Arab American University- Jenin

Faculty of Graduate Studies

Text Mining Using Radial Basis Function Neural Networks and Optimization Algorithms

By

Monir Sayel Hamed Foqaha

Supervisor: Dr. Mohammed Awad

This thesis was submitted

in partial fulfillment of the requirements for the Master's degree in

Computer Science

December, 2016

Text Mining Using Radial Basis Function Neural Networks and Optimization Algorithms

By

Monir Sayel Hamed Foqaha

This thesis was defended successfully on **17/12/2016** and approved by:

Committee Members	Signature
1. Supervisor: **Dr. Mohmmed Awad**	.....….
2. Internal Examiner: **Dr. Labib Arafeh**	.....
3. External Examiner: **Dr. Mohammed Aldasht**	.....

Dedication

I would like to thank all my friends who are always willing to provide any supporting.

I would like to dedicate this work to my parents; my parents support me on every step I made it.

I dedicate this work to my dear wife. Also, I thank each of my brothers and sisters.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Mohammed Awad, who has always been providing support, encouragement, and guidance throughout the completion of this thesis.

Also, I extend my sincere thanks and appreciation to the staff of the faculty of engineering and information technology at AAUJ- master program.

Finally, I would like to thank my family: my parents, wife, brothers, and sisters for supporting and helping me throughout this work.

Abstract

Text mining is a process for extracting information from an unstructured text. Text mining can work with unstructured or semi-structured data sets such as E-mails. Spam emails are unsolicited emails. It consumes storage of mail servers, waste of time and consumes network bandwidth. One of the most powerful tools that are used for email classification is Artificial Neural Networks (ANNs); it has the capability of dealing with huge amount of data with high dimensionality in better accuracy.

In this thesis, we proposed a hybrid approach which combines Radial basis function Neural Network (RBFNN) and Particle Swam Optimization (PSO) algorithm. This approach is applied on two applications; the first is HRBFN-PSO that is applied on function approximation and time series prediction. The other is HC-RBFN-PSO which is used for classification in order to classify the emails spams. In the both proposed hybrid applications (HRBFN-PSO and HC-RBFN-PSO), the parameters of RBFNN are optimize as follow: Center is optimized using Particle Swam Optimization algorithm (PSO), Radii is optimized using K-Nearest Neighbors algorithm (KNN), and weights is optimized using Singular Value Decomposition algorithm (SVD). These two traditional algorithms (KNN and SVD) are integrated within each iterative process of Particle Swam Optimization, the weights and Radii are updated depending on the fitness (error) function.

In the first application of the function approximation and time series prediction, the method HRBFN-PSO conducts experiments on nonlinear benchmark mathematical functions and Mackey Glass time series. The results obtained on the training data clarify that the proposed approach improved the approximation accuracy more than other traditional approaches. Also, this result shows that, HRBFN-PSO reduces the Root

Mean Square Error (RMSE) and Sum Square Error (SSE) dramatically compared with other approaches. Through our experiments for function approximation, we got the best RMSE value which is 0.000034.

However, in the other application, the proposed method HC-RBFN-PSO conducted experiments on benchmark spam dataset namely SPAMBASE downloaded from UCI Machine Learning Repository, to classify emails into two classes namely spam and non-spam. The experimental results of this application show that our approach is more accurate compared with other approaches that use the same dataset. Through our experiments for email classification, we got the best classification accuracy value is 93.7% that is best compared with other approaches, they had obtained the best accuracy value is 93.28.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ANN	Artificial Neural Network
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
PSO	Particle Swarm Optimization
KDT	Knowledge Discovery from Text
SVM	Support Vector Machine
MLPNN	Multilayer Perceptron Neural Network
MLP	Multi-Layer Perceptron
ELM	Extreme Learning Machine
RMSE	Root Mean Square Error
MSE	Mean Square Error
SSE	Sum Square Error
N/RMSE	Normalized/Root Mean Squared Error
NNs	Neural Networks
FFNN	Feed Forward Neural Network
LMS	Least Mean Square
EA	Evolutionary Algorithm
SI	Swarm Intelligence
GA	Genetic Algorithm
BPSO	binary particle swarm optimization
KNN	K-Nearest Neighbors
SVD	Singular Value Decomposition
GUI	Graphical User Interface

CSV	Comma Separated Values
TP	True Positives
FP	False Positives
FN	False Negative
TN	True Negatives

CHAPTER 1

INTRODUCTION

1. Introduction

Artificial Neural Network (ANN) is a set of mathematical models that inspired by biological neural networks. It is used in many fields such as classification, clustering, function approximation, pattern recognition, signal processing, and feature extraction. In this thesis, we employed two important ANN fields which are function approximation and classification. Function approximation is used to describe the behavior of complicated mathematics functions. The main goal of function approximation is to minimize the error function value as much as possible between real and target output. Text mining is a process of deriving high-quality information from text such as natural language texts; text mining includes the discovery of new and previously unknown information. The basic tasks of text mining are text clustering and text classification. In this thesis, we focused on classification data that is the process of organizing data into a set of categories based on the most effective and efficient use. In our experiments, we used one example text mining classifier which is email spam classification. Email spam is the major problem that faces emails; that is because the email spam affects email servers storage space, networks bandwidth, processing power and user time negatively. We proposed one RBFNN classifier which is email classification. Email classification categorizes emails into two categories. These categories are spam and non-spam.

In this thesis, we used one type of ANN that is Radial Basis Function Neural Network (RBFNN). It is a feedforward network and type of supervised learning network. Hence, we applied RBFNN for function approximation and classification. Thereupon, to reach the best accuracy for the results of function approximation, we use fitness function measuring, which is the error function between the target and real output of the

RBFNN. In contrast, the process of classification data to reach the best accuracy for results, we use classification accuracy measuring. As a result, each output node computes a sort of score for the associated category; then, determining the suitable class is done by assigning the input to the category with the highest score.

In this thesis, we used Particle Swarm Optimization (PSO) hybrid with RBFNN to optimize centers of RBFNN. PSO is applied on many applications such as Function Approximation, Artificial Neural Network training, Fuzzy system, Clustering, and Classification. In Function Approximation, PSO used to minimizing the error function between the target and real output. In Classification, it is possible to use PSO for feature selection to select small number of features and achieve better classification performance instead of using all features.

For this hybrid approach, we propose two applications, the first application is HRBFN-PSO used for function approximation problems and time series prediction, while the second application is HC-RBFN-PSO used for classification data.

1.1 Thesis Objectives

The main objective of this thesis comes to introduce a hybrid approach combined Radial Basis Function Neural Network (RBFNN) with Particle Swarm Optimization (PSO) to be able to overcome the problem of tuning parameters (centers) of RBFNN. For this hybrid approach, we propose two applications HRBFN-PSO and HC-RBFN-PSO to train RBFNN using PSO algorithm. The aim for HRBFN-PSO approach is to build a system used for function approximation and time series prediction for any problem and nonlinear function. For this application, we conduct experiments on three nonlinear benchmark function and Mackey-Glass time series. On the other hand, the aim of HC-RBFN-PSO application is to develop a system that used of problems classification.

Here, at the beginning, we conduct experiments on two datasets namely Wisconsin Diagnostic Breast Cancer and Ionosphere, then it conducted experiments for this application on spam filtering technique to classify email message as spam and non-spam.

1.2 Thesis Contributions

This thesis contributes mainly in building effective hybrid system (RBFNN combining with PSO) that is used in function approximation problems, time series prediction and classification; specially E-mail spam classification. This hybrid system overcomes the problem of tuning parameters of RBFNN and optimizes the centers of hidden neurons in RBFNN, radii and weights optimized using traditional mathematical algorithms like KNN and SVD. We aim to create a model that has the ability to approximate functions and classify categories with better approximation error and better classification accuracy.

1.3 Thesis Overview

The overview of this thesis is organized as follows: in chapter 2, we introduced the thesis background topics. At the beginning, we introduced the introduction about function approximation, time series prediction, text mining, email spam, and email spam classification. Then, we presented an introduction about the Artificial Neural Network (ANN). After that, we explained the components and architecture of ANN and enumerate some important fields using the Neural Networks and clarify calculations taking place in ANN. Then we show the types of Neural Networks; as well we described the Radial Basis Function Neural Network (RBFNN) that is used in conducting our experiments. In the next section, we demonstrate an introduction about Bio-inspired optimization algorithms. Then, we explained two major types of Bio-

inspired optimization algorithms. These types are Evolutionary Algorithms (EA) and Swarm Intelligence (SI); then we clarify Particle Swarm Optimization algorithm (PSO) that is used in our proposed approaches. At the last section, we demonstrate the importance of combined ANN and PSO.

Chapter 3 shows some literature review of proposed approaches several studies that combine RBFNN and PSO within all aspects such as RBFNN parameter and architecture network. After that, we show numerous related works about spam filtering techniques using machine learning methods.

In chapter 4, we have described the proposed method. Firstly, we talked in general about combined RBFNN and PSO. After that, the first HRBFN-PSO proposed an approach that is used for function approximation and time series prediction which also shows the pseudo code for HRBFN-PSO approach. Then we introduced the second HC-RBFN-PSO proposed approach that is used for classification. However, the aforementioned approach is used for spam filtering technique and presents the pseudo code for HC-RBFN-PSO approach. We have illustrated the K-Nearest Neighbors (KNN) algorithm that is used to optimize the radii parameter of RBFNN. At the end of this chapter, Singular Value Decomposition (SVD) algorithm is explained which is used to find the optimal values of connection weights parameter.

Experiments and Results are shown in chapter 5, at the beginning, it has been clarifying computer specification which conducts our experiments and environment that are used in this experiments. In the next section, we have shown the benchmark mathematical function that used in our experiments. Then, we presented the results the whole experiments and compare it with previous works that use the same benchmark mathematical functions. Also, to prove the strength and efficiency of HRBFN-PSO

approach; we conducted multiple experiments on Mackey-Glass time series which is a complex mathematical function. The next section shows the conducted experiments for classification on two datasets for second proposed approach which aims to test the strength and accuracy of HC-RBFN-PSO. Then, we show spam classification experiment for this approach. Here we have talked in details about spam dataset used in our experiments, and then we talked about the most important point before conducting the experiment that is preprocessing data. After that, we show the results for the experiments using multiple methods and we compared it with previous works used the same dataset. In the last section, we have discussed the experiments and results in details.

Finally, chapter 6 presents the conclusion and the future works.

CHAPTER 2

Background

2. Background

This chapter provides a brief introduction to the main topics of this thesis, we present a general introduction to functions approximation problem as one of the application using on this work, and also, we introduce an overview of time series prediction to be used to validate the utility of the proposed method. As an application of the proposed approach on classification problem, we provide an introduction to text mining in general, and email spam classification as the application performed by the proposed method. In this chapter, we have introduced full explanation about algorithms for proposed hybrid approach that is Radial Basis Function Neural Networks (RBFNN) and Particle Swarm Optimization (PSO).

2.1 Function Approximation

In the field of mathematical and application sciences, it is important to determine unknown functions from a set of input/output data. It is common finding data with discrete values, this produce pattern of the relationship between input and output which called curve fitting [1]. The functions approximation generally obtained by linear combination of elementary functions, which take the form as explained in the following expression:

$$c_n f_n(x) + c_{n-1} f_{n-1}(x) + \ldots + c_0 f_0(x) : \quad 0 \le i \le n \tag{2.1}$$

Where *c* is the constant values that we want to find and *f(x)* is specific elementary functions [2].

Function Approximation describes the behavior of complicated mathematics functions by collecting simpler functions and reducing the computing time to calculate the values of function [3]. The main goal of function approximation is improving the estimation accuracy which minimizes error function value as much as possible. Function

approximation is used in the most real world applications [4], such as pattern recognition, prediction, classification and data mining [5]. To achieve good approximations of complex non-linear functions, the cluster of the input data must be grouped small patches [6]. Artificial Neural Network (ANN) is one of the function approximation tools; this is due to its ability to find a pattern in input-output data without needing predefined models. In many preceding works, researchers focused basically on the most applicable way in function approximation techniques, which is a supervised learning problem, which may be formulated as a regression task [7]. An efficient type of neural networks for the problem of function approximation called Radial Basis Function Neural Network (RBFNN) which is universal approximations and it achieves faster convergence. The process of using RBFNN to approximate unknown functions from a set of I/O data is illustrated in figure 2.1.

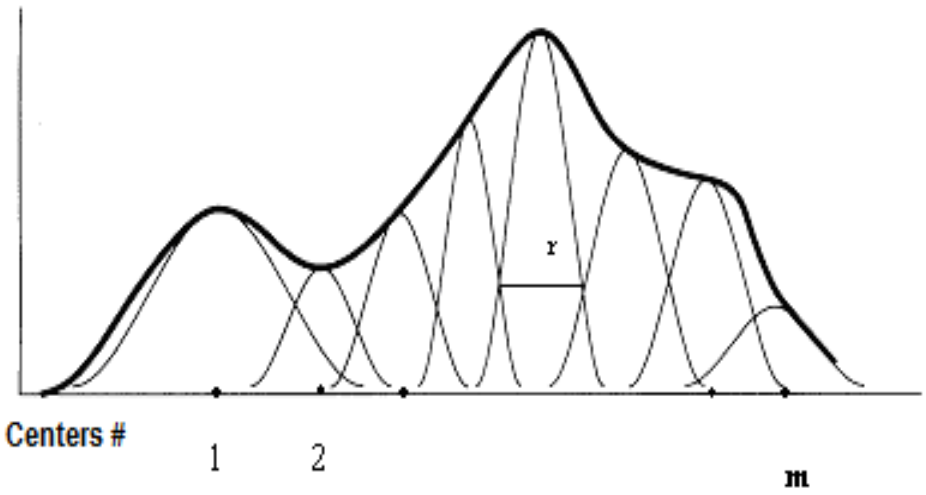

Figure 2.1: Function Approximation using RBFNN [8]

2.2 Time Series Prediction

Time series prediction, which is also known as a signal prediction, it takes a set of current data that is used to predict future data [9]. The main goal of time series is to build a model to deduce future unknown data from current data with minimizing the

error function between input and output. Establishing a mapping between inputs and outputs is the first step for time series, the mapping is linear and nonlinear. Time series prediction has a widespread in many applications such as technology, finance [10], weather forecast [11], sunspot prediction [12], signal processing [13], medicine and economics.

The researchers have proposed many prediction methods to resolve time series prediction problems. One of the most important methods is the Artificial Neural Network (ANN) which is used in both linear and nonlinear time series prediction. Research in [14] presented a review for ANN time series.

ANN uses a set of N data point as inputs and one or multiple outputs as the target value of the ANN. The main architecture of time series prediction in ANN is illustrated in figure 2.2. RBFNN are universal approximations, which prove its efficient ability to predict time series.

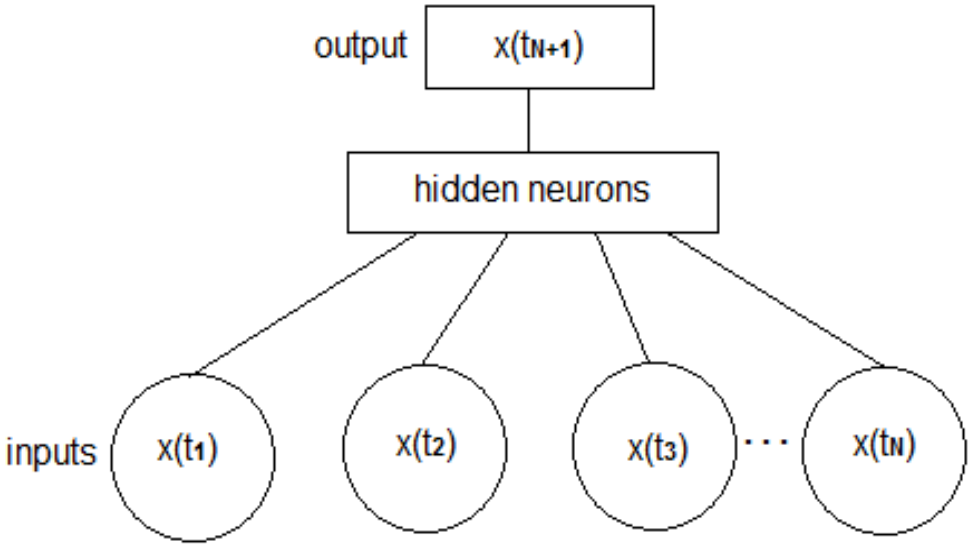

Figure 2.2: Main architecture of time series prediction in ANN

2.3 Text Mining

Nowadays, there is a huge volume of text data in the world which accumulates continuously, so we need to find methods to discover knowledge from the text. Text mining is important for planning and decision-making which known as text data mining [15] or knowledge discovery from the text (KDT) [16]. Text mining is a process of deriving height-quality information from text such as natural language texts; text mining includes the discovery of new and previously unknown information. Generating new information from a large collection of resources is the main purpose of text mining. Text mining is similar to data mining, but they differ in design to handle structured data from databases. On the other hand, text mining can be fit with unstructured or semi-structured datasets such as e-mails and HTML files. Text mining can be considered as an extension of data mining or knowledge discovery from (structured) databases [17, 18]. It is also based on the application of mathematical methods, analysis and specifically using artificial neural networks, which are very useful to perform intelligent analysis of large volumes of digital information. The basic tasks of text mining are text clustering and text classification, but it covers many fields such as text analysis, database technology, information extraction, information retrieval, machine learning, visualization, and data mining. In this thesis, we proposed one text mining classifier that uses the radial basis function neural network (RBFNN). This classifier covers text classification that includes email classification. Email classification classifies emails message into two categories, which are spam and non-spam.

2.4 Email Spam

In the twenty-first century, E-mail has become one of the most important methods for communication among people. This occurred due to its free availability, fast and free or lower cost. The major problem has become in email messages is the unwanted message

(spam). The person that sends the spam messages is called spammer who collects email addresses from websites, chat rooms, and viruses.

The spam traffic volume is so large, which negatively affects email servers storage space, networks bandwidth, processing power and user time [19]. According to the studies conducted between 2010 -2014, 18% of the traffic is spam [20]. Another study according to [21], there are 13 billion of unwanted commercial email nearly 50% of all email sent. On the other hand, Kaspersky security bulletin in 2013 reported that the counted spam during 2013 approximately equal 70% of the total email traffic [22].

2.5 Email Spam Classification

Classification is an example of supervised learning, where each data points consisting pair of an input and a target output value. There are many classification techniques used to classify data into categories, including probabilistic, decision tree, artificial immune system [23], Support Vector Machine (SVM) [24], Artificial Neural Networks (ANN) [25], and case-based technique [26]. Typically the classification process divided two phases: training and testing. In the first training phase, the learning machine is introduced with class label samples, and then learning machine generates a classifier for prediction of the class label of the input samples. But in the second testing phase, we test the generalization ability of the model.

This thesis proposes using one type of ANN for spam classification which is RBFNN. Most important sides that must be taken into account when the ANN is constructed are the network size and classification accuracy. In general, to use these classification techniques for spam filtering, using content-based filtering approach that identifies attributes (usually keywords often used in spam emails). The Frequency occurrence of these attributes within email determines the probabilities for each attribute within the

email, and then it is compared to a threshold value. Email messages that exceed the threshold value are classify as spam otherwise classify as non-spam. ANN consists of simple processing unit called neurons and processes information to do computation operations [27, 28]. Many researches used a neural network to classify spam using content-based filtering, these methods determines attributes to calculate the frequency of keywords or patterns in the email messages. Neural Networks algorithms that are used in email filtering achieve reasonable classification performance. The most famous algorithms are Multilayer Perceptron Neural Network (MLPNN) and Radial Basis Function Neural Networks (RBFNNs). Most researchers used MLPNN as a classifier for spam filtering but very too few of them used RBFNNs as a classifier. In this thesis, we used the hybrid RBFNN and PSO to filters email spams.

2.6 Artificial Neural Networks

Neural Networks (NNs) are mathematical models inspired by biological neural networks. The main part of the human brain is a biological neuron which is a very complicated system with the ability to handle and process the very large amount of information simultaneously. Biological neural networks are able to process multiple inputs much faster than any modern computer system. As a result, many tasks performed by human brain are more efficiently than computers [29].

The first model of an artificial neuron is introduced by Warren McCulloch and Walter Pitts in 1943 [30] and Donald Hebb [31] proposed a Hebbian learning that is the first type of unsupervised learning. There is no single definition for an artificial neural network. Zurada [32] defined a neural network as follows: "*A neural network is simply a class of mathematical algorithms since a net-work can be regarded essentially as a*

graphic notation for a large class of algorithms. Such algorithms produce solutions to a number of specific problems".

The main basic component for all Artificial Neural Networks (ANNs) is neuron. Each ANN is consisting of one or multiple neurons. Each neuron receives data from the input signal or from other neurons in the neural network and computes an output based on these data. Here, the connections between neurons or between neurons and inputs are called weights (W).

Artificial Neural Networks (ANNs) are used and appropriate in many fields [33], such as clustering [34], classification [35], function approximation [36], control [37], and pattern recognition [38].

The calculations that take place in ANN to get the outputs is performed in two stages: in the first stage, the neurons receive the input **n** signals (X_1, X_2, . . ., X_n). Then each input signal is associated with weight [39] values (w_1, w_2, …, w_n), which is distributed normally between 0 and 1. After that the sum of product transfer function (S) is calculated as in equation 2.2.

$$S = \sum_{i=1}^{n} X_i W_i \qquad (2.2)$$

Then it applies activation function F to get the outputs (Y) as in equation 2.3:

$$Y = F(S) \qquad (2.3)$$

The architecture of a neural network shows how their neurons (nodes) are related to each other [40]. ANN consists of a number of layers, the number of nodes per layer and the weight connections between all the nodes. Figure 2.3 illustrates the general architecture of ANN. Each layer can be one of the three different kinds that are described as follows:

- Input layer: Contains source nodes that receive data or signals from the external world and passes it to the next layer [41]. The nodes in this layer are called Input nodes.
- Hidden layer: Any ANN can contain one hidden layer or more. It contains a number of neurons (nodes). Hidden layers are located between input and output layers. The nodes in this layer are called Hidden nodes.
- Output layer: It is the final layer of the ANN. Output layer contains one or multiple output nodes, also the output layer provides the response to the outside world in ANN. The nodes in this layer are called Output nodes.

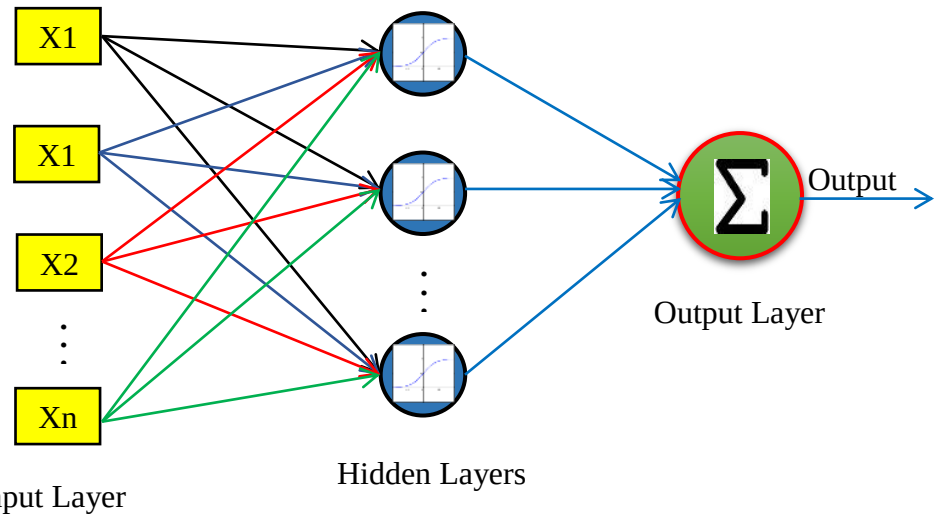

Figure 2.3: Architecture of ANN

The inputs for any neural network must be numerical values. The principle of applying NNs for approximating unknown functions depends on the process of learning by applying examples of these functions. The weights in NNs are updated to produce nearly the same output as in the examples (objective output). The goal behind adjusted the weights are when the ANN is offered a new set of input data; it will give a correct

output and reduce the error between the objective output (target) and the actual output to improve the performance of the ANNs [42].

The activation function is used to transforms a set of inputs to the output. The purpose of an activation function is to ensure that the representation of the data in input space that is mapped to the output for different space. Activation functions in hidden nodes for hidden layer are needed to introduce non-linearity into the network. There are many functions that are commonly used as activation functions. In general, in ANN, it is not necessary to use the same activation functions for all neurons. Most common activation functions include, step function, linear function, and sigmoid function.

Based on the connection pattern between layers and neurons, ANN can be classified into two types: Feed Forward networks and Recurrent/Feedback Networks. Feed Forward Neural Network (FFNN) is artificial neural network; allow data travel from input nodes to output nodes in only one direction. The data transfers from layer to next layer when conduct required calculations; thus the output of any layer does not affect the same layer. But in Recurrent/Feedback Networks, the data travel from node to node in two directions.

2.7 Radial Basis Function Neural Network (RBFNN)

Radial Basis Function Neural Network (RBFNN) was first introduced by Broomhead and Lowe in 1988 [43]. It's a type of feedforward network and type of supervised learning.

RBFNN model is spired from the Cover's theorem [44] in 1965: *"A complex pattern classification problem cast in high-dimensional space nonlinearly is more likely to be linearly separable than in a low dimensional space".* The RBFNN has three layers feed-forward fully connection: the input layer, the hidden layer, and output layer. Figure

2.4 illustrates the architecture of RBFNN that including three layers. The input data flow from the input layer to send the information to the hidden layer. In the hidden layer, the neurons are activated depending on the distance between each input pattern and the centroid stores each hidden neuron, which determines the structure behavior of network. Different types of activation functions used in the hidden layer. The output layer consists of one neuron or more which calculates the linear sum of values of the hidden neuron and its outputs [45].

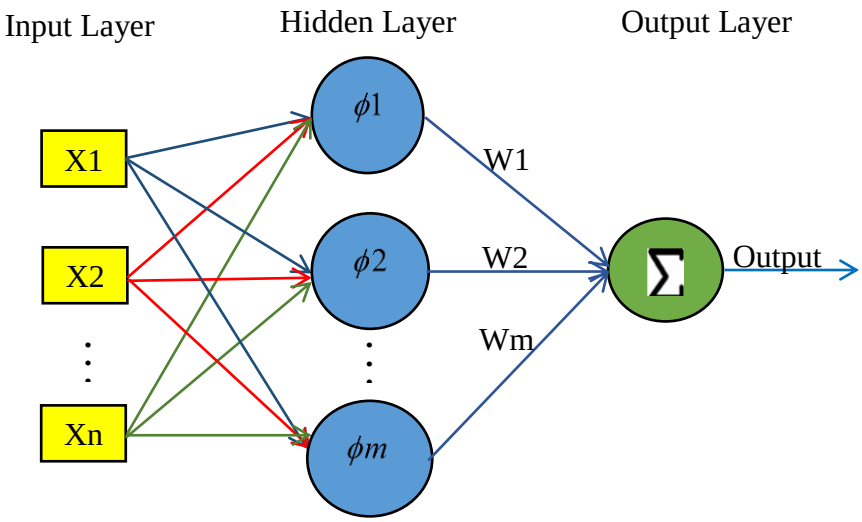

Figure 2.4: RBFNN architecture

RBFNN was originally performing as exact interpolation of a set of data points in multidimensional space [46], for every input vector data must be mapped exactly to the corresponding target vector (output function passes through all data points) as illustrated in figure 2.5. Also, exact interpolation can perform by assign one basis function for each data point, and then assign the weights for the linear combination of basis functions. For noisy data, the exact interpolation function highly oscillates values. On the other hand, for exact interpolation the numbers of basis functions are equal to the number of input data, therefore the mathematical computation can become costly [47].

Radial Basis Function Neural Networks are the type of neural networks whose

activation functions in the hidden layer are radially symmetric, which means that the output depends on the distance between input data vector and weights vector, which is called the center. Due to their simple structure network and faster learning algorithm, RBFNN has been used in many areas, such as function approximation, system control, speech recognition, classification [48], time-series prediction, and curve fitting.

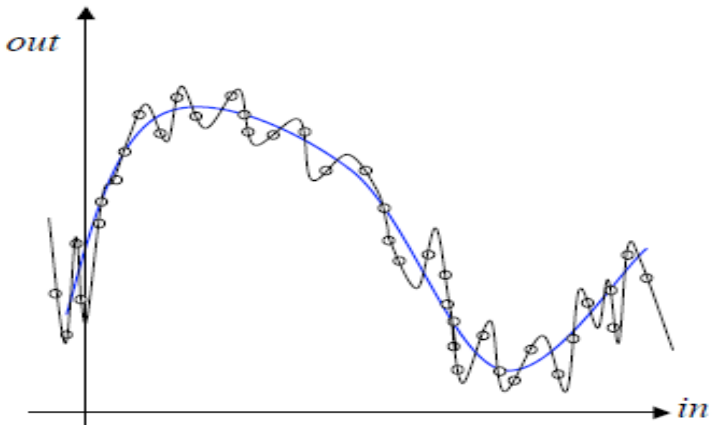

Figure 2.5: Exact Interpolation using RBF

In this thesis, we used Gaussian function in RBFNN as activation function. Both the position and shape of Gaussian function is more flexible to adjust compared with other activation functions [49].

Two categories of parameters need to be determined in RBFNN: the first category is the center and radii, and the second one is the connection weights between the hidden layer and output layer. The Gaussian activation function in the hidden layer (ϕ) is calculated as follows:

$$\phi = \exp\left[\frac{-(x-c_j)^2}{r_j^2}\right] \qquad (2.4)$$

Where $x = (x_1, x_2, x_3 \ldots, x_n)$ is the input data, c_j is the center of j-th hidden neuron and r_j is the width of j-th hidden neuron.

The output of RBFNN is calculated as in the following expression:

$$Y = \sum_{k=1}^{m} W_{jk} * \phi_j(x) \qquad (2.5)$$

Where $k=1, 2, 3 \ldots, m$ is the number of nodes in the hidden layer, W_{jk} is the connection weights values between the *j-th* hidden layer nodes and the *k-th* output nodes.

One important side of RBFNN determines the suitable number of neurons in hidden layer, which affects the network complexity and the generalization of RBFNN. If the number of neurons in RBFNN is too small, the accuracy of the output will decrease. On the other hand, if the number of neurons is too large, this cause over-fitting for the input data [50]. The drawback of RBFNN is that it treats all the input parameters with the same level of significance [51]. One key advantage of RBFNN can be more efficiently training compared with other feed forward networks, this is due to the training of the input and output layers for RBFNN that are trained separately [52].

2.8 Bio Inspired Optimization Algorithms

Bio-inspired algorithms involve many major types of well-known algorithms. The most important of them all are the following two types: Evolutionary Algorithm (EA) and Swarm Intelligence (SI). Generally, this relies basically on the science fields of biology, computer science, and mathematics. Bio-inspired algorithms are based on some characteristics of a biological system [53]. Bio-inspired algorithms are based on two factors of self-organization and complex systems. Bio-inspired algorithms are metaheuristics that based on mimics the nature for solving optimization problems, for example, a genetic algorithm is inspired by the human evolution, swarm intelligence is inspired by the behaviors of animals, and ANN is mimic the human brain.

2.8.1 Evolutionary Algorithms (EA)

Evolutionary Algorithm (EA) [54] is an optimization algorithm inspired by biological natural evolution. Nowadays, EA is the most widely used in Artificial Intelligence. EA can be considered as the most search techniques used for complex problems. EA is also considered as the best nature-inspired algorithms compared with others. EA can be divided into three kinds as follows: Genetic Algorithms, Genetic Programming, and Evolutionary Strategies. The most important widely used kind of EA is Genetic Algorithms (GA). The main components of any evolutionary algorithm are choosing initial population, parents and survivors, reproduction, mutation, recombination, offspring, and selection. The general scheme for EA is illustrated in figure 2.6.

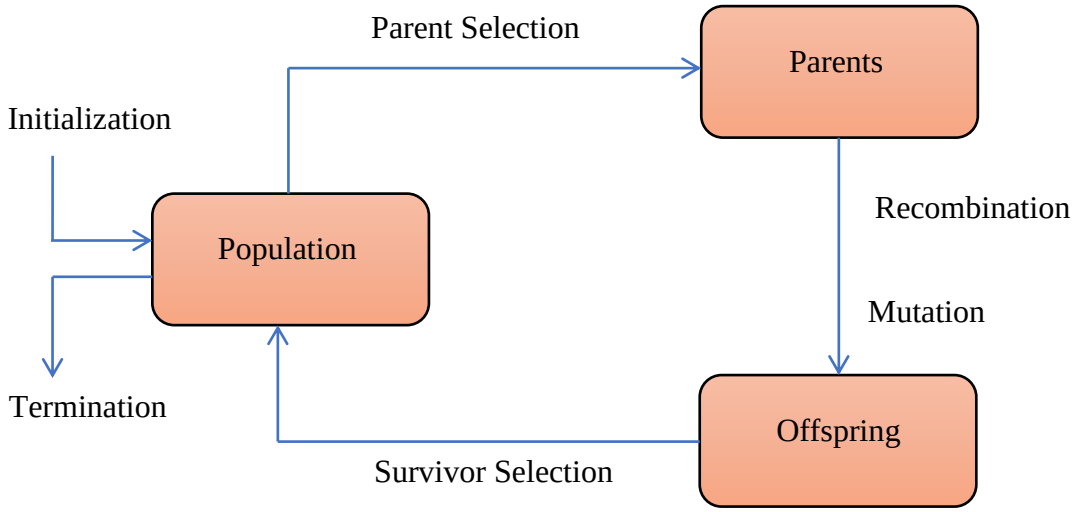

Figure 2.6: General schema for Evolutionary Algorithm [55]

In EA we must distinguish between two types of parameter values: parameter tuning and parameter control. The main objective of parameter tuning is finding best values for the parameters before start runs the algorithm and then running the algorithm using these parameters values, as for parameter control, it's used as an alternative to starting a run with initial parameter values which are changed during run time.

Evolutionary algorithms can be used in many fields as, chemistry, art, economics, biology, marketing, social sciences, physics, robotics, and engineering.

2.8.2 Swarm Intelligence (SI)

Swarm Intelligence (SI) was first introduced by Beni and Wang in 1989 in robotics system field [56]. Swarm intelligence (SI) is an Artificial Intelligence (AI) technique that is concerned to design intelligent multi-agent systems; this system is inspired by a different swarm of animals and insects such as ants, bees, fishes, and birds. SI is can be used to solve the optimization problem in the fields of management, engineering, and financial. The researchers have developed numerous algorithms in this field such as Ant Colony Optimization and Particle Swarm Optimization which is the most famous swarm intelligence algorithm. SI is based on self-organization system and decentralized behavior. Swarm Intelligence algorithms have the ability to solve complex tasks in the form of swarms. Swarm Intelligence has many features for instance fault tolerance, feedback, parallelism, adaptation, scalability, modularity, and speed.

In this thesis, we used Particle Swarm Optimization (PSO) to combine with RBFNN in our experiments.

2.9 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was firstly proposed by Eberhart in 1995 [57] which is the most famous swarm intelligence algorithms. PSO mimics the behaviors of birds flocking. In bird flocking, a group of birds is randomly searching food in the specific area. All the birds don't know where the food is found. But they know how far the food in each step.

The population in PSO is called swarm that is initialized with a group of random values called particles (solutions) and then searches for the optimal position of particles by

updating it for all iterations. In all iterations, each particle is updated by following two best fitness values are evaluated by using proper fitness function according to the problem. The first value is the best position it has achieved so far for each particle; this value is called personal best (pbest). Another value is the best position for the entire swarm obtained so far by any particle in the swarm; this best value is called a global best (gbest) [58]. Often all particles are initialized randomly and run the PSO algorithm number of times until achieve stopping criteria. The stopping criteria to stop training PSO is finding the optimal solution or exceeds number of predeterme iterations [59].

The basic algorithm of PSO is as following steps:

- **Initialize** each particle i of the swarm, with random values for the position *(Xi)* and velocity *(Vi)* in the search space according to the dimensionality of the problem.
- **Evaluate fitness** value of particle by using fitness function.
- **Compare** the value obtained from the fitness function from particle *i*, with the value of Pbest.
- **If** the value of the fitness function is better than the Pbest value, then update the particle position to take the place of Pbest.
- **If** the value of Pbest from any particle is better than gbest, then update gbest = Pbest.
- **Modify** the position X and velocity V of the particles using equations (2.6) and (2.7), respectively.
- **If** the maximum number of iterations or the ending criteria isn’t achieved so far, then return to step 2.

$$V_{id}(t+1) = \omega * V_{id}(t) + c_1 r_1 \left(P_{id}(t) - X_{id}(t)\right) + c_2 r_2 (P_{gd}(t) - X_{id}(t)) \quad (2.6)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (2.7)$$

Where $i = 1,2,\ldots,m$ is the number of particle, $d = 1,2,\ldots,n$ is the dimension of problem, $t+1$ is the current iteration number, t is the previous iteration number, ω is the inertia weight, c_1 and c_2 are the acceleration constant factor usually between (0,2), $P_i = (P_{i1}, P_{i2}, \ldots, P_{in})$ is the best previous position of particle i known as the personal best position (pbest) , $P_g = (P_{g1}, P_{g2}, \ldots, P_{gn})$ is the position of the best particle among all the particles in the swarm known as the global best position (gbest) and r_1 and r_2 are random numbers distributed uniformly in (0, 1).

There are two basic topologies to run PSO algorithm: Global best (gbest) and Local best (lbest) topologies. Global best (gbest) is affected by the best particle in the entire swarm. It uses a star network topology as illustrated in figure 2.7, getting the best solution based on all particles in the swarm. A feature of gbest method is quick converges to the best solution but may be trapped in local minima [60].

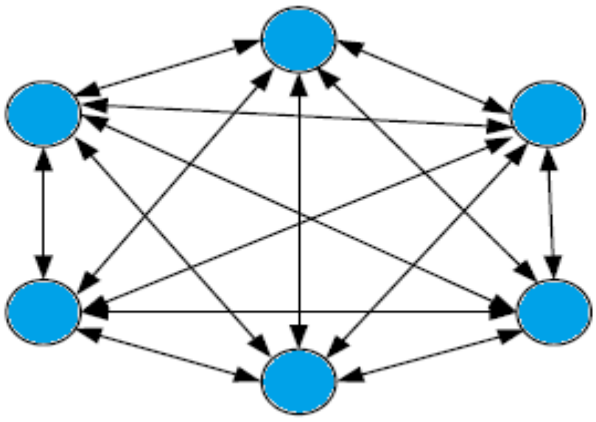

Figure 2.7: Star topology for gbest [60]

Local best (lbest) is affected by the best particle in the selected neighbor particles. It uses a ring network topology as illustrated in figure 2.8, getting the best solution based

on only neighborhood particles. In lbest, it may slowly converge but the feature of this method it gives less chance of local minima [60].

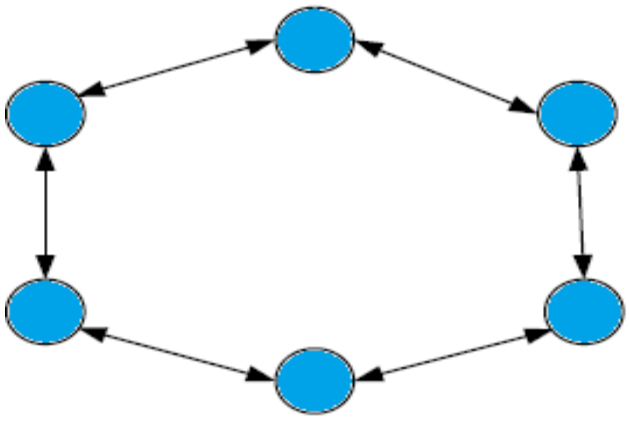

Figure 2.8: Ring topology for lbest [60]

Many techniques were proposed to enhance PSO; the major of these techniques are shown in details. At the beginning, inertia weight ω was first introduced by Shi and Eberhart in 1998 [61] in order to control the search speed and make particles converge to local minima quickly. Inertia weight is used to influence on the current velocity based on the previous velocity [62].

Kennedy proposed a method that determines the maximum velocity value (V_{max}), the velocity values bounded limited between ($-V_{max}$, V_{max}) [63]. This limitation controls the velocity value even prevents it does not become a very large value. Indeed determining the best V_{max} is a challenge according to the problem.

In 1997, Kennedy and Eberhart proposed Binary Particle Swarm Optimization (BPSO) that modifies the basic of PSO [64] to solve complicated problems. In BPSO, the particles represent a position in binary space and particle position vectors can take the binary value (0 or 1). The advantages of PSO are [65]: simplicity, low computational, fast convergence speed than other algorithms, fast optimization method due to need to

adjust few parameters, high robustness, strong global search capability, does not require gradient information and is easy to implement.

PSO has been successfully applied in many applications such as Communication Network, Function Optimization, Artificial Neural Network training, Fuzzy System Control, Clustering, Classification, Financial, Prediction System, and Robotics.

The basic PSO parameters are swarm size, iterations number, velocity, and acceleration coefficients. There are some parameters in PSO algorithm that may affect its performance and some parameters may have small or no affect its performance. Swarm size is the number of particles in the swarm; large swarm size may reduce the number of iterations to obtain a good result but may increase the execution time and computational complexity. Iteration numbers are affected the performance algorithm. If the iteration numbers are too small they don't obtain a good result, but if iteration numbers are too large increase computational complexity and execution time. Acceleration coefficients c_1 and c_2, these coefficients are responsible for maintaining the stochastic influence of the cognitive and social components of the particle velocity respectively. The c_1 is responsible of how much a particle confidence itself, while c_2 is responsible for how much a particle confidence with its neighbors [66].

2.10 ANN and PSO

Artificial Neural network (ANN) is used with combined of PSO in many applications which include: neural network control for nonlinear processes [67], design of radial basis function networks [68], feed forward neural network training [69, 70], design of recurrent neural networks [71], cellular neural networks [72], and neuron controllers [73]. In this thesis, we used PSO to find the optimal values for some parameters for RBFNN. In most cases, using PSO to train ANN, it is faster and gets better results.

CHAPTER 3

LITERATURE REVIEW

3. Literature Review

In this chapter, we show related works about hybrid RBFNN and PSO in many sides and show related work for Spam Filtering methods using machine learning Techniques.

3.1 Hybrid RBFNN and PSO

The advantages of evolutionary algorithms and swarm intelligence in function approximation are that these algorithms behave robustly and efficiently in broad domains. PSO algorithm is widely used in resolution problems of optimization, and it has become popular depends on to its speed of convergence and the simplicity of its implementation.

There are several studies that combine RBFNN and PSO. The authors in [74] combined Incremental RBFNN with Particles Swarm Optimization (IRBF-PSO) to improve the accuracy of classification in intrusion detection system; PSO was used to find optimal values for weight and bias. In [75], the authors have proposed a PSO-RBF to control the design of RBF Networks and evaluate parameter of RBF to solve pattern classification problems, in this model PSO used to finds the size of network, in addition, to optimizing the center and the width for each basis function.

The authors in [76] presented an intelligent fault diagnostic approach for a steer-by-wire (SBW) system. They used a clustering algorithm to find the number of the centers in RBF, and then use PSO algorithm to optimize the centers and widths parameters for RBF. PSO-RBFNN proposed in [77] concentrated on finding the best center values of RBF using PSO in the induction motor parameter system. The experimental results of PSO-RBFNN show this method can enhance the induction motor performance accuracy. In [78] present new approach called PSO-ELM algorithm that combines PSO and extreme learning machine (ELM) to train RBFNN. Here PSO is used to find optimal parameters for center and width of neurons, while ELM uses to find the values

connection weights. MuPSO-RBFNN in [79] is present a novel learning algorithm that combines an RBFNN and PSO with mutation operation and genetic algorithm to train RBF parameters (center, width). To evaluate the performance, they use Sum Square Error (SSE) function in experiments. Several experiments are conducted for function approximation; results show that MuPSO-RBFNN approach is better than other approach.

As for in [80] they presented a time variant multi-objective PSO of RBF called TVMOPSO for diagnosing the medical diseases. TVMOPSO is used to optimize centers and weights of RBF network. In [81] they proposed the transformer fault diagnosis approach based on RBF neural network improved by PSO algorithm. PSO algorithm is used to optimize (center, width, and weight) values of RBF neural network. Hybrid PSO-RBF proposed in [82] was used to evaluate the levels of underground water quality in the ten monitoring points of the black dragon hole, the value of weight for output layer determined by PSO.

In [83] the presented method growth the number of RBF, the basic idea in this model increases the model complexity by add one neuron at each iteration, which is based on the PSO algorithm to optimize parameters of the newly RBF neuron only, while save the parameters of all older RBF neuron that optimized earlier and gradually decreased root mean square error RMSE at each iteration. While in [84], Propose a newly evolutionary search technique called Quantum-Behaved Particle Swarm Optimization (QPSO), RBF used in the training phase. The proposed QPSO was testing on nonlinear system identification problem. As for in [85], presents an approach to solving the path planning problem treated as an optimization problem by combining RBFNN and PSO algorithm. Here the input pattern is a group of trajectory constraints depends on control

points, which explore the used robot to approximate functions with PSO-RBF approach, towards getting smooth and collision-free paths. Moreover, the proposed combined fitness function, it has three goals achieve minimum mean square RBF function approximation error, minimize the obtained path length, and avoid collisions.

In [86], they present a system that applies to the electric gas pressure regulator; this system is to improve PID intelligent control algorithm. The system is used the improved RBFNN which is based on PSO algorithm to produce an online adjustment of PID parameters. System simulation result and theoretical analysis result show that the algorithm reduces the step response time and improves tracking performance. In [87], the authors present adaptive multi-objective PSO to RBFNN design namely Adaptive Multiobjective PSO-RBF (AMOPSO) to optimize the architectures and connections of the network. The structure of RBFNN and its parameters are encoded to the particles in PSO comparable results similar to multiobjective PSO. The presented approach optimizes the network structure and the network performance at the same time, in terms of hidden neurons and connection weights. They conduct experiments on two benchmark datasets. The obtained results show that this approach provides effective methods for training RBFNNs that is competitive with PSO based multi-objective algorithm. In [88], this study proposed a method for electrocardiogram (ECG) beat classification depend on particle swarm optimization (PSO) and radial basis function neural network (RBFNN). Here the centers and the widths of RBFNN are mapped as a particle in the solution space. Initialized the training set and optimum RBFNN structure is found. Then ten experiments are conducted on the test set with a different number of hidden neurons and the results are compared with the K-Means algorithm. To compare two methods set the number of clusters as the number of neuron centers in K-Means

algorithm. From the experiment result, it can be noted that the proposed method classifies the ECG beats with a smaller size of the network without affecting on classification performance. Another important notation is that the proposed method can classify the ECG beats with a good performance by using only 6 hidden neurons. Note that the proposed method for classification time of whole training set within 0.2250 s which is roughly half of the K-Means time.

Authors presented in [89] a particle swarm optimization (PSO) algorithm to automate the design of RBFNN, to solve pattern classification problems. PSO is used to develop RBFNN parameter and structure network. Here the important feature is the number of basis function adjusted dynamically according to the number of particles in PSO. Simulation results for the benchmark in the pattern classification show that the PSO-RBF outperforms in performance with other two learning algorithms in terms of network size and generalization performance. In [90], authors proposed a nonlinear time-varying evolution particle swarm optimization (NTVE-PSO) system that used for time series prediction of a practical power system. This system was developed by combined RBFNN and particle swarm optimization (NTVE-PSO) algorithm. NTVE-PSO method is used to determine the optimal structure of the RBFNN to predict time series, in NTVE-PSO algorithm using the nonlinear time-varying evolutionary functions for adjusting inertia and acceleration coefficients. The data for the experiments getting from the different practical load types of Taiwan power system (Taipower) are used for time series prediction of one-day ahead and five days ahead. Simulation results explain that the NTVE-PSO proposed system has better prediction accuracy and computational efficiency of time series data than the other PSO-RBFNNs methods. Finally in [91], authors proposed hybrid RBFNN with Particle Swarm Optimization (PSO) used for

error rate, better convergence, and classification results. RBFNN hybrid learning includes two stages; the first stage determines the RBF centers and widths by using a k-mean algorithm and standard derivation respectively, the second phase is determined connections weights between the hidden layer and the output layer by using Least Mean Squares (LMS) and gradient-based methods. Then the RBFNN hybrid with PSO is performed by optimizing the centers, widths, and weights of RBFNN. The experiment conducted on five datasets (XOR, Balloon, Cancer, Iris, and Ionosphere) then the results for training, testing and validation show the PSO efficiency in enhancing RBFNN compared with conventional Backpropagation.

3.2 Spam Filtering Techniques using Machine Learning

Machine learning is a subfield of computer science that is the type of artificial intelligence (AI). It provides computers to be able to learn without using explicitly programmed. Machine learning interested in developing computer programs that can teach themselves to grow and change when new data coming. The aim of this program is to be able to classify data that wasn't seen before, by learning patterns from training data on similar data. The machine learning process is similar to the data mining process in terms of both processes need to search through data to find specific patterns. One of the uses machine learning techniques is classification spam email, these techniques need to predefined classified training data. Several machine learning techniques used for SPAM filterings such as artificial neural networks, genetic algorithms, clustering techniques, artificial immune systems and decision trees.

The most widely used methods for spam filtering are artificial neural networks (ANN) and Bayesian method, also support vector machines, there are many techniques described for mail filtering and spam detection.

Initially, in [92], authors presented a LINGER based on ANN; LINGER is an ANN-based system used for automatic email classification. Although LINGER was tested in the email classification field, LINGER is a public architecture for all text classifications. LINGER is flexible, adaptive system and compatible for most operations. ANN can be effective to be used for spam filtering and automated email filing into mailboxes. In [93], authors proposed spam filtering approach for the Turkish language in particular. Their approach is dynamic and depends on Artificial Neural Networks (ANN) and Bayesian Networks. This approach is dealing with the characteristics of the incoming e-mails. Through their experiments for 750 e-mails (410 spams and 340 hams) the accuracy was achieved about 90%. In [94], they combined Support Vector Machine (SVM) and Genetic Algorithm (GA) namely GA-SVM, GA is used to select features that are most favorite to SVM classifier. The experiments show that GA-SVM approaches better results than main SVM.

In [95], authors presented a novel approach of spam filtering includes the seven several steps that depends on using the history of previous emails and spam emails which are specific for each mailbox of the user. Using the knowledge base, detection of spam emails is performed by using artificial neural network techniques. It also using keywords list to get some words in the incoming mail, then perform the detection operation. The proposed approach works well with all kinds of spam emails (text spam and image spam). Experiments results show that the detected spam at least 98.17 %. But the limitation of this approach is needed higher memory space and more hardware for execution. So to implement this approach for large mail servers, we need intelligent mail servers. In [96], authors presented Continuous Learning Approach Artificial Neural Network CLA_ANN, which includes modifying core modifications on ANN in the

input layers, which allow the input layers to be changed over time and to replace useless layers with new promising layers which give best results. The experiment result of CLA_ANN by using 300 input layers and using Spam Assassin dataset, achieves results with 0.534 % false positive and 3.668% false negative. In [97], authors presented a new technique for filtering spam; this technique consisted of a single perceptron that was designed to distinguish between spam and legitimate mail messages. The perceptron algorithm gives suitable detection rates; this is due to the incorporation of a continuous learning feature. The results show that the best false positive value is found when the number of iterations is 900.

In [98], authors proposed Gaussian mixture functions and particle swarm optimization (PSO), called PSO-LM approach which is a new learning method for process neural networks (PNNs) based on the Gaussian mixture functions and particle swarm optimization (PSO).According to experiments results, PSO-LM had better performance on time-series prediction and pattern recognition than basis function expansion based learning (BFE-LM) and back propagation neural networks (BPNNs). But in PSO-LM approach, it needed more computations for the global search strategy of PSO. In [99], authors presented a novel approach that links Self-Organized Feature Map (SOFM) and principal component analysis (PCA). In this approach, each email is represented by a series of textual and non-textual features. PCA is used to choose the most important textual features. Finally, the selected features and the non-textual features should enter as input to train SOFM to classify emails into a spam or normal. The experimental results demonstrate that the approach will increase the classification accuracy compared with some other traditional classification methods. In [100], Support Vector Machine (SVM) can be used in spam filtering because in SVM separate the data into two

categories. Standard SVM includes minimization of the error function and improves the classification accuracy but the percentage of misclassification for legitimate emails is high. To solve this problem, in this paper proposed spam filtering depends on weighted support vector machines. Experimental results illustrate the proposed method can improve the spam filtering performance effectively.

In [101], authors proposed a novel model RBF-SF to detect and classify email messages. The experiments are conducted on two benchmark spam testing. The empirical results of RBF-SF illustrate the performance comparison with other popular text classifiers have illustrated that the proposed model is capable of filtering spam efficiently. In [102], the authors present a study that uses support vector machine (SVM) for classifying email message. SVM experiments are conducted on two data sets: one data set to use 1000 features and another data set to use 7000 features. Also, the experiments are conducted on three other classification algorithms: Ripper, Rocchio, and boosting decision trees to comparing it with SVM. SVM performed best when using binary features. For both data sets, boosting decision trees and SVM are enhancing test performance for the accuracy and speed. It should be noted that the SVM has significantly less training time. Lastly in [103] propose a combined simple artificial immune system (SAIS) with particle swarm optimization (PSO) for using in spam filtering. PSO was used with the mutation to increase the strength of immune system's searches to find the best class of email message. The experiment results show that the Hybrid SAIS and PSO are superior in terms of classification accuracy than SAIS system.

CHAPTER 4

THE PROPOSED METHOD

4. The Proposed Method

The beginning of this chapter in particular focuses on the process of combining RBF and PSO; it then moves to describe the proposed approaches in details and all algorithms used in our approaches.

4.1 Methodology

In this thesis, we used an incremental method to determine the suitable number of neurons in the hidden layer. The process starts with a little number of neurons and increases the number of neurons gradually until reach to the best solution. The training process is terminated in three cases: when exceed threshold value of error, reach to the maximum number of iteration, and changing of weights are met.

4.1.1 Hybrid RBF and PSO

Traditional RBFNN learning suffers from the problem of how to find the optimal solution in local space to identify RBFNN parameters. If these parameters are not selected correctly, it will reduce the accuracy and even produce network divergence [104]. PSO algorithm is characterized by several features including fast convergence, global search ability and does not require gradient information. It therefore can be used to enhance convergence speed, stability of RBFNN and finding the optimal architecture of RBFNN. Several studies used PSO for training RBFNN to deal with difficult tasks in many areas. For instant, the PSO algorithm is applied to optimize the RBFNN parameters in the following fields function approximation [105], image processing [106], pattern clustering [107], and time series prediction [108]. Figure 4.1 illustrates the system chart of general steps for hybrid RBFNN and PSO.

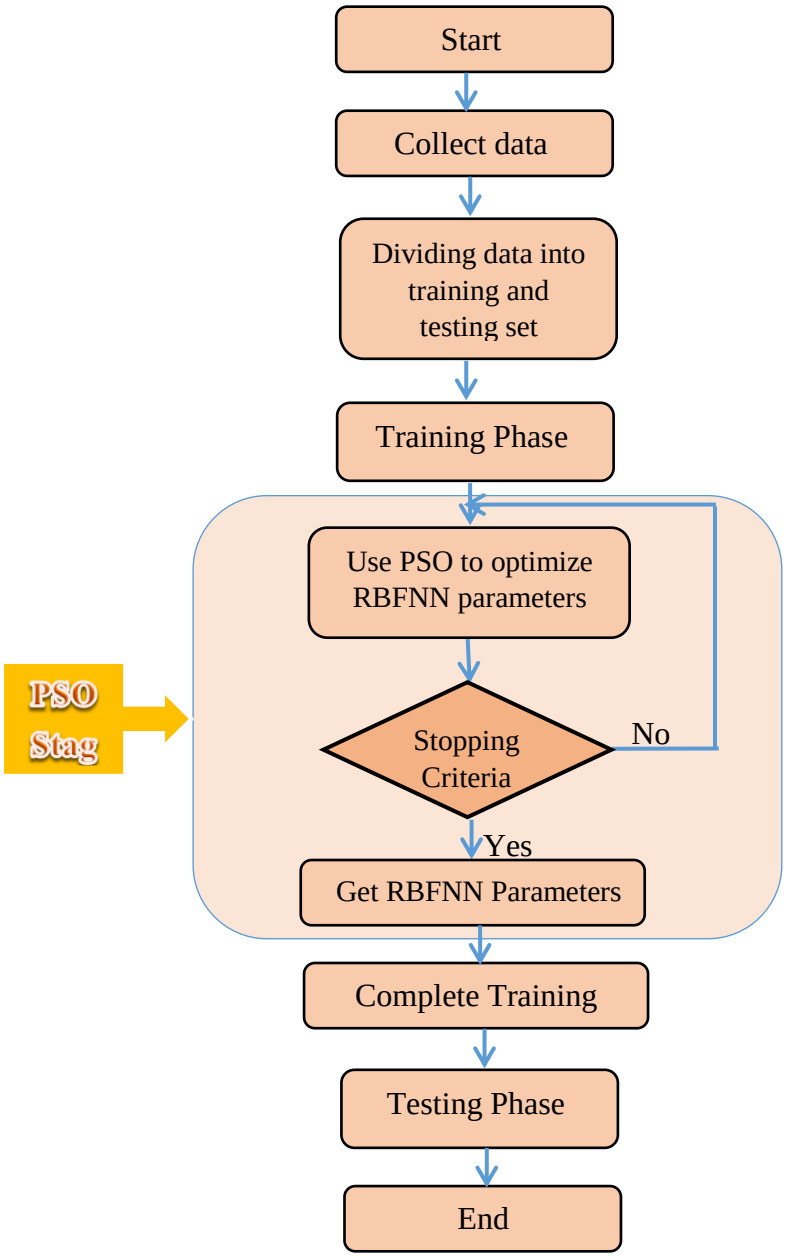

Figure 4.1: System chart of general steps for hybrid RBFNN and PSO

PSO has been used to improve RBFNN in several sides like network architecture, learning algorithm and network connections. In this work, we use PSO algorithm to optimize some parameters of RBFNN. However, this thesis proposes two hybrid approaches namely (HRBFN-PSO and HC-RBFN-PSO). These approaches are combining RBFNN and PSO to training RBFNN. HRBFN-PSO approach is used for function approximation and time series prediction; whereas HC-RBFN-PSO is used for classification data. One of the most important parameter that determine the efficiency

and accuracy results in the RBFNN algorithm is the center. Usually, the centers of RBFNN are determined randomly or by using the famous traditional k-means clustering algorithm. In this work, we used PSO algorithm to get the best values of RBFNN centers to enhance the output of RBFNN.

4.1.2 Proposed HRBFN-PSO Approach

In this thesis, we present a hybrid approach that combined RBFNN with PSO which called HRBFN-PSO, used for function approximation and time series prediction, to evolve the optimization accuracy for RBFNN. This approach is used to achieve several goals including reducing the number of neurons in hidden layer of RBFNN, improving the prediction process which is the error value between target output and real output in RBFNN, this value of error is calculated based on equation 4.1, finally selecting the best values of RBFNN centers using PSO.

$$E = T_i - Y_i \qquad (4.1)$$

Where E is the error value, T_i is the target value of the output for each i^{th} data point, and Y_i is the real output value of the i^{th} data point.

In RBFNN for function approximation to reach the best accuracy for results is used fitness function measured, which takes the error between the output (target) and the real output. There are many fitness functions measured that can be used to calculate the error such as Mean Square Error (MSE), Root Mean Square Error (RMSE), Sum Square Error (SSE) and Normalized/Root Mean Squared Error (N/RMSE). In this thesis, we used two fitness functions namely, Mean Square Error (MSE) as shown in equation 4.2 and we used Sum Square Error (SSE) as shown in equation 4.3.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(T - Y)^2 \qquad (4.2)$$

$$SSE = \sum_{i=1}^{n}(T - Y)^2 \qquad (4.3)$$

Where n is the number of input data, T is the target output and Y is the real output.

HRBFN-PSO approach is used for function approximation for any simple and complex mathematical function and time series prediction. The training process for this approach (HRBFN-PSO) is the mapping between the input-output data of the RBFNN. Initially, the centers of RBFNN hidden neurons has been chosen randomly, then run PSO algorithm number of times to obtain the best values of centers. These centers are encoding in PSO as shown in table 4.1 that represents encoding each particle in PSO. The remaining parameters of RBFNN, we used tradition algorithms namely KNN and SVD to optimize radii and weight respectively. HRBFN-PSO approach architecture is illustrated in figure 4.2.

Table 4.1: particle encoding in HRBFN-PSO approach

$Center_1$	$Center_2$	$Center_3$	$Center_4$	…..	$Center_n$

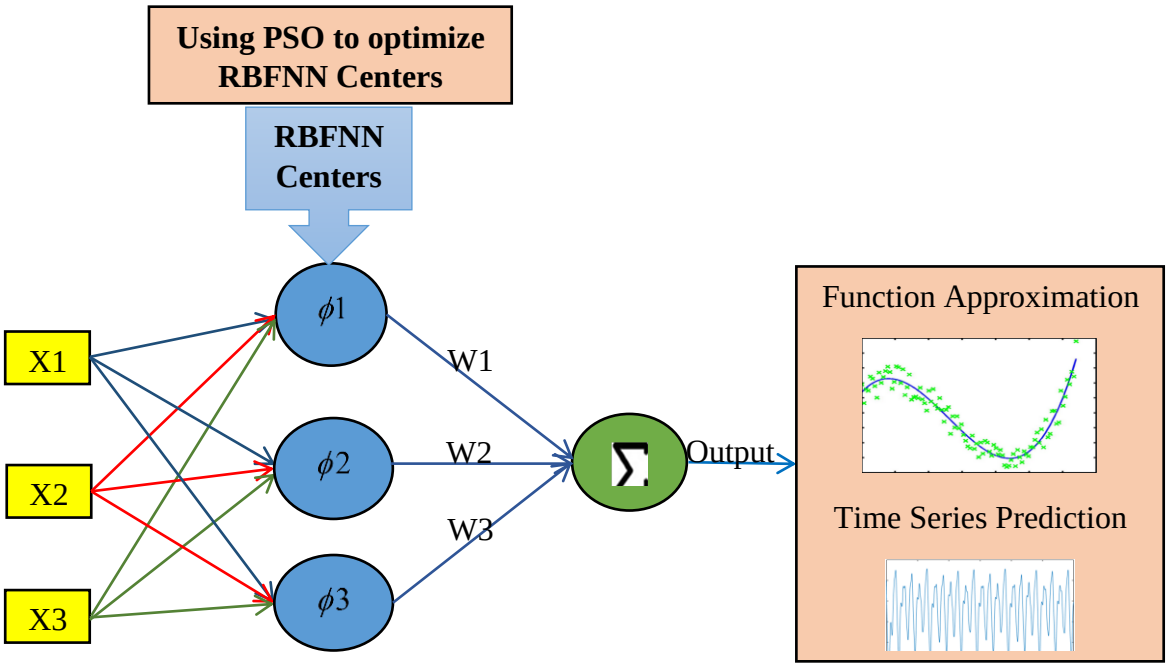

Figure 4.2: HRBFN-PSO architecture for function approximation and time series

Every single solution of PSO called a particle. In PSO, we need fitness function to evaluate the particles to reach the optimal solution. Fitness function used for PSO in this approach is Mean Square Error (MSE) that is illustrated in equation 4.2. Figure 4.3 illustrates the HRBFN-PSO pseudo code.

Start training RBFNN
Enter Number of Hidden Neurons
Initialize RBFNN parameters
- *Initialize the centers c randomly from input data*
- *Use **K-nn** to initialize Radii r*
- *Use **SVD** to initialize Weights w*

Start optimizing centers of RBFNN using PSO.
Initialize particles position and velocity randomly between (0, 1)
While not reach the maximum numbers of iteration ***do***
For each particle do
Calculate fitness value (MSE between Real output of RBFNN and Target)
If fitness value is better than best fitness value ***pbest*** in particle then
Set current position as ***pbest***
End IF
Select gbest of the particle which the best fitness value among all particles in current iteration
Calculate particle velocity based on equation (2.6).
Update particle position (center) based on equation (2.7).
End For
End While
Take the ***gbest*** values of particle as centers ***c*** of RBFNN

Complete training RBFNN using K-nn and SVD.
Calculate Real Output of RBFNN
Calculate Error value (RMSE or SSE)

Figure 4.3: HRBFN-PSO pseudo code

Inertia weight ω for PSO often has restricted between two numbers during execution. In this work, we calculate the inertia weight by using equation 4.4.

$$\omega = \omega_{\max} - \left(\frac{\omega_{\max} - \omega_{\min}}{T_{\max}} \right) * t \qquad (4.4)$$

Where ω is the inertia weight, ω_{max} is the maximum ω value, ω_{min} is the minimum ω value, t is the current iteration number and T_{max} is the maximum iteration number.

In this approach, the training RBFNN is stopped if either reaches the maximum number of iterations or got the threshold value of error. Here, we used the Root Mean Square Error (RMSE) to stop training RBFNN when the RMSE value is less than threshold value, RMSE is presented as illustrated in the following expression:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (T - Y)^2} \qquad (4.5)$$

Where n is the number of input data, T is the target output and Y is the real output.

4.1.3 Proposed HC-RBFN-PSO Approach

Classification is the method of categorizing data into various classes by distributing data into categories according to its characteristics. Data classification enables the separation and classification of data based on data set requirements of the objectives. The aim of the classification process is to assign each input patterns to one suitable class based on the characteristics.

Artificial Neural Networks is very suitable to deal with problems of classification. It is an adaptive system that changes its structure that depends on input data during the learning process. ANN learning process will continue until the classification accuracy rate exceeding the predetermined value (threshold value).

As a main aim of this approach, it states that to reach the best classification accuracy, each output node computes a sort of score for the associated category. Generally, classification decision is done by assigning the input to the category with the highest score. The score is computed by taking a weighted sum of the activation values from every RBFNN neuron. Equation 4.6 illustrates how to calculate Classification accuracy for RBFNN.

$$Accuracy = \frac{(TP + TN)}{N} \tag{4.6}$$

Where *TP* is the True Positive, *TN* is the True Negative and *N* is the size of samples data.

The input of email classification task can be shown as two-dimensional matrixes, the row axis represents the email messages and the column axis represents the features (attributes) for this message. Email classification tasks are usually divided into several steps; the first step is data collection and representation of email messages. The second step is email feature selection and feature reduction to remove some features to reduce the dimensionality. Finally, the email classification process step comes to classify this message as spam or non-spam. The basic idea for spam filtering technique is illustrated in figure 4.4.

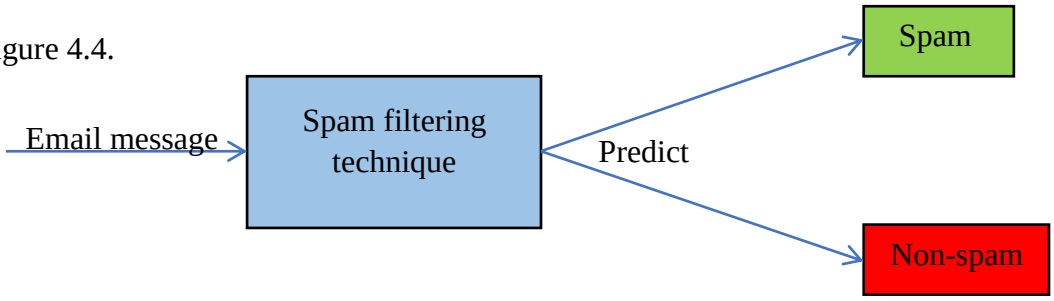

Figure 4.4: Basic idea for spam filtering

In this work, we employed content-based filtering technique for email classification by checking the content of email messages then extract useful information, and then dividing the whole information to words. After that, extracting the feature vector of spam information and translate it into the suitable form of vector format.

HC-RBFN-PSO is useful in terms of reducing the number of neurons and improve classification accuracy, we used PSO algorithm to select the best values of RBFNN centers. These centers are encoding in PSO as shown in table 4.2 that represent encoding each particle in PSO. The remaining parameters of RBFNN, we used tradition algorithms namely KNN to optimize radii values and SVD to optimize weight values. Every single solution of PSO called a particle. In PSO, we used fitness function to evaluate the particles to reach the optimal solution that is RMSE that is illustrated in equation 4.5. Figure 4.5 illustrate the HC-RBFN-PSO architecture.

Table 4.2: particle encoding in HC-RBFN-PSO approach

# Center	$Attribute_1$	$Attribute_2$	$Attribute_3$	.....	$Attribute_m$
$Center_1$	C_1A_1	C_1A_2	C_1A_3	.....	C_1A_1
$Center_2$	C_2A_1	C_2A_2	C_2A_3	.....	C_2A_2
$Center_3$	C_3A_1	C_3A_2	C_3A_3	.....	C_3A_3
⋮	⋮	⋮	⋮	⋮	⋮
$Center_n$	C_nA_1	C_nA_2	C_nA_3	.....	C_nA_m

Where C_1:C_n are Centers and A_1:A_m are Attributes of dataset

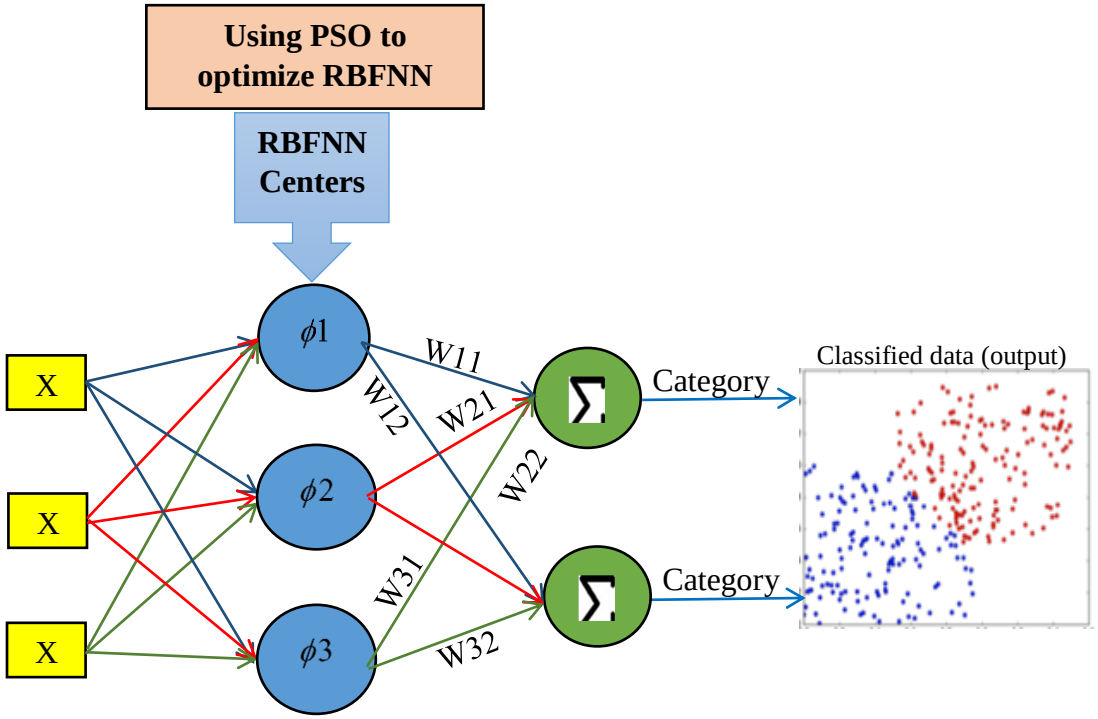

Figure 4.5: HC-RBFN-PSO architecture for classification

Figure 4.6 shows the HC-RBFN-PSO pseudo code.

Start training RBFNN

Enter Number of Hidden Neurons

Initialize RBFNN parameters.

- *Initialize the centers c randomly from dataset.*
- *Use **KNN** to initialize Radii r.*
- *Use **SVD** to initialize Weights w.*

Start optimizing centers of RBFNN using PSO.

Initialize particles position randomly from the dataset and initialize velocity randomly between (0, 1)

While not reach the maximum numbers of iteration ***do***

For each particle do

Calculate fitness value (RMSE between Real output of RBFNN and Target)

If fitness value is better than best fitness value ***pbest*** in particle then

Set current position as ***pbest***

End IF

Select *gbest* of the particle which the best fitness value among all particles in current iteration

Calculate particle velocity based on equation 2.6

Update particle position (centers) based on equation 2.7

End For

End While

Take the ***pbest*** values of particle as centers ***c*** of RBFNN

Complete Training RBFNN using KNN and SVD

Calculate Real Output of RBFNN

Calculate Classification Accuracy

Complete Testing Phase

Figure 4.6 HC-RBFN-PSO pseudo code

To evaluate HC-RBFN-PSO approach, we used two datasets. These datasets are Wisconsin diagnostic breast cancer (wdbc) and Ionosphere. SPAMEBASE dataset used for spam filtering. For each dataset, split the data into two phases:

- Training set (70% data): this phase implemented to build up a model and used to find the optimal values of parameters.
- Testing set (30% data): this phase implemented to validate the model built and to assess the performance system (generalization).

Inertia weight ω for PSO often has restricted between two numbers during execution. Here in this work, we calculated the inertia weight by using equation 4.4 in order to control the search speed and make particles converge to local minima quickly.

In this approach, the training RBFNN is stopped if either reaches the maximum number of iterations or exceeds the required classification accuracy.

4.2 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is used in this work to determine the radii (r) of each RBFNN. KNN is a simple algorithm used for classification and regression [109]. KNN stores all available cases and classifies new cases based on a similarity measure. The commonly used similarity measure is Euclidean distance functions. Mathematical calculations regarding KNN algorithm are illustrated in equation 4.7.

$$D(X,Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \qquad (4.7)$$

Where *D(X,Y)* is the Euclidean distance between two objects *X* and *Y*, *i* is the current attribute, *n* is the number of attributes and X, Y are the objects.

KNN is a type of the lazy learning algorithms, where the function is only approximated locally and all computation continues until classification [110]. On the other hand, this

algorithm is successful in a number of classification and regression problems; such as handwritten digits and satellite image scenes [111]. The number k is used to decide how many neighbors influence the classification for new value.

4.3 Singular Value Decomposition (SVD)

In this work, we use Singular Value Decomposition (SVD) to optimize the weights of output layer for RBFNN. SVD is a powerful and useful matrix decomposition that has been used in many fields such as data analysis, reducing dimension transformations of images, data compression, signal processing, and pattern analysis [112]. If $A \in R^{mxn}$, there exist orthogonal matrices $S \in R^{mxm}$ and $H \in R^{nxn}$ such that:

$$S^T A H = diag(\sigma_1,...,\sigma_p) \tag{4.8}$$

Where *p* is the minimum of *(m,n)*, σ are the singular values of A. The use of SVD technique to calculate the optimal weights w of the RBFNN depends on using matrix notation described in the following reducing expression:

$$\vec{Y} = \vec{w}\ \Phi \tag{4.9}$$

Where *Y* is the real output of RBFNN, *w* are the weights vector and *Φ* is the Gaussian activation function matrix.

Using the next following expression:

$$A = H\ diag(\frac{1}{k})\ S^T \tag{4.10}$$

Where $k = diag(\sigma_1,...,\sigma_p)$, by replacing Φ in equation (5.6), using equation (5.7); the weights vector (5.8):

$$\vec{w} = \left[H\ diag(\frac{1}{k})\ S^T \right] \vec{y} \tag{4.11}$$

SVD can solve any equation system. In the proposed case SVD effect in reducing the of

the output error, it can also be used to remove any RBFNN when its associated singular value had a small value or if the approximation error can't affect the result.

CHAPTER 5

EXPERIMENT AND RESULT

5. Experiments and Results

In this chapter, we show the results of our experiments as well as all the necessary requirements that conduct our experiments.

5.1 Experiment Procedure

Our proposed approaches (HRBFN-PSO and HC-RBFN-PSO) are experimented using MATLAB 2012 under Windows 7 with core i5-3210M CPU 2.5GHz, 4GB RAM memory. MATLAB is a high-performance language for technical computing developed by mathworks [113]. It combines between programming, data analysis, visualization, and computation. The environment of Matlab is easy to use and tests algorithms immediately without recompilation and plotting of functions and data. The basic data item in Matlab is a matrix. Matlab is built in graphics tool and provides Graphical User Interface (GUI) to interpret data easier in models and curves. On the other hand, Matlab uses a large amount of memory that consumes the CPU time; therefore in real-time applications, it is very complicated.

5.2 Function Approximation and Time Series

In this section, we talk about nonlinear benchmarks functions that used in our experiments and Parameters used in PSO algorithm for HRBFN-PSO approach. Then we show the results of the experiments compared with our approaches.

5.2.1 Benchmarks and Parameters

To compare the performance in function approximation of our HRBFN-PSO approach, we conduct multiple experiments on the training data by using three nonlinear function examples as shown in table 5.1 and to compare with other previous works that used the same nonlinear function. All these benchmark functions reflect varying degrees of complexity. In all nonlinear function, we conducted several experiments using a

different number of hidden neurons until reaching the same number of hidden neurons used in other approaches.

Table 5.1: The Benchmark Nonlinear Function Information

No	Nonlinear Functions	# Data Point	Range
1	$y = \frac{\sin(x)}{x}$	50	x ϵ [-10,10]
2	$y = -sin\left(\frac{\pi x}{10}\right) + cos\left(\frac{2\pi x}{5}\right)$	50	x ϵ [-10,10]
3	$y = 1.1\left(1 - x + 2x^2\right)\exp\left(-\frac{x^2}{2}\right)$	100	x ϵ [-4,4]

For each of the benchmark functions, HRBFN-PSO optimizes the parameters of the RBFNN. These parameters are optimized by using PSO and traditional algorithms.

The parameters of the PSO algorithm that are used in this experiment were set as inertia weight ω is calculated based on the equation 4.4. The rest of parameters for PSO algorithm are shown in table 5.2. To choose the best values of PSO parameters, we conduct multiple experiments using different values of PSO parameters, until we got these values of parameters shown in table 5.2. Figure 5.1 demonstrates the fitness function (MSE) behavior of iteration number for PSO to select the best number of iterations. A number of particles have been tested to select the best number of particles. Figure 5.2 shows the relation between the number of particles and RMSE values. We have selected 20 particles in our experiments because it gives good results in a short time.

Table 5.2: Parameters for PSO used in HRBFN-PSO

Parameter	Value
Number of Particles	20
Number of Iterations	500
c_1	0.5
c_2	2
Vmax	1
ωmax	1.2
ωmin	0.8
r_1	Randomly (0,1)
r_2	Randomly (0,1)

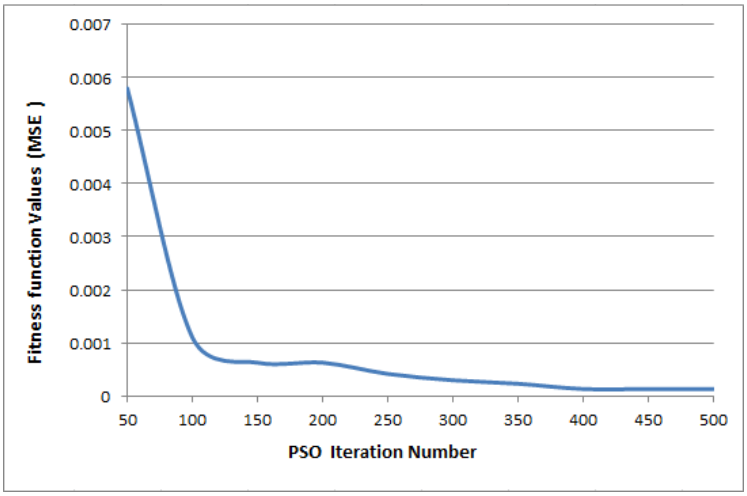

Figure 5.1: Fitness function (MSE) behavior of iteration number for PSO

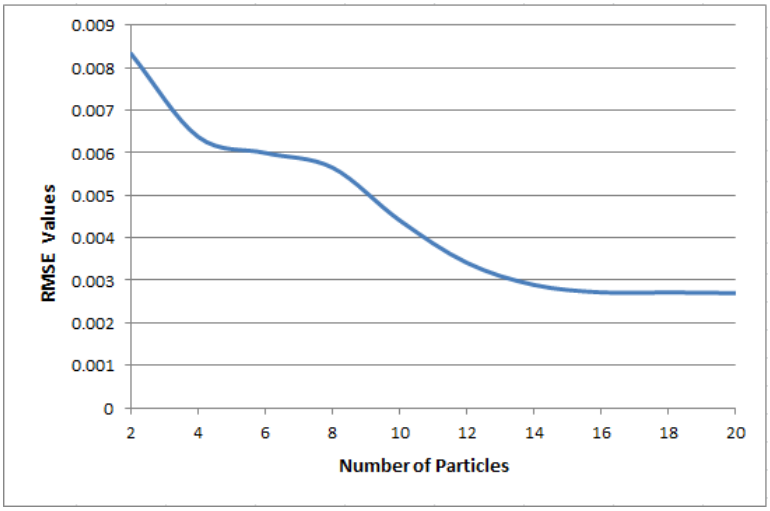

Figure 5.2: Relation between number of particles and RMSE values

5.2.2 First Benchmark Example

In this function, we conduct our experiments on 50 training data points as shown in table 5.1. Table 5.3 shows the RMSE results of our approach compared with Sun [114] approach. Hence, figure 5.3 comes to show the curve behavior of RMSE for first benchmark function. This curve demonstrates how the RMSE is reduced dramatically with small increase number of hidden neurons. Figure 5.4 shows the curve results between target data and real output data in HRBFN-PSO approach for first benchmark function. Moreover, table 5.3 illustrates that our approach introduces better results compared with other approaches in terms of reducing the number of hidden neurons and enhance RMSE value. The ability to achieve optimal values by using the proposed approach is increased and the approximation error decreases with less number of hidden neurons.

Table 5.3: Results for Nonlinear first function

Approach	# Hidden Neuron (RBFNN)	RMSE
Sun [114]	28	0.0035
Proposed Approach HRBFN-PSO	2	0.147143
	4	0.063337
	6	0.024315
	8	0.020456
	10	0.014585
	12	0.008749
	14	0.003839
	16	0.002299
	18	0.000499
	20	0.000291
	22	0.000230
	24	0.000198
	26	0.000175
	28	0.000034

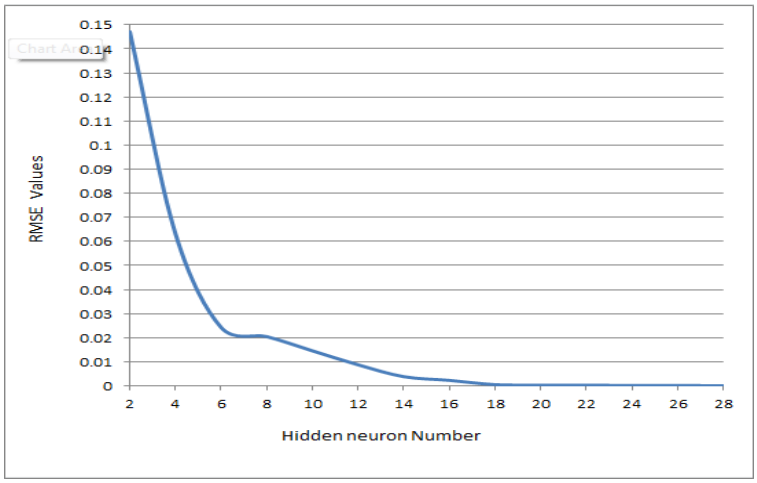

Figure 5.3: Curve behavior of RMSE for first benchmark

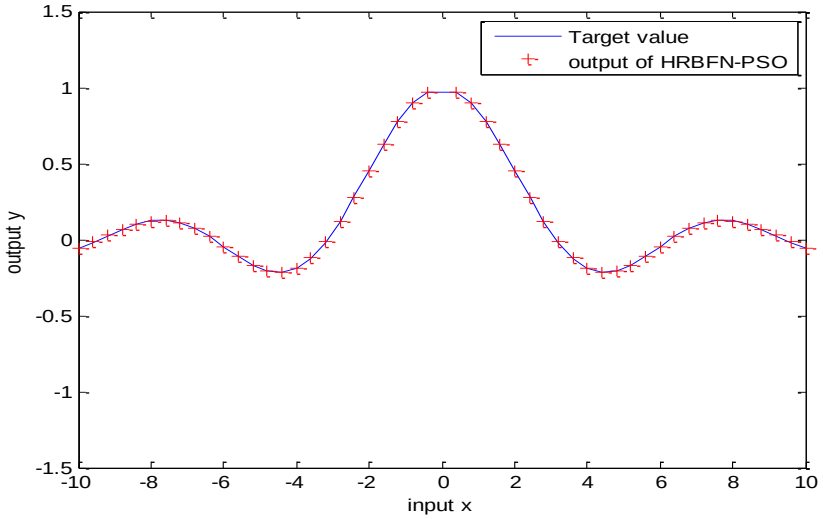

Figure 5.4: Target and real output (using 28 neurons)

For the first benchmark $\left(y = \frac{\sin(x)}{x} \right)$

5.2.3 Second Benchmark Example

In this function, we conduct our experiments on 50 training data points as shown in table 5.1. Table 5.4 shows the RMSE results of this approach compared with Sun [114] approach. Figure 5.5 comes to show the curve behavior of RMSE of the second function. This curve demonstrates how the error is reduced dramatically with small increase number of hidden neurons. Figure 5.6 shows that the curve results between target data and real output data in HRBFN-PSO approach for second benchmark function. Through table 5.4, we noticed that our approach comes with better results

compared with other approaches in terms of reducing the number of hidden neurons and enhance RMSE value. This experimental nonlinear function has been selected to demonstrate the ability of HRBFN-PSO to approximate more difficult function and very variable output function.

Table 5.4: Results for Nonlinear second function

Approach	# Hidden Neuron (RBFNN)	RMSE
Sun [114]	29	0.0079
Proposed Approach HRBFN-PSO	2	0.212729
	4	0.148829
	6	0.053260
	8	0.046253
	10	0.034498
	12	0.020638
	14	0.019029
	16	0.014451
	18	0.012506
	20	0.008818
	22	0.007013
	24	0.005543
	26	0.003295
	28	0.002629
	29	0.001403

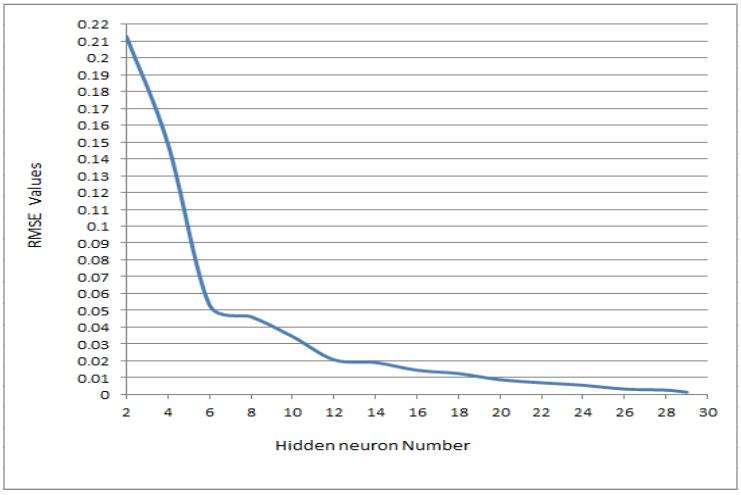

Figure 5.5: Curve behavior of RMSE for second benchmark

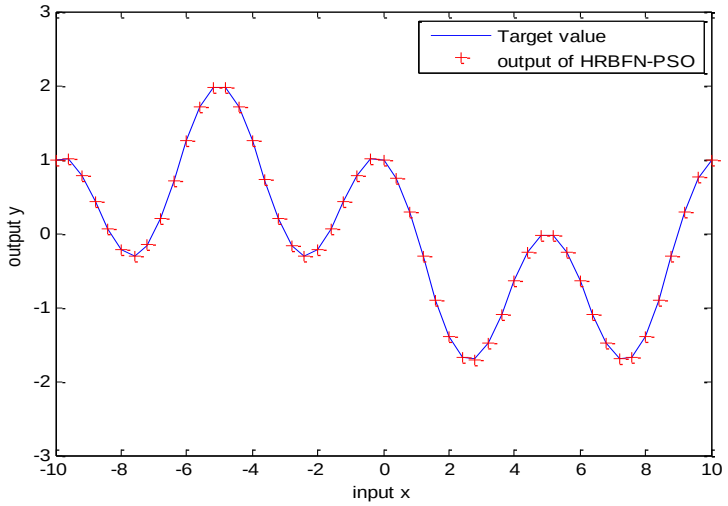

Figure 5.6: Target and real output (using 29 neurons)

For the second benchmark $\left(y = -sin\left(\frac{\pi x}{10}\right) + cos\left(\frac{2\pi x}{5}\right)\right)$

5.2.4 Third Benchmark Example

In this function, the experiments are conducted on 100 training data points as shown in table 5.1. Table 5.5 shows the SSE results for our approach compared with Liu [115] approach. Figure 5.7 show the curve behavior of SSE for third function. This curve demonstrates how the error is reduced dramatically with small increase number of hidden neurons. Figure 5.8 shows the curve results between target data and real output data in HRBFN-PSO approach for third benchmark function. In Table 5.5, we noticed that our approach introduces better results compared with other approaches in terms of reducing the number of hidden neurons and enhance SSE value.

Table 5.5: Results for Nonlinear third function

Approach	# Hidden Neuron (RBFNN)	SSE
Liu [114]	10	0.8756
Proposed Approach HRBFN-PSO	2	1.074451
	4	0.328720
	6	0.016419
	8	0.007835
	10	0.003806

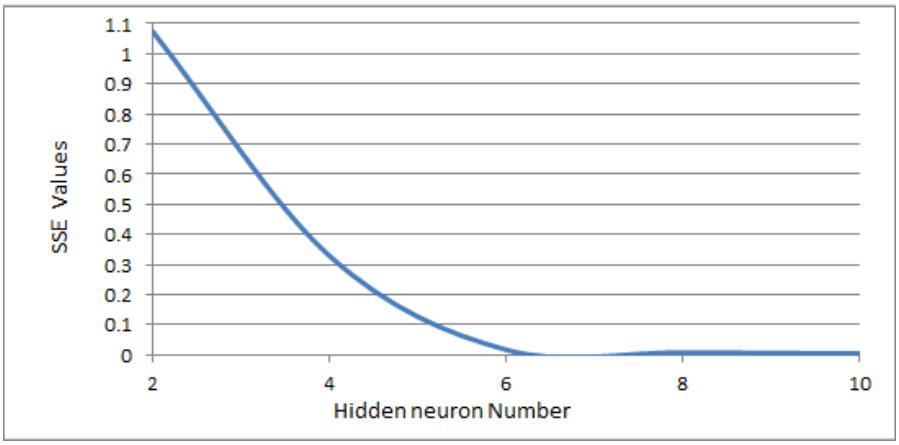

Figure 5.7: Curve behavior of SSE for third benchmark

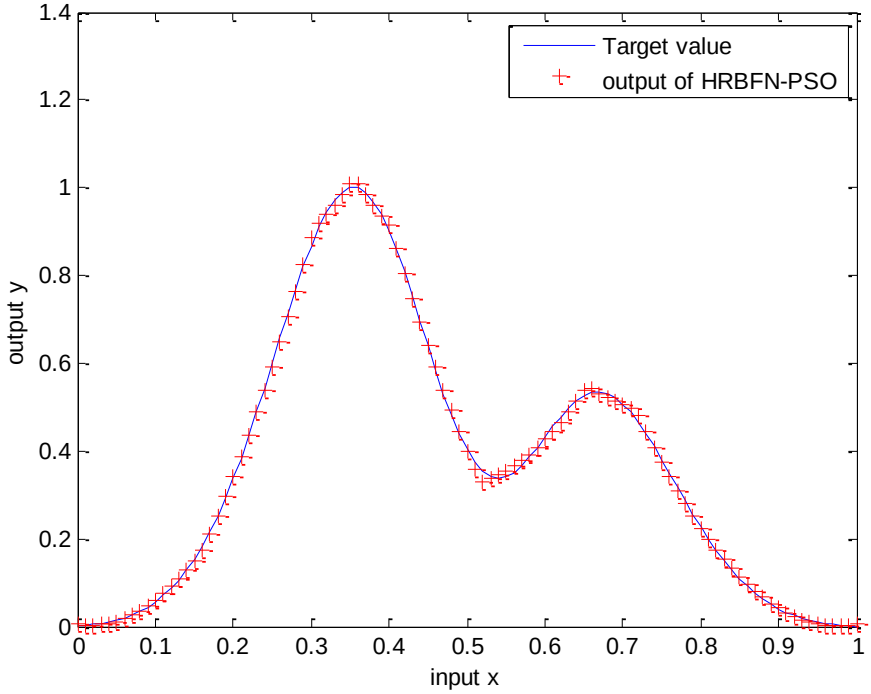

Figure 5.8: Target and real output (using 10 neurons)

For the third benchmark $\left(y = 1.1\left(1 - x + 2x^2\right)\exp\left(-\frac{x^2}{2}\right)\right)$

5.2.5 Mackey-Glass Time Series

Mackey-Glass is an artificial chaotic time series [116], it is a well-known benchmark widely used in time series prediction field. The Mackey-Glass equation is a nonlinear time delay equation that generates the following equation:

$$\frac{dx(t)}{dt} = \alpha . \frac{x(t-\tau)}{1+(x(t-\tau))^n} - \beta x(t) \qquad (5.1)$$

Where *x(t)* is the value of the time series at time t, α , β , τ , and *n* are real numbers.

Mackey-Glass time series prediction is used to prove the prediction efficiency of the proposed approach (HRBFN-PSO). In this work, initial parameters are used in our test of Mackey-Glass benchmark [117, 118] are set as follows:

x(0)=1.2, x(t) = 0 when t<0, $\alpha = 0.2$, $\beta = 0.1$ and $\tau = 17$.

Figure 5.9 shows the curve of Mackey-Glass time series using 1200 data samples.

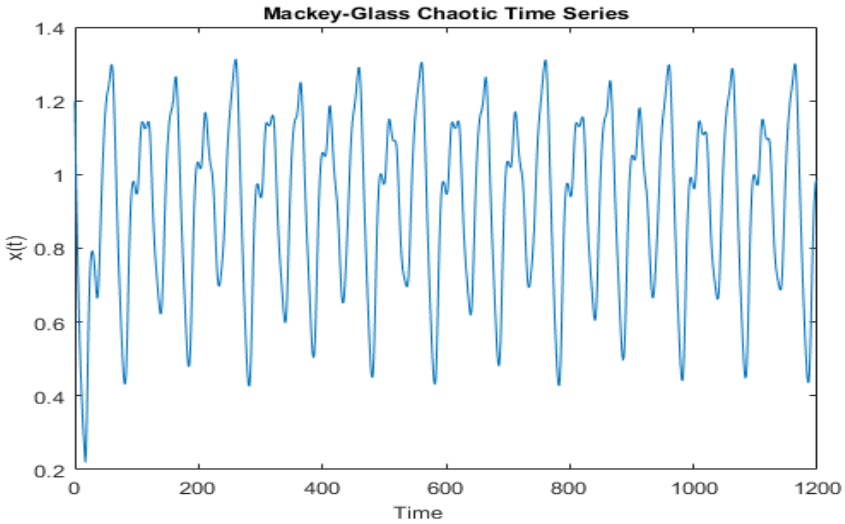

Figure 5.9**:** Mackey- Glass time series (1200 samples)

In Mackey-Glass function, we conducted our experiments on 600 data samples training. Table 5.6 shows the RMSE results for our approach. Figure 5.10 shows the curve behavior of RMSE for Mackey-Glass function. This curve demonstrates how the RMSE is reduced dramatically with an increased number of hidden neurons. Figure 5.11 shows the curve results between target data and real output data in HRBFN-PSO approach for Mackey-Glass time series.

Table 5.6: Results for Mackey-Glass time series

Approach	# Hidden Neuron (RBFNN)	RMSE
Proposed Approach HRBFN-PSO	10	0.177812
	20	0.102181
	30	0.065767
	40	0.054361
	50	0.042056
	60	0.033026
	70	0.027997
	80	0.025698
	90	0.018152
	100	0.013403

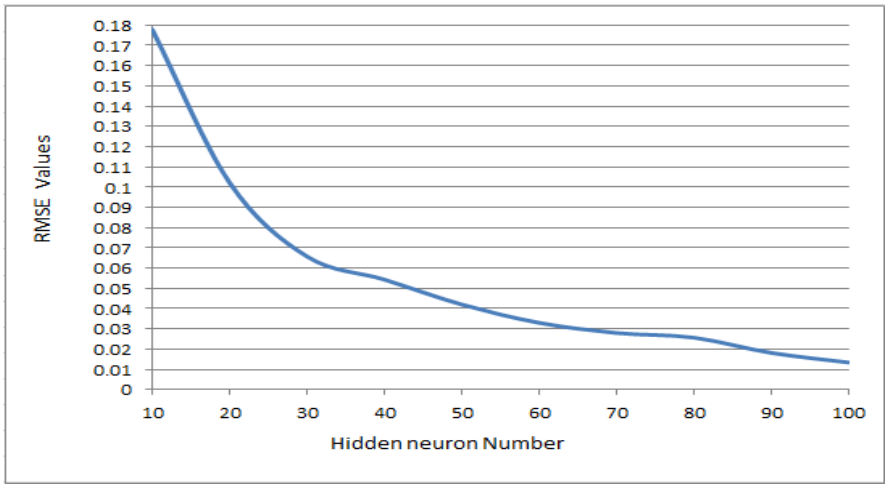

Figure 5.10: Curve behavior of RMSE for Mackey- Glass time series

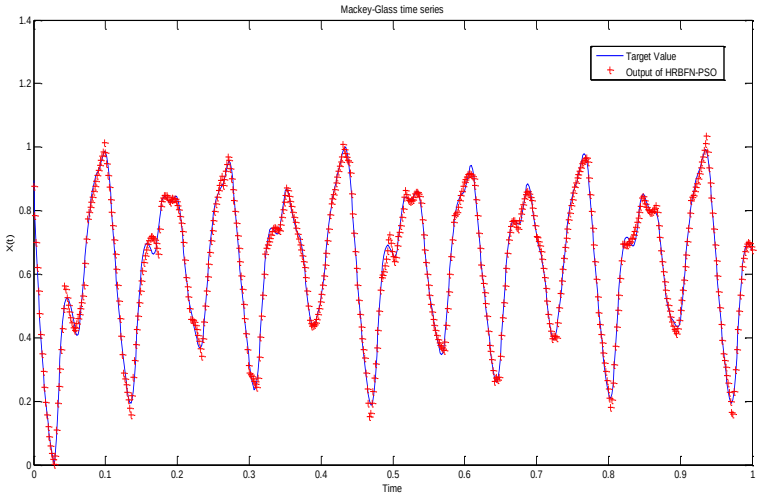

Figure 5.11: Mackey-Glass time series result (600 samples)

5.3 Classification

In this section, we talked about datasets and Parameters that are used in PSO algorithm for HC-RBFN-PSO approach. Then we introduced the results of the experiments for each dataset.

5.3.1 Datasets and Parameters

In HC-RBFN-PSO approach, pre-conducting experiments on spam classification, we conducted experiments on two datasets namely Wisconsin diagnostic breast cancer (wdbc) and Ionosphere. These two datasets are obtained from UCI repository site; each column contains attributes value for each item. In each record, the data is delimited by commas. We used these datasets after the process of converting to CSV (Comma Separated Values) file compatible with Matlab environment. We split the data into two phases, training set (70% data) and testing set (30% data). The data is selected randomly from the datasets for training and testing sets. We used classification accuracy measure [119] to find the performance of datasets. Classification Accuracy is the proportion of instances which are correctly classified, as illustrated in equation 4.6.

The parameters of the PSO algorithm that are used in HC-RBFN-PSO approach were set as inertia weight ω in calculated based on the equation 4.4. The rest of parameters are illustrated in table 5.7. To choose the best values of PSO parameters, we conducted multiple experiments for PSO until we achieved these values of parameters which are illustrated in table 5.7. Figure 5.12 illustrates the fitness function (RMSE) behavior of iteration number for PSO to select the best number of iterations. The number of particles is determined in accordance with the number of the hidden neurons in RBFNN.

Table 5.7: Parameters for PSO used in HC-RBFN-PSO

Parameter	Value
Number of Particles	Number of hidden neurons
Number of Iterations	100
c1	1.4
c2	1.4
Vmax	1
ωmax	0.9
ωmin	0.4
r_1	Randomly (0,1)
r_2	Randomly (0,1)

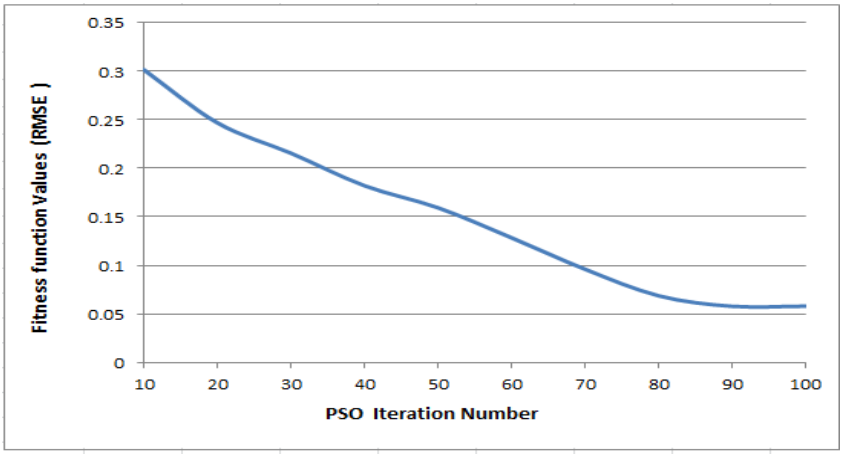

Figure 5.12: Fitness function (RMSE) behavior of iteration number for PSO

5.3.2 WDBC dataset

The Wisconsin Diagnostic Breast Cancer (WDBC) dataset was created by Wolberg, Street and Olvi. This dataset is downloaded from UCI Machine Learning Repository site [120]. This dataset consists of data from 569 breast FNA cases containing 30 attributes and one binary classification variable (benign or malignant). This attributes are obtained by semi-automated image analysis applied to digital photomicrographs obtained from the FNA slides. WDBC dataset includes 357 items of benign breast and 212 items of malignant breast cancer. There are no missing attribute values in this

dataset. Table 5.8 shows the results of our experiments on WDBC dataset; we presented the results for training and testing phases. The experiments are conducted repeatedly by using a different number of hidden neurons in RBFNN as shown in table 5.8. We recorded the results when using 5, 10, 15, and 20 hidden neurons for each class.

Table 5.8: Experiment result of WDBC dataset

# Hidden Neuron	Accuracy	
5	Training phase	98 %
	Testing phase	95.9 %
10	Training phase	98.5 %
	Testing phase	95.9 %
15	Training phase	98.8 %
	Testing phase	96.2 %
20	Training phase	99 %
	Testing phase	96.3 %

5.3.3 Ionosphere dataset

This dataset is collected by a radar system which consists of phased array of 16 high-frequency antennas with a total transmitted power in the order of 6.4 kilowatts. This dataset is downloaded from UCI Machine Learning Repository site [120]. This dataset contains 351 instances and 34 a numeric attributes. There is one attribute that all of its values are zeros, which is discarded in our experiments. So the total number of attributes is 33. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some types of structure in the ionosphere. "Bad" returns are those that do not.

Table 5.9 shows the results of our experiments on Ionosphere dataset; we present the results for training and testing phases. The experiments are also conducted repeatedly using a different number of hidden neurons in RBFNN as shown in table 5.9. We recorded the results when using 5, 10, 15, and 20 hidden neurons for each class.

Table 5.9: Experiment result of Ionosphere dataset

# Hidden Neuron	Accuracy	
5	Training phase	92.7 %
	Testing phase	88.7 %
10	Training phase	94.3 %
	Testing phase	89.6 %
15	Training phase	96.7 %
	Testing phase	90 %
20	Training phase	98.8 %
	Testing phase	90.5 %

5.4 Spam Classification

5.4.1 Dataset Spam

In HC-RBFN-PSO approach, we used SPAMBASE dataset to classify email as spam or non-spam. It is downloaded from UCI Machine Learning Repository site [120]. SPAMBASE was proposed by Mark Hopkins and in his colleagues. SPAMBASE dataset is a multivariate dataset contains data from a single email account. SPAMBASE contains 4601 records that are identified previously, 1813 classified as non-spam (39.4%) and 2788 classified as spam (60.6%). SPAMBASE is containing fifty-seven data attributes and one classification attribute to determine the type email (value 0 for non-spam and value 1 for spam). Most of the attributes (1-54) express particular words or characters were frequently in an email. The attributes (55-57) measure the length of sequences of consecutive capital letters.

Here we introduced definitions of the SPAMBASE attributes:

- Attributes (1-48) 48 real attributes of type word_freq_WORD = percentage of words in the e-mail that match WORD.
- Attributes (49-54) 6 real attributes of type char_freq_CHAR = percentage of characters in the e-mail that match CHAR.

- Attribute (55) 1 real attribute of type capital_run_length_average = average length of uninterrupted sequences of capital letters.
- Attribute (56) 1 integer attribute of type capital_run_length_longest = length of longest uninterrupted sequence of capital letters.
- Attribute (57) 1 integer attribute of type capital_run_length_total = sum of the length of uninterrupted sequences of capital letters = a total number of capital letters in the e-mail.

5.4.2 Preprocessing

The available data in the SPAMBASE dataset is in a numeric form. All the fifty-seven attributes in the SPAMBASE dataset are mostly represent frequencies of various words and characters of the emails. We wish to normalize this data before conducting experiments on HC-RBFN-PSO approach. Normalization is one of the basic methods for ANN learning technique, due to the data normally is not equally distributed and the large values become dominant for fewer values during neural network learning, so we need to distribute data within a specific range.

Normalization processing is an important stage due to speeding up model, convergence and reducing the effect of imbalance in data to the classifier. In the training and testing phases for these experiments, normalize the data in the range (0, 1) using equation 5.2.

$$Z = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \qquad (5.2)$$

Where Z is the new value after normalization, X_i is the old value, X_{min} is the minimum value for all data, and X_{max} is the maximum value for all data.

5.4.3 Results

In the file SPAMBASE.data from UCI repository site, each column contains attribute value for each email, in each record, the data is delimited by commas. In this work, we

used the SPAMBASE dataset after converted to CSV (Comma Separated Values) file compatible with Matlab environment. We split the data into two phases, training set (70% data) and testing set (30% data). The performance is measured by evaluating the accuracy for each phase. We calculated classification accuracy based on confusion matrix. The confusion matrix is commonly used for visualizing the performance of classification algorithms. Confusion matrix for an email classification can be illustrated as in table 5.10 as follows:

Table 5.10: Confusion matrix component

	Spam	Non-spam
Spam	TP	FN
Non-spam	FP	TN

TP (True Positives): Number of spam messages that are correctly classified.

FP (False Positives): Number of non-spam messages that are incorrectly classified.

FN (False Negative): Number of spam messages that are incorrectly classified.

TN (True Negatives): Number of non-spam messages that are correctly classified.

Cross validation is a technique used to examine the neural network performance and to determine the robustness neural classifiers with respect to sampling variation [121]. One simple idea used for cross validation is randomly splitting data into two subsamples of training and testing sets, which used to choose data in our experiments. In cross validation technique, the RBFNN parameter values obtained from training data will be applied in the test data. At the beginning of these experiments, a twofold cross validation is used and split the total samples data into two sets. In the first fold, we split the total samples data into 70% of training and the remaining samples 30% of testing, while in the second fold, we split the total samples data into 30% of testing and the remaining samples 70% of training phase. The results of cross validation (using 80

hidden neurons) for first fold are 94.7% of training phase and 86.5% of testing phase, the results of cross validation (using 80 hidden neurons) for second fold are 92.6% of training phase and 89.8% of testing phase. We note from cross validation results that there is a convergence in the results of two folds for cross validation, which indicates that the proposed approach can be used in any classification data.

Now in our experiments, the data of the samples is selected randomly from the SPAMBASE dataset for training and testing sets, to eliminate any data particular behavior for experiment results. In our approach HC-RBFN-PSO, we used the most common approach to find the performance of spam filtering that is the classification accuracy [119]. Classification Accuracy is the proportion of instances which are correctly classified. We compute Classification Accuracy as illustrated in equation 4.6.

To compare the classification accuracy of our approach, we conduct multiple experiments for our approach on the training data and test data number of times using different number of hidden neurons in RBFNN. After that, we compared our results with other previous works as shown in table 5.13.

In the training and testing phase, we show the results using 80 hidden neurons and represent the results using confusion matrix. Table 5.11shows the confusion matrix for training phase, and also it shows the results for testing Phase using confusion matrix as shown in table 5.12.

Table 5.11: Confusion matrix for training phase

Number of hidden neuron = 80		
	Spam	Non-spam
Spam	1890	65
Non-spam	137	1130

Table 5.12: Confusion matrix for testing phase

Number of hidden neuron = 80		
	Spam	Non-spam
Spam	805	35
Non-spam	64	475

Now, we present the results for training and testing phases for HC-RBFN-PSO approach. The experiments are conducted in a number of times by using a different number of hidden neurons in RBFNN hidden layer as shown in table 5.13. We recorded the results when using 10, 20, 30, 40, 50, 60, 70, and 80 hidden neurons for each class.

Table 5.13: Experiment result and comparison of spam email classification

Approach	# Hidden Neuron	Accuracy	
COSTA [122] – RBF	---	84.3 %	
SHARMA [123]– MLP	---	93.28 %	
JUKIC [124] – ANN	30	91 %	
FARIS [125] – MLP	---	91.85 %	
Proposed Approach HC-RBFN-PSO	10	Training phase	89.5 %
		Testing phase	88.5 %
	20	Training phase	91 %
		Testing phase	90.1 %
	30	Training phase	91.8 %
		Testing phase	90.9 %
	40	Training phase	92.5 %
		Testing phase	91.4 %
	50	Training phase	93.1 %
		Testing phase	91.6 %
	60	Training phase	93.3 %
		Testing phase	92.1 %
	70	Training phase	93.5 %
		Testing phase	92.5 %
	80	Training phase	93.7 %
		Testing phase	92.8 %

In accordance with the obtained results, we concluded that the HC-RBFN-PSO approach has better results in the classification accuracy as shown the table 5.13. Through the results, we concluded that the proposed approach used a little number of hidden neurons; therefore the computational cost is decreased.

5.5 Discussion

In this section, we discussed the results of the two proposed approaches HRBFN-PSO and HC-RBFN-PSO. We summarized the results obtained from the function approximation experiments in table 5.14. Here in HRBFN-PSO, the experiments are conducted on three Benchmark Nonlinear function as illustrated in table 5.1. We noted from table 5.14 that the error values (RMSE and SSE) of HRBFN-PSO approach are good values using the same number of neurons in other approaches (see Tables 5.3, 5.4 and 5.5). From the figures 5.2, 5.4 and 5.6, we noted that the squared error between the real output and the target of each benchmark is minimized.

Table 5.14: A summary of the results for function approximation

Benchmark Function	# Hidden Neuron (RBFNN)	Error function	Error value
First benchmark function	28	RMSE	0.000034
Second benchmark function	29	RMSE	0.001403
Third benchmark function	10	SSE	0.003806

It is clear that the proposed approach converges to the optimum value with less number of hidden neurons in RBFNN with better approximation error; take in account that the number of particles in PSO algorithm is Median and small number of iteration. Therefore the HRBFN-PSO approach is improved approximation accuracy than other approaches.

To prove the strength efficiency of HRBFN-PSO approach, we conducted another experiment that is applying complex well-known benchmark Mackey-Glass time series. Apply this function a number of times by using a different number of hidden neurons. The experiment results for Mackey-Glass are shown in table 5.6 which shows that the RMSE value (using 100 hidden neurons) is 0. 013403. Figure 5.9 illustrates the applicability of the real output with the target value. Through the aforementioned results, we conclude that the HRBFN-PSO proposed approach is suitable the problems prediction.

To test HC-RBFN-PSO approach for classification, we conduct experiments on two datasets namely Wisconsin diagnostic breast cancer (wdbc) and Ionosphere, and then record the obtained results from experiments. Through previous results (table 5.8 and table 5.9) we concluded that the HC-RBFN-PSO approach is suitable for classification data. After that we conducted experiment on big SPAMBASE dataset to classify email as spam or non-spam. To ensure the distribution of data within a specific range, we made normalization of data before conduct experiments. We divided the data into two phases training and testing, as for the data are selected randomly from SPAMBASE dataset. We conducted experiment for SPAMBASE dataset eight times with different number of hidden neurons. To show the result of spam filtering, we used commonly visualization method that is a confusion matrix. Tables 5.11 and 5.12 show the confusion matrix results (using 80 neurons), for training and testing phases respectively. Table 5.13 shows the results for all experiments compare with multiple previous studies that use the same dataset, and we used classification accuracy measure to assess the results. From table 5.13, we noted that the most previous studies compute the classification accuracy without determines the number of hidden neurons but in our

approach for each experiment determines the number of hidden neurons used and the classification accuracy. The classification accuracy using 80 neurons is 93.7% for training phase and 92.8 for the testing phase. Figure 5.13 show the graph of accuracy (using 10, 20, 30, 40, 50, 60, 70, 80 neurons), from this graph we noted how to increase accuracy gradually.

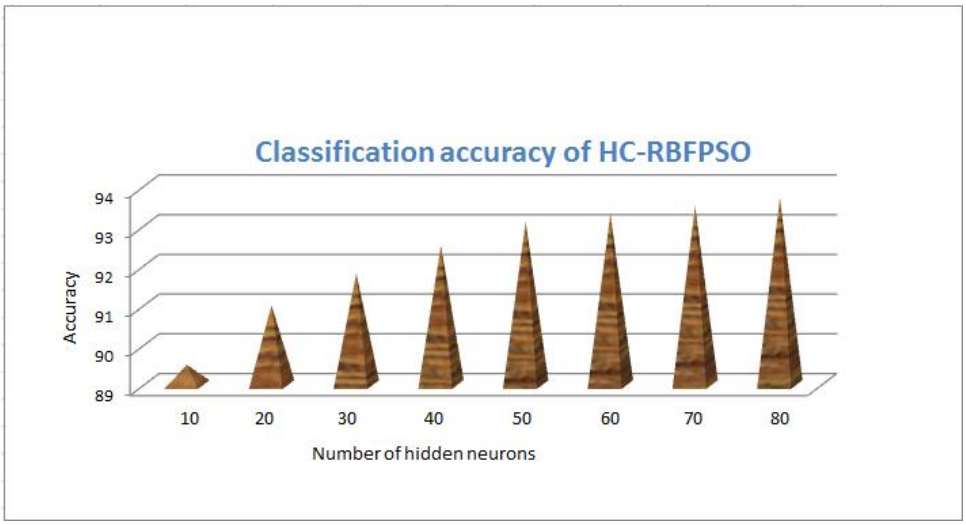

Figure 5.13: Graph accuracy for HC-RBFN-PSO approach

We note that from experiment results for our approach be effective in spam filtering messages using a small number of neurons in a large dataset.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6. Conclusion and Future Work

Electronic mail service has become an essential part in the means of communications sector in which depends on it the economy of the countries and companies due to their usefulness and benefits. The abuses of the email service refer to the presence of SPAM in the user mailboxes. These messages consume time to be eliminated. This thesis shows the development of a method to decrease the SPAM problem. This work also meets the objectives of this master thesis; we proposed an enhanced method of combining RBFNN and PSO algorithm to filters the email spam. However, the proposed method proves the ability to approximate functions from unknown input-output data.

Function approximation is a prediction process used to select function that is approximated to target function. ANN is one of the function approximation tools have the ability to find input-output pattern data. One type of neural networks of the problem of function approximation is RBFNN which is used in our experiment. It is characterized by the following features: better approximation, simpler network structures, and faster learning algorithms. Time series prediction is a process of taking a set of current data that is used to predict future data. The main goal of time series model is to build a model used current data to conclude future data.

In this thesis, we proposed two-hybrid approaches (HRBFN-PSO and HC-RBFN-PSO) that combining RBFNN and PSO used for function approximation problem, time series prediction, and classification problem. Here, we used PSO to find optimal values for the centers of hidden neurons in RBFNN; the rest of RBFNN parameters use the traditional KNN algorithm to optimize the radii and SVD technique to optimize the weight.

The proposed HRBFN-PSO algorithm has been evaluated using three nonlinear

mathematical functions and Mackey-Glass time series. This is tested on the specific training data. The results obtained are comparable with other approaches shows HRBFN-PSO algorithm with more improved approximation accuracy, and also reducing the Root Mean Square Error (RMSE) and Sum Square Error (SSE) compared with other approaches. The results of the simulations show that HRBFN-PSO is an effective method that is a reliable alternative for approximation nonlinear mathematical functions and time series prediction. The quality of the results improves the convergence. This enhanced algorithm approximate with the best error (RMSE) is 0.000034 according to our experiments.

The proposed HC-RBFN-PSO approach is used for classification data. In this thesis, HC-RBFN-PSO used in email classification, firstly we conducted experiments using two datasets are Wisconsin diagnostic breast cancer (wdbc) and Ionosphere, then using large spam dataset that is SPAMBASE, which presents a collection of spam and non-spam emails contains 4601 records with 57 attributes. The results obtained from the experiments on SPAMBASE dataset are comparable with other approaches that use the same dataset. Our approach has given us better performance in terms of classification accuracy. The results of the simulations show that HC-RBFN-PSO is an effective method for classification and improves the convergence. This enhanced algorithm classified email spam with accuracy 93.7% using 80 hidden neurons according to our experiments.

For future work, we suggest to compare our approaches with other Bioinspired Optimization Algorithms that is a Genetic Algorithm (GA) combined with RBFNN. Additionally, we propose method that detects spam messages. To make our approach more effective, it is also possible to develop it to detect image spam messages. Other

idea that we can use in the future work is the feature selection methodology to decrease the number of features in the input data.

References

[1] YANG, Sibo, et al. Investigation of neural networks for function approximation. Procedia Computer Science, 2013, 17: 586-594.

[2] MORDOHAI, Philippos; MEDIONI, Gérard. Dimensionality estimation, manifold learning and function approximation using tensor voting. Journal of Machine Learning Research, 2010, 11.Jan: 411-450.

[3] BUSONIU, Lucian, et al. Reinforcement learning and dynamic programming using function approximators. CRC press, 2010.

[4] XU, Xin; ZUO, Lei; HUANG, Zhenhua. Reinforcement learning algorithms with function approximation: Recent advances and applications. Information Sciences, 2014, 261: 1-31.

[5] VUKOVIĆ, Najdan; MILJKOVIĆ, Zoran. A growing and pruning sequential learning algorithm of hyper basis function neural network for function approximation. Neural Networks, 2013, 46: 210-226.

[6] PARK, Junheung; KIM, Kyoung-Yun. Instance variant nearest neighbor using particle swarm optimization for function approximation. Applied Soft Computing, 2016, 40: 331-341.

[7] BAUDAT, Gaston; ANOUAR, Fatiha. Kernel-based methods and function approximation. In: Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on. IEEE, 2001. p. 1244-1249.

[8] MOHAMMED AWAD. Forecasting of Chaotic Time Series Using RBF Neural Networks Optimized By Genetic Algorithms. The International Arab Journal of Information Technology. Volume *(15), No. (1), January 2018.

[9] MAKRIDAKIS, Spyros; WHEELWRIGHT, Steven C.; HYNDMAN, Rob J. Forecasting methods and applications. John Wiley & Sons, 2008.

[10] DAS, Atin; DAS, Pritha. Chaotic analysis of the foreign exchange rates. Applied mathematics and computation, 2007, 185.1: 388-396.

[11] LORENZ, Edward N. The essence of chaos. University of Washington Press, 1995.

[12] GHOLIPOUR, Ali; ARAABI, Babak N.; LUCAS, Caro. Predicting chaotic time series using neural and neurofuzzy models: a comparative study. neural processing letters, 2006, 24.3: 217-239.

[13] KENNEL, Matthew B.; ISABELLE, Steven. Method to distinguish possible chaos from colored noise and to determine embedding parameters. Physical Review A, 1992, 46.6: 3111.

[14] ZHANG, Guoqiang; PATUWO, B. Eddy; HU, Michael Y. Forecasting with artificial neural networks: The state of the art. International journal of forecasting, 1998, 14.1: 35-62.

[15] HEARST, Marti A. Text data mining: Issues, techniques, and the relationship to information access. In: Presentation notes for UW/MS workshop on data mining. 1997. p. 112-117.

[16] FELDMAN, Ronen; DAGAN, Ido. Knowledge Discovery in Textual Databases (KDT). In: KDD. 1995. p. 112-117.

[17] FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From data mining to knowledge discovery in databases. AI magazine, 1996, 17.3: 37.

[18] SIMOUDIS, Evangelos. Reality check for data mining. IEEE Expert: Intelligent systems and their applications, 1996, 11.5: 26-33.

[19] DE MORAIS FONSECA, Osvaldo Henriques, et al. Measuring, Characterizing, and Avoiding Spam Traffic Costs. IEEE Internet Computing: Computer Society, 2016, 99.

[20] Email Statistical Report, http://www.radicati.com/wp/wp-content/uploads/2010/04/Email-Statistics-Report-2010-2014-Executive-Summary2.pdf, 15/5/2016.

[21] GAIKWAD, Bhagyashri U.; HALKARNIKAR, P. P. Random Forest Technique for E-mail Classification. International Journal of Scientific & Engineering Research, 2014, 5.3: 145-153.

[22] GUDKOVA, Darya, et al. Kaspersky security bulletin. Spam Evolution, 2013.

[23] BAHGAT, Eman M.; RADY, Sherine; GAD, Walaa. An e-mail filtering approach using classification techniques. In: The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015, Beni Suef, Egypt. Springer International Publishing, 2016. p. 321-331.

[24] BOUGUILA, Nizar; AMAYRI, Ola. A discrete mixture-based kernel for SVMs: application to spam and image categorization. Information Processing & Management, 2009, 45.6: 631-642.

[25] CAO, Yukun; LIAO, Xiaofeng; LI, Yunfeng. An e-mail filtering approach using neural network. In: International Symposium on Neural Networks. Springer Berlin Heidelberg, 2004. p. 688-694.

[26] FDEZ-RIVEROLA, Florentino, et al. SpamHunting: An instance-based reasoning system for spam labelling and filtering. Decision Support Systems, 2007, 43.3: 722-736.

[27] STUART, Ian; CHA, Sung-Hyuk; TAPPERT, Charles. A neural network classifier for junk e-mail. In: International Workshop on Document Analysis Systems. Springer Berlin Heidelberg, 2004. p. 442-450.

[28] HAN, Jiawei; PEI, Jian; KAMBER, Micheline. Data mining: concepts and techniques. Elsevier, 2011.

[29] QUINN, Alexander J.; BEDERSON, Benjamin B. Human computation: a survey and taxonomy of a growing field. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, 2011. p. 1403-1412.

[30] MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 1943, 5.4: 115-133.

[31] HEBB, Donald Olding. The organization of behavior: A neuropsychological theory. New York: Science, 1949.

[32] ZURADA, Jacek M. Introduction to artificial neural systems. St. Paul: West, 1992.

[33] MITCHELL, Tom M. Machine learning. 1997. Burr Ridge, IL: McGraw Hill, 1997, 45: 37.

[34] KOHONEN, Teuvo; SOMERVUO, Panu. Self-organizing maps of symbol strings. Neurocomputing, 1998, 21.1: 19-30.

[35] WAN, Eric A. Neural network classification: a Bayesian interpretation. IEEE transactions on neural networks/a publication of the IEEE Neural Networks Council, 1990, 1.4: 303.

[36] HORNIK, Kurt; STINCHCOMBE, Maxwell; WHITE, Halbert. Multilayer feedforward networks are universal approximators. Neural networks, 1989, 2.5: 359-366.

[37] GE, Shuzhi Sam, et al. Stable adaptive neural network control. Springer Science & Business Media, 2013.

[38] RIPLEY, Brian D. Pattern recognition and neural networks. Cambridge university press, 2007.

[39] YAO, Xin. Evolving artificial neural networks. Proceedings of the IEEE, 1999, 87.9: 1423-1447.

[40] HAYKIN, Simon; LIPPMANN, Richard. Neural networks, a comprehensive foundation. International journal of neural systems, 1994, 5.4: 363-364.

[41] GROTHMANN, Ralph. Multi-agent market modeling based on neural networks. Faculty of Economics, University of Bremen, 2002.

[42] AGATONOVIC-KUSTRIN, S.; BERESFORD, R. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. Journal of pharmaceutical and biomedical analysis, 2000, 22.5: 717-727.

[43] BROOMHEAD, David S.; LOWE, David. Radial basis functions, multi-variable functional interpolation and adaptive networks. ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (UNITED KINGDOM), 1988.

[44] COVER, Thomas M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE transactions on electronic computers, 1965, 3: 326-334.

[45] SANGWAN, Om Prakash; BHATIA, Pradeep Kumar; SINGH, Yogesh. Radial basis function neural network based approach to test oracle. ACM SIGSOFT Software Engineering Notes, 2011, 36.5: 1-5.

[46] POWELL, Michael JD. Radial basis functions for multivariable interpolation: a review. In: Algorithms for approximation. Clarendon Press, 1987. p. 143-167.

[47] GURNEY, Kevin. An introduction to neural networks. CRC press, 1997.

[48] HAN, Hong-Gui; CHEN, Qi-li; QIAO, Jun-Fei. An efficient self-organizing RBF neural network for water quality prediction. Neural Networks, 2011, 24.7: 717-725.

[49] RUNXUAN, Zhang. Efficient sequential and batch learning artificial neural network methods for classification problems. 2005. PhD Thesis. Nanyang Technological University.

[50] MASHOR, M. Y. Some properties of RBF network with applications to system identification. International Journal of Computer and Engineering Management, 1999, 7.1: 34-56.

[51] OU, Yu-Yen, et al. TMBETADISC-RBF: discrimination of-barrel membrane proteins using RBF networks and PSSM profiles. Computational biology and chemistry, 2008, 32.3: 227-231.

[52] CARBONELL, Jaime G.; MICHALSKI, Ryszard S.; MITCHELL, Tom M. An overview of machine learning. In: Machine learning. Springer Berlin Heidelberg, 1983. p. 3-23.

[53] FISTER JR, Iztok, et al. A brief review of nature-inspired algorithms for optimization. arXiv preprint arXiv:1307.4186, 2013.

[54] EIBEN, Agoston E.; SMITH, James E. Introduction to evolutionary computing. Heidelberg: springer, 2003.

[55] EIBEN, Agoston E.; SMITH, James E. Introduction to evolutionary computing. Heidelberg: springer, 2003.

[56] BENI, Gerardo; WANG, Jing. Swarm intelligence in cellular robotic systems. In: Robots and Biological Systems: Towards a New Bionics?. Springer Berlin Heidelberg, 1993. p. 703-712.

[57] EBERHART, Russ C., et al. A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science. 1995. p. 39-43.

[58] VAN DEN BERGH, Frans; ENGELBRECHT, Andries P. A new locally convergent particle swarm optimizer. In: Proceedings of the IEEE international conference on systems, man and cybernetics. IEEE, 2002. p. 94-99.

[59] ROCHE, David; GIL, Debora; GIRALDO, Jesus. Study on Evolutionary Computation Stopping Criteria. 2010.

[60] ENGELBRECHT, A. P. Particle swarm optimization: Global best or local best?. In: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence. IEEE, 2013. p. 124-135.

[61] SHI, Yuhui; EBERHART, Russell. A modified particle swarm optimizer. In: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on. IEEE, 1998. p. 69-73.

[62] BANSAL, J. C., et al. Inertia weight strategies in particle swarm optimization. In: Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on. IEEE, 2011. p. 633-640.

[63] KENNEDY, James. The behavior of particles. In: International Conference on Evolutionary Programming. Springer Berlin Heidelberg, 1998. p. 579-589.

[64] KENNEDY, James; EBERHART, Russell C. A discrete binary version of the particle swarm algorithm. In: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on. IEEE, 1997. p. 4104-4108.

[65] LEE, Kwang Y.; PARK, Jong-Bae. Application of particle swarm optimization to economic dispatch problem: advantages and disadvantages. In: 2006 IEEE PES Power Systems Conference and Exposition. IEEE, 2006. p. 188-192.

[66] ENGELBRECHT, Andries P. Computational intelligence: an introduction. John Wiley & Sons, 2007.

[67] SONG, Ying; CHEN, Zengqiang; YUAN, Zhuzhi. New chaotic PSO-based neural network predictive control for nonlinear process. IEEE Transactions on Neural Networks, 2007, 18.2: 595-601.

[68] HUANG, Chao-Ming; WANG, Fu-Lu. An RBF network with OLS and EPSO algorithms for real-time power dispatch. IEEE Transactions on Power Systems, 2007, 22.1: 96-104.

[69] LIU, Hong-Bo, et al. Neural networks learning using vbest model particle swarm optimisation. In: Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. IEEE, 2004. p. 3157-3159.

[70] AL-KAZEMI, Buthainah; MOHAN, Chilukuri K. Training feedforward neural networks using multi-phase particle swarm optimization. In: Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on. IEEE, 2002. p. 2615-2619.

[71] JUANG, Chia-Feng; LIOU, Yuan-Chang. On the hybrid of genetic algorithm and particle swarm optimization for evolving recurrent neural network. In: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. IEEE, 2004. p. 2285-2289.

[72] FIRPI, Hiram A.; GOODMAN, Erik D. Designing Templates for Cellular Neural Networks Using Particle Swarm Optimization. In: AIPR. 2004. p. 119-123.

[73] DU, Yaping; WANG, Ning; ZHANG, Jianming. An optimum design method based on PSO algorithm for neuron controllers. In: Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on. IEEE, 2004. p. 2617-2621.

[74] PRASAD, MV Siva; GOTTIPATI, Ravi. A Novel Incremental Instruct Dynamic Intrusion Detection System Using PSO-RBF. International Journal of Future Computer and Communication, 2015, 4.4: 280.

[75] QIN, Zheng, et al. Evolving RBF neural networks for pattern classification. In: International Conference on Computational and Information Science. Springer Berlin Heidelberg, 2005. p. 957-964.

[76] WU, Fangyuan; KONG, Feng; YAO, Jiangyun. Intelligent Fault Diagnosis of Steer-By-Wire Automobile. Journal of Computers, 2012, 7.5: 1204-1211.

[77] RASHAG, Hassan Farhan, et al. Investigation of induction motor parameter identification using particle swarm optimization-based RBF neural network (PSO-RBFNN). International Journal of Physical Sciences, 2011, 6.19: 4564-4570.

[78] MALALI, Praveen; KOTINIS, Miltiadis. PSO-based Training, Pruning, and Ensembling of Extreme Learning Machine RBF Networks. 2015.

[79] LIU, Xiaoyong. Radial basis function neural network based on PSO with mutation operation to solve function approximation problem. In: International Conference in Swarm Intelligence. Springer Berlin Heidelberg, 2010. p. 92-99.

[80] QASEM, Sultan Noman; SHAMSUDDIN, Siti Mariyam. Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis. Applied Soft Computing, 2011, 11.1: 1427-1438.

[81] CUI, Liu; WANG, C. Y.; YANG, B. S. Application of RBF neural network improved by PSO algorithm in fault diagnosis. Journal of Theoretical and Applied Information Technology, 2012, 46.1: 268-273.

[82] ZHANG, Bi; WANG, Jia Yang; ZHANG, Su Lan. A New PSO-RBF Model for Groundwater Quality Assessment. In: Advanced Materials Research. Trans Tech Publications, 2012. p. 922-925.

[83] VACHKOV, Gancho; STOYANOV, Valentin; CHRISTOVA, Nikolinka. Growing RBF Network Models for solving nonlinear approximation and classification Problems. In: Proc. of the 29 th European Conference on Modeling and Simulation ECMS-2015, Albena, Bulgaria. 2015. p. 26-29.

[84] SUN, Jun; XU, Wenbo; LIU, Jing. Training RBF neural network via quantum-behaved particle swarm optimization. In: International Conference on Neural Information Processing. Springer Berlin Heidelberg, 2006. p. 1156-1163.

[85] ARANA-DANIEL, Nancy, et al. Smooth Path Planning for Mobile Robot Using Particle Swarm Optimization and Radial Basis Functions. In: Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012. p. 1.

[86] ZHONG, Yuanchang, et al. PSO-RBF neural network PID control algorithm of electric gas pressure regulator. In: Abstract and Applied Analysis. Hindawi Publishing Corporation, 2014.

[87] QASEM, Sultan Noman; SHAMSUDDIN, Siti Mariyam Hj. Improving generalization of radial basis function network with adaptive multi-objective particle swarm optimization. In: Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on. IEEE, 2009. p. 534-540.

[88] KORÜREK, Mehmet; DOĞAN, Berat. ECG beat classification using particle swarm optimization and radial basis function neural network. Expert systems with Applications, 2010, 37.12: 7563-7569.

[89] QIN, Zheng, et al. Evolving RBF neural networks for pattern classification. In: International Conference on Computational and Information Science. Springer Berlin Heidelberg, 2005. p. 957-964.

[90] LEE, Cheng-Ming; KO, Chia-Nan. Time series prediction using RBF neural networks with a nonlinear time-varying evolution PSO algorithm. Neurocomputing, 2009, 73.1: 449-460.

[91] NOMAN, Sultan; SHAMSUDDIN, Siti Mariyam; HASSANIEN, Aboul Ella. Hybrid learning enhancement of RBF network with particle swarm optimization. In: Foundations of Computational, Intelligence Volume 1. Springer Berlin Heidelberg, 2009. p. 381-397.

[92] CLARK, James; KOPRINSKA, Irena; POON, Josiah. A Neural Network Based Approach to Automated E-Mail Classification. In: Web Intelligence. 2003. p. 702-705.

[93] ÖZGÜR, Levent; GÜNGÖR, Tunga; GÜRGEN, Fikret. Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish. Pattern Recognition Letters, 2004, 25.16: 1819-1831.

[94] WANG, Huai-bin; YU, Ying; LIU, Zhen. SVM classifier incorporating feature selection using GA for spam detection. In: International Conference on Embedded and Ubiquitous Computing. Springer Berlin Heidelberg, 2005. p. 1147-1154.

[95] TAK, Gaurav Kumar; TAPASWI, Shashikala. Query Based approach towards spam attacks using artificial neural network. International Journal of Artificial Intelligence & Applications (IJAIA), 2010, 1.4.

[96] SABRI, Alia Taha, et al. Developing new continuous learning approach for spam detection using artificial neural network (CLA_ANN). European Journal of Scientific Research, 2010, 42.3: 525-535.

[97] KUFANDIRIMBWA, Owen; GOTORA, Richard. Spam Detection using Artificial Neural Networks (Perceptron Learning Rule). Online J Phys Environ Sci Res, 2012, 1.2: 22-29.

[98] LIU, Kun; TAN, Ying; HE, Xingui. Particle swarm optimization based learning method for process neural networks. In: International Symposium on Neural Networks. Springer Berlin Heidelberg, 2010. p. 280-287.

[99] CAO, Yukun; LIAO, Xiaofeng; LI, Yunfeng. An e-mail filtering approach using neural network. In: International Symposium on Neural Networks. Springer Berlin Heidelberg, 2004. p. 688-694.

[100] CHEN, Xiao-li, et al. A method of spam filtering based on weighted support vector machines. In: IT in Medicine & Education, 2009. ITIME'09. IEEE International Symposium on. IEEE, 2009. p. 947-950.

[101] JIANG, Eric. Detecting spam email by radial basis function networks. International journal of knowledge-based and intelligent engineering systems, 2008, 11.6: 409-418.

[102] DRUCKER, Harris; WU, Donghui; VAPNIK, Vladimir N. Support vector machines for spam categorization. IEEE Transactions on Neural networks, 1999, 10.5: 1048-1054.

[103] SALEHI, Saber; SELAMAT, Ali. Hybrid simple artificial immune system (SAIS) and particle swarm optimization (PSO) for spam detection. In: Software Engineering (MySEC), 2011 5th Malaysian Conference in. IEEE, 2011. p. 124-129.

[104] CHEN, Zhifeng; QIAN, Peide. Application of PSO-RBF neural network in network intrusion detection. In: Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on. IEEE, 2009. p. 362-364.

[105] CHUN-TAO, Man; KUN, Wang; LI-YONG, Zhang. A new training algorithm for RBF neural network based on PSO and simulation study. In: Computer Science and Information Engineering, 2009 WRI World Congress on. IEEE, 2009. p. 641-645.

[106] LIU, Bo, et al. Improved particle swarm optimization combined with chaos. Chaos, Solitons & Fractals, 2005, 25.5: 1261-1271.

[107] CHEN, Jun Ying; QIN, Zheng; JIA, Ji. A PSO-based subtractive clustering technique for designing RBF neural networks. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, 2008. p. 2047-2052.

[108] JIHONG, Qu; JUAN, Zhou; NANXIANG, Chen. Groundwater Table Prediction Based on Improved PSO Algorithm and RBF Neural Network. In: Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on. IEEE, 2010. p. 228-232.

[109] ALTMAN, Naomi S. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 1992, 46.3: 175-185.

[110] SONG, Yang, et al. Iknn: Informative k-nearest neighbor pattern classification. In: European Conference on Principles of Data Mining and Knowledge Discovery. Springer Berlin Heidelberg, 2007. p. 248-264.

[111] WEINBERGER, Kilian Q.; SAUL, Lawrence K. Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 2009, 10.Feb: 207-244.

[112] GOLUB, Gene H.; VAN LOAN, Charles F. Matrix computations. JHU Press, 2012.

[113] http://www.mathworks.com/company/aboutus/?s_cid=wiki_mathworks_1, 17/6/2016

[114] SUN, YING, et al. A Radial Basis Function Neural Network with Adaptive Structure via Particle Swarm Optimization. Taiwan: National Dong Hwa University, technical report 2009.

[115] LIU, Xiaoyong. Radial basis function neural network based on PSO with mutation operation to solve function approximation problem. In: International Conference in Swarm Intelligence. Springer Berlin Heidelberg, 2010. p. 92-99.

[116] MACKEY, Michael C., et al. Oscillation and chaos in physiological control systems. Science, 1977, 197.4300: 287-289.

[117] JANG, J.-SR. ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics, 1993, 23.3: 665-685.

[118] CHIU, Stephen L. Fuzzy model identification based on cluster estimation. Journal of Intelligent & fuzzy systems, 1994, 2.3: 267-278.

[119] SOKOLOVA, Marina; JAPKOWICZ, Nathalie; SZPAKOWICZ, Stan. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In: Australasian Joint Conference on Artificial Intelligence. Springer Berlin Heidelberg, 2006. p. 1015-1021.

[120] Spambase Data Set https://archive.ics.uci.edu/ml/datasets/Spambase. 18/5/2016.

[121] STONE, Mervyn. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1974, 111-147

[122] COSTA, Kelton, et al. Comparison of the Intelligent Techniques for Data Mining in Spam Detection to Computer Networks. 2014.

[123] SHARMA, Sumant; ARORA, Amit. Adaptive approach for spam detection. IJCSI International Journal of Computer Science Issues, 2013, 10.4: 23-26.

[124] JUKIC, Samed, et al. COMPARISON OF MACHINE LEARNING TECHNIQUES IN SPAM E-MAIL CLASSIFICATION. Southeast Europe Journal of Soft Computing, 2015, 4.1.

[125] FARIS, Hossam, et al. Improving knowledge based spam detection methods: The effect of malicious related features in imbalance data distribution. International Journal of Communications, Network and System Sciences, 2015, 8.5: 118.

ملخص

تحليل النص أو اكتشاف المعرفة في النص: هو عملية لاستخراج المعلومات ـ من النص غير المنظم . تحليل النص يمكن أن يعمل مع النص غير المنظم أو مجموعات البيانات الشبه منظمة مثل البريد الإلكتروني. المشكلة الاساسية التي واجهها مستخدمو البريد الالكتروني هي مشكلة البريد الالكتروني المزعج المسمى الرسائل غير المرغوب فيها، التي تستهلك مساحة تخزينية من خوادم البريد الالكتروني، تُضيِّع الوقت وتستهلك النطاق الترددي للشبكة.

تعتبر الشبكات العصبية الاصطناعية واحدة من أقوى أدوات تصنيف رسائل البريد الالكتروني، ولديها القدرة في التعامل مع كمية ضخمة من البيانات ذات الأبعاد العالية والحصول على أفضل النتائج.

وقد اقتصرت الدراسة في هذه البحث على نهج هجين الذي يمزج بين شبكة القاعدة الشعاعية العصبية وخوارزمية سرب الجسيمات. لهذا النهج الهجين اقترحنا تطبيقان، التطبيق الأول يستخدم لتقريب الاقترانات و للتنبؤ بالمتسلسلة الزمنية. وهناك تطبيق آخر هام من النموذج المقترح هو التصنيف، الذي يستخدم لتصنيف رسائل البريد الالكتروني. في كلا التطبيقين من النموذج الهجين المقترح، تم تحديد قيم عوامل شبكة القاعدة الشعاعية العصبية باستخدام الخوارزميات التالية: استخدمنا خوارزمية سرب الجسيمات لتحسين أفضل المواقع للمركز، خوارزمية الجار الأقرب لتحديد قيمة نصف القطر، وخوارزمية تفكيك القيمة الأحادية لتحديد قيمة الأوزان. داخل كل عملية تكرارية في خوارزمية سرب الجسيمات يتم تحديث عناصر شبكة القاعدة الشعاعية العصبية اعتماداً على اقتران الملاءمة (الخطأ).

في التطبيق الأول لتقريب الاقترانات و للتنبؤ بالمتسلسلة الزمنية، أجريت التجارب على ثلاثة اقترانات معيارية رياضية غير خطية و التنبؤ بمتسلسلة ماغي جلاس ، والنتائج التي تم الحصول عليها من بيانات التدريب توضح أن الطريقة المقترحة حسّنت دقة التقريب مقارنة مع الطرق التقليدية الأخرى. هذه النتائج تبين أنه تم تقليل قيمة جذر متوسط مربع الخطأ و مجموع مربعات الخطأ بشكل كبير مقارنة مع طرق أخرى. من خلال تجاربنا على تقريب الاقترانات، حصلنا على أفضل قيمة لجذر متوسط مربع الخطأ التي هي 0.000034

لتصنيف رسائل البريد الالكتروني في التطبيق الثاني، أجرينا التجارب على مجموعة بيانات رسائل مزعجة معيارية لتصنيف رسائل البريد الالكتروني إلى صنفين هما مزعج و غير مزعج، نتائج التجارب على هذا التطبيق تبين أنَّ هذه الطريقة عملت بدقة أكثر مقارنة مع الطرق الأخرى التى استخدمت نفس مجموعة البيانات. من خلال

تجاربنا على تصنيف رسائل البريد الالكتروني، حصلنا على قيمة دقة 93.7% التي هي الأفضل مقارنة مع أنهج أخرى، حيث كانت أفضل قيمة دقة لديهم 93.28 .