened the way to great convenience in use.

Both, MLP models (MLP-PSO and MLP-AAA) used for the classification of hepatitis and diabetes diseases in this study can also be adapted for similar problems.

Generally, much of the networks being designed presently are statistically quite accurate (up to 100 % accuracy).

References

Abdul-Kader, H., 2009, Neural networks training based on differential evolution algorithm compared with other architectures for weather forecasting34, *International Journal of Computer Science and Network Security*, 9 (3), 92-99.

Alba, E. ve Martí, R., 2006, Metaheuristic procedures for training neural networks, Springer Science & Business Media, p.

Beni, G. ve Wang, J., 1993, Swarm intelligence in cellular robotic systems, In: Robots and Biological Systems: Towards a New Bionics?, Eds: Springer, p. 703-712.

Çalişir, D. ve Doğantekin, E., 2011, An automatic diabetes diagnosis system based on LDA-Wavelet Support Vector Machine Classifier, *Expert systems with applications*, 38 (7), 8311-8315.

Carvalho, M. ve Ludermir, T. B., 2007, Particle swarm optimization of neural network architectures andweights, *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on*, 336-339.

Chen, M., 2008, Second generation particle swarm optimization, *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, 90-96.

Cheung, N., 2001, Machine learning techniques for medical analysis, *School of Information Technology and Electrical Engineering*.

Conforth, M. ve Meng, Y., 2008, Toward Evolving Neural Networks using Bio-Inspired Algorithms, *IC-AI*, 413-419.

Da, Y. ve Xiurun, G., 2005, An improved PSO-based ANN with simulated annealing technique, *Neurocomputing*, 63, 527-533.

Das, R., Turkoglu, I. ve Sengur, A., 2009, Effective diagnosis of heart disease through neural networks ensembles, *Expert systems with applications*, 36 (4), 7675-7680.

Eberhart, R. C. ve Kennedy, J., 1995, Particle swarm optimization, *Proceedings of the IEEE international conference on neural networks*, 1942-1948.

Garro, B. A., Sossa, H. ve Vazquez, R. A., 2009, Design of artificial neural networks using a modified particle swarm optimization algorithm, *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, 938-945.

Garro, B. A., Sossa, H. ve Vázquez, R. A., 2010, Design of artificial neural networks using differential evolution algorithm, *International Conference on Neural Information Processing*, 201-208.

Garro, B. A., Sossa, H. ve Vázquez, R. A., 2011a, Back-propagation vs particle swarm optimization algorithm: which algorithm is better to adjust the synaptic weights of a feed-forward ANN?, *International Journal of Artificial Intelligence*, 7 (11), 208-218.

Garro, B. A., Sossa, H. ve Vázquez, R. A., 2011b, Evolving neural networks: A comparison between differential evolution and particle swarm optimization, *International Conference in Swarm Intelligence*, 447-454.

Garro, B. A., Sossa, H. ve Vázquez, R. A., 2011c, Artificial neural network synthesis by means of artificial bee colony (abc) algorithm, *Evolutionary Computation (CEC), 2011 IEEE Congress on*, 331-338.

Gu, T.-Y., Ju, S.-G. ve Han, F., 2009, A PSO algorithm with the improved diversity for feedforward neural networks, *Intelligent Information Technology and Security Informatics, 2009. IITSI'09. Second International Symposium on*, 123-127.

Güler, i. Ve übeyli, e. D., 2006, çok katmanli perseptron sinir ağlari ile diyabet hastaliğinin teşhisi, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 21 (2).

Higashi, N. ve Iba, H., 2003, Particle swarm optimization with Gaussian mutation, *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, 72-79.

Jaganathan, P. ve Kuppuchamy, R., 2013, A threshold fuzzy entropy based feature selection for medical database classification, *Computers in Biology and Medicine*, 43 (12), 2222-2229.

Kahramanli, H. ve Allahverdi, N., 2008, Design of a hybrid system for the diabetes and heart diseases, *Expert systems with applications*, 35 (1-2), 82-89.

Kohavi, R., 1995, A study of cross-validation and bootstrap for accuracy estimation and model selection, *Ijcai*, 1137-1145.

Kuok, K. K., Harun, S. ve Shamsuddin, S., 2010, Particle swarm optimization feedforward neural network for modeling runoff, *International Journal of Environmental Science & Technology*, 7 (1), 67-78.

Løvbjerg, M., Rasmussen, T. K. ve Krink, T., 2001, Hybrid particle swarm optimiser with breeding and subpopulations, *Proceedings of the 3rd ANNual Conference on Genetic and Evolutionary Computation*, 469-476.

Mohais, A. S., Mohais, R., Ward, C. ve Posthoff, C., 2007, Earthquake classifying neural networks trained with random dynamic neighborhood PSOs, *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 110-117.

Ninomiya, H. ve Zhang, Q.-J., 2008, Particle with Ability of Local search Swarm Optimization: PALSO for training of feedforward neural networks, *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, 3009-3014.

Polat, K., Sahan, S., Kodaz, H. ve Günes, S., 2005, A new classification method to diagnosis heart disease: supervised artificial immune system (AIRS), *proceedings of the turkish symposium on artificial intelligence and neural networks (TAINN)*.

Polat, K. ve Güneş, S., 2007, An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease, *Digital Signal Processing*, 17 (4), 702-710.

Polat, K., Güneş, S. ve Arslan, A., 2008, A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine, *Expert systems with applications*, 34 (1), 482-487.

Reibnegger, G., Weiss, G., Werner-Felmayer, G., Judmaier, G. ve Wachter, H., 1991, Neural networks as a tool for utilizing laboratory information: comparison with linear discriminant analysis and with classification and regression trees, *Proceedings of the National Academy of Sciences*, 88 (24), 11426-11430.

Rivero, D. ve Periscal, D., 2009, Evolving graphs for ann development and simplification, In: Encyclopedia of Artificial Intelligence, Eds: IGI Global, p. 618-624.

Shi, Y. ve Eberhart, R., 1998, A modified particle swarm optimizer, *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, 69-73.

Subbulakshmi, C. ve Deepa, S., 2015, Medical dataset classification: a machine learning paradigm integrating particle swarm optimization with extreme learning machine classifier, *The Scientific World Journal*, 2015.

Temurtas, H., Yumusak, N. ve Temurtas, F., 2009, A comparative study on diabetes disease diagnosis using neural networks, *Expert systems with applications*, 36 (4), 8610-8615.

Tewolde, G. S. ve Hanna, D. M., 2007, Particle swarm optimization for classification of breast cancer data using single and multisurface methods of data separation, *Electro/Information Technology, 2007 IEEE International Conference on*, 443-446.

Uymaz, S. A., Tezel, G. ve Yel, E., 2015, Artificial algae algorithm (AAA) for nonlinear global optimization, *Applied Soft Computing*, 31, 153-171.

Wang, Z., Sun, X. ve Zhang, D., 2007, A PSO-based classification rule mining algorithm, *International Conference on Intelligent Computing*, 377-384.

Yalcin, N., Tezel, G. ve Karakuzu, C., 2015, Epilepsy diagnosis using artificial neural network learned by PSO, *Turkish Journal of Electrical Engineering & Computer Sciences*, 23 (2), 421-432.

Yao, X. ve Liu, Y., 1997, A new evolutionary system for evolving artificial neural networks, *IEEE transactions on neural networks*, 8 (3), 694-713.

Yao, X., 1999, Evolving artificial neural networks, *Proceedings of the IEEE*, 87 (9), 1423-1447.

Yu, J., Xi, L. ve Wang, S., 2007, An improved particle swarm optimization for evolving feedforward artificial neural networks, *Neural Processing Letters*, 26 (3), 217-231.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Saeed Shakir Mahmood
Uyruğu : IRAK
Doğum Yeri ve Tarihi : 13.10.1981
Telefon : 05070404799
Faks :
e-mail : alkasmk@gmail.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Al khaldiya lisesi, al khaldiya,Al_Anbar	**2000**
Üniversite	:	AL_Maarf Üniversitesi Bilgisayar Mühendisliği Fakültesi.	**2006**
Yüksek Lisans	:		
Doktora	:		

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2008	I.Q. Sağlık Bakanlığı_ Al-Anbar il sağlık müdürlüğü	Bilgisayar Mühendisliği

UZMANLIK ALANI: Bilgisayar Mühendisliği

YABANCI DİLLER: Ingilizce, Türkçe ve Arapça

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR : Saeed, M., and G.Tezel, 2017, *Solve complex problems using artificial Neural network learned by PSO, Internatıonal Conference of Engineering Technologies (ICENTE17) 978-605-67535-4-1*.