DYNAMIC SELF-ORGANISED NEURAL NETWORK INSPIRED BY THE IMMUNE ALGORITHM FOR FINANCIAL TIME SERIES PREDICTION AND MEDICAL DATA CLASSIFICATION

HAYA MOHMMAD ALASKAR

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

April 2014

ABSTRACT

Artificial neural networks have been proposed as useful tools in time series analysis in a variety of applications. They are capable of providing good solutions for a variety of problems, including classification and prediction. However, for time series analysis, it must be taken into account that the variables of data are related to the time dimension and are highly correlated. The main aim of this research work is to investigate and develop efficient dynamic neural networks in order to deal with data analysis issues. This research work proposes a novel dynamic self-organised multilayer neural network based on the immune algorithm for financial time series prediction and biomedical signal classification, combining the properties of both recurrent and self-organised neural networks.

The first case study that has been addressed in this thesis is prediction of financial time series. The financial time series signal is in the form of historical prices of different companies. The future prediction of price in financial time series enables businesses to make profits by predicting or simply guessing these prices based on some historical data. However, the financial time series signal exhibits a highly random behaviour, which is non-stationary and nonlinear in nature. Therefore, the prediction of this type of time series is very challenging. In this thesis, a number of experiments have been simulated to evaluate the ability of the designed recurrent neural network to forecast the future value of financial time series. The resulting forecast made by the proposed network shows substantial profits on financial historical signals when compared to the self-organised hidden layer inspired by immune algorithm and multilayer perceptron neural networks. These results suggest that the proposed dynamic neural networks has a better ability to capture the chaotic movement in financial signals.

The second case that has been addressed in this thesis is for predicting preterm birth and diagnosing preterm labour. One of the most challenging tasks currently facing the healthcare community is the identification of preterm labour, which has important significances for both healthcare and the economy. Premature birth occurs when the baby is born before completion of the 37-week gestation period. Incomplete understanding of the physiology of the uterus and parturition means that premature labour prediction is a difficult task. The early prediction of preterm births could help to improve prevention, through appropriate medical and lifestyle interventions. One promising method is the use of Electrohysterography. This method records

the uterine electrical activity during pregnancy. In this thesis, the proposed dynamic neural network has been used for classifying between term and preterm labour using uterine signals. The results indicated that the proposed network generated improved classification accuracy in comparison to the benchmarked neural network architectures.

Acknowledgements

In the Name of Allah, the Most Gracious, the Most Merciful

At the beginning, Praise and Great Thanks be to Allah (God) for providing me with the ability and patience to carry out this study.

I am indebted to many people and organisations that helped and supported me to complete this study. Firstly, I am very grateful to my supervisors, Dr Abir Hussain for her unlimited advice, encouraging remarks, feedback, and support throughout this journey. I also owe a great debt to Dr. Dhiya Al-Jumeily for his advice and encouragement from the beginning until now and I would like to thank him for his constructive comments on my thesis. I would like also to thank the Liverpool John Moores University, for their guidance and beneficial comments throughout the period of the research.

My heartiest and warm thanks to my family, who deserve great thanks for their support, patience and understanding throughout my PhD time: starting with to my son Abduallah and daughter Norah and my husband who suffered a lot and shared the ups and downs of this journey. My greatest thanks go to my parents for their prayers for me to return home successful. Finally, I would like to thank the Saudi government (Ministry of Higher Education) for sponsoring me through the scholarship. I owe a lot to my country and I hope I can pay back some of this favour.

CONTENTS

LIST OF FIGURE

LIST OF TABLES

ACRONYMS

ANN	Artificial Neural Network
AR	Annualised Return
AUC	Area under the Curve
AV	Annualised Volatility
CDC	Correct Directional Change
DSIA	Dynamic Self-organised network Inspired by the Immune Algorithm
DSMIA	Dynamic Self-organised Multilayer Inspired by the Immune Algorithm
E	Error Function
ECG	Electrocardiogram
EEG	Electroencephalography
EHG	Electrohysterography
EMG	Electromyograms
ERNN	Elman Recurrent neural network
FFT	Fast Fourier Transform
Lr	Learning rate
MAE	Mean Absolute Error
MDD	Maximum Drawdown
MF	Median Frequency
MLP	multilayer neural network
Mom	Momentum term

MSE	Mean Square Error
NMSE	Normalized mean square error
PF	Peak Frequency
PSD	Power Spectral Density
R-DSMIA	Regularised DSMIA
RDP	Relative Difference in Percentage of price
RecSOM	Recursive Self-organised Map
RMSE	Root Mean square Error
RNN	Recurrent neural Network
ROC	Receiver Operating Characteristic
SNR	Signal to Noise Error
SOM	Self-organised Map
SONIA	Self-Organised Network Inspired by Immune Algorithm
SR	Sharp Ratio
Std	Stander deviation
TPEHG	Term-Preterm EHG Database

LIST of PUBLICATION

International Journal:

Alaskar, H., Lamb, D. , Hussain, A. J. , Al-Jumaily, D. , Randles, M. , Fergus, P. (2013). Predicting Financial Time Series Data Using Artificial Immune System-Inspired Neural Networks, Int. J. of Artificial Intelligence and Soft Computing.

Hussain, A., Fergues, P., Alaskar, H., Al-Jumeily, D., (2014). Dynamic Neural Network Architecture Inspired by The Immune Algorithm to Predict Preterm Deliveries in Pregnant Women, Neurocomputing.

Alaskar, H., Hussain , A., Al-Jumeily, D., (Expected 2014).Survey of Recurrent neural network Application in medical. Submitted to *Plos One*, (under process).

International Proceeding:

Al-Jumeily, D., Hussain, A., & Alaskar, H., (2013) Recurrent Neural Networks Inspired by Artificial Immune Algorithm for Time Series Prediction. In Proceedings of International Joint Conference on Neural Networks, Dallas,, 2013, pp. 2047–2054.

Alaskar, H., Al-Jumeily, D., Hussain, A. J., Fergus, P., (2013). The Application of Self-Organised Network Inspired by Immune Algorithm for Prediction of Preterm Deliveries from EHG Signals, The 14th Annual Conference on the Convergence of Telecommunications, Networking & Broadcasting (PGNet), Liverpool, UK.

Alaskar, H., Hussain , A., Fergus, P., Al-Jumeily, D. (2014). Feature Analysis of Uterine Electrohystography Signal Using Dynamic Self-organised Multilayer network Inspired by the Immune Algorithm . The Tenth International Conference on Intelligent Computing, Taiyuan, China. (under review).

Al-Askar,H., Hussain, A.J., Al-Jumeily, D. & Radi, N. (2014). Regularized Dynamic Self Organized Neural Network Inspired by the Immune Algorithm for financial time series prediction. The Tenth International Conference on Intelligent Computing, Taiyuan, China. (underreview).

CHAPTER 1 INTRODUCTION

1.1 Introduction

Artificial neural networks (ANNs) have been prevalent in the usage of most machine learning applications in recent times. They have the power to predict and classify unknown patterns which are too complex for human observation (Makeshwar et al. 2010). In the literature, ANNs are also known as neurocomputer, connectionist network, and parallel distributed processor (Haykin 1998). ANNs have been proposed as useful tools in time series analysis in a variety of applications. Historically, they have also been proved to provide ways to overcome and solve practical problems such as prediction, classification, clustering, optimisation, etc. (Hertz et al. 1991). One type of neural network is the dynamic neural network, which is a neural network with feedback links. This dynamic neural network is applicable to various domains in order to deal with dynamical behaviour in time series data. The highly popular feed forward neural network is the multilayer neural network (MLP). It has been applied extensively in time series prediction. The main aim of this research is to use dynamic neural networks for the purpose of prediction and classification of time series data.

The prediction process is used to detect values or events that will occur in the future based on some previous and current knowledge of certain data. Examples of prediction include weather forecast, stock rate prediction, earthquake prediction, marketing and sales forecasting. Artificial neural networks can also be used for the prediction task and have very high success levels.

Classification techniques are able to categorise a set of data into groups of objects that share similar behaviours based on small subsets of training sets. Kohavi (1995) defines the classifier as a function that is able to map unlabelled data to label type. Classification techniques have been considered as the most crucial methods for decision-making and are widely used in data analysis. The purpose of such classification is to analyse the data in order to simplify the understanding of its structure.

Time series is a sequence of observations created by a complex system. Real time series are extremely useful in monitoring the behaviour of any complex systems over a given period. They can be used for analysis and forecasting of complex systems. Time series analysis has recently gained much attention from scientists and researchers, whose interest has led to different types of time series in different worldwide applications such as biological signals, time series for monitoring industrial processes, financial time series, etc. (Mirea & Marcu 2002; Roverso 2000; Fergus et al. 2013; Phinyomark et al. 2012; Adhikari & Agrawal 2013; Chowdhury et al. 2013a).

Time series usually contain a trend of random behaviour. Analysis of such data is not an easy task considering the various internal and external factors affecting these time series. In the theoretical analysis, Herrera assumed that time series are generated by a dynamic system (Herrera 1999). The systems that generate time series involve complex properties, which are: the relationships that exist between the elements of a time series are nonlinear, and include extensive dynamic behaviour. These properties make it very difficult to accurately analyse the behaviour of such systems even if the underlying properties are completely known. The investigation into analysing time series has essentially helped in the development of a number of techniques such as traditional and intelligent methods. While the traditional method requires assumptions about the characteristics of data, the intelligent technique is based on learning methodology, which is more dependent on large amounts of examples called training data. The learning methods help to learn the behaviour of time series and generate models based on using the training data set, consequently achieving a better learning model. However, the complexity of time series is such that there are no known details about the system that creates such time series; therefore such issues cannot be resolved by traditional methods. Analysis of time series behaviour of any complex systems such as the human body, stock markets or even countries' economies has always created a major challenge. The main advantage of using intelligent methods is not requiring any pre-information or details about time series.

1.2 The problem statements

Although there are massive applications of the well-known MLP neural networks, they suffer from difficulties such as the determination of the optimal number of hidden units, and estimating the best weight values. The selection of these parameters is very important to improve the performance of neural networks. Furthermore, the MLP neural network is affected by some learning algorithm problems such as over-fitting problems (Cao & Tay 2001; Giles et al. 2001; Widyanto et al. 2005). This means that the neural network can perfectly map between input and output in training data but it will not be able to sufficiently generalise its learning to new data.

There are a number of studies which have investigated the ability to use different techniques to improve the generalisation ability of feed-forward neural networks and to automatically select the best number of hidden units and their weights. One of these techniques was proposed by Widyanto et al. (2005). They designed a self-organised hidden layer inspired by an immune algorithm (SONIA). SONIA contains an immune algorithm in the self-organised hidden layer. The main aim of this network is to improve the recognition and the generalisation propriety of the MLP neural network. SONIA was used to predict temperature-based food quality; it showed an 18% improvement in correct recognition in comparison to the MLP network (Widyanto et al., 2005). Furthermore, this network has been used for prediction of financial and physical time series (Hussain & Al-jumeily 2007; Mahdi 2010; Mahdi et al. 2009; Mahdi et al. 2010) and for classification (Widyanto et al., 2006). However, SONIA is a feed-forward neural network, which means that it can solve static problems but cannot remember past behaviours and as a result cannot generate good results with dynamical temporal data (Ling et al. 2007). Therefore, the SONIA neural network has been improved by using recurrent links between its layers in order to deal efficiently with temporal patterns in time series. The main advantage of recurrent connections in a neural network is their ability to deal with static and dynamical situations (Ling et al., 2007; Makarov et al., 2008). These connections can offer the cognitive function such as memory association, classification or predication of dynamic system (Hopfield 1982; Jaeger 2004). The work of Makarov et al. (2008) showed that recurrent networks can be used to learn dynamic as well as static problems. Furthermore, it has been proved that using recurrent feedback in feed-forward neural networks can enhance the dynamic behaviour of the feed-forward neural network. It can improve the network’s ability to analyse time series that has been created by complex systems. Therefore, this research is focused on finding an optimal dynamic neural network that can deal with complex time series analysis problems, and this will be achieved by designed a novel dynamic self-organised neural network which is inspired by immune algorithm. These links can improve the performance of the network to deal with data better that the ordinary SONIA network. They are applied to model and analysis some time series signals emerging from two complex domains, financial and medical.

1.3 The scope of the research work

In the experiments undertaken in this research work, some interesting time series have been used. These time series signals are processed using the proposed dynamic neural network for prediction and classification purposes. These time series signals are:

- Financial time series signals.
- Biomedical signals

1.3.1 Financial Time Series Signals

Any time series generated from financial systems has the properties that its components, such as prices, are presented at certain times (daily, weekly, monthly, quarterly and yearly) and the movement of the prices is affected by factors (such as political events, rumours, business strategies, etc.). Financial time series signal is in the form of historical prices for example, the stock market. The future prediction of price in financial time series enables businesses to make profits. In this experiment, ten time series signals were used: six stock markets' signals, three currency exchange rate signals and one oil pricing signal.

1.3.2 Biomedical Signals

Uterine EHG signals refer to the electricity activity captured at the uterus during pregnancy using an electrodes. These signals reveal a great deal of information about the uterine contraction. This further helps to predict the chances of a woman going into labour as preterm or term. Early detection of preterm can provide early involvement to decrease and stop preterm birth (Chen, Chuang, Yang, & Wu, 2011; Iams, 2003). The experiment in this time series is based on using the proposed neural network for classifying EHG signals. Before classification, the time series must be transformed by using pre-processing methods: firstly, it has been filtered into 0.3-3 Hz. Then, these EHG time series are transformed into a four-dimensional vector space (feature space), as such features make the separation of groups easier. Three of these features are linear and one is nonlinear; these feature and filter parameters have been recommended by Fele-Zorz et al. (2008).

1.4 Motivation

The main motivation behind this research work was that, although extensive studies have been conducted for finding and designing the optimal neural network for classifying and predicting time series, these studies are mainly limited to performance. Classifying and predicting the time series signals is useful from a bigger perspective. For example, predicting potential prices in the stock market helps not only an independent investor point of view but also a country's economy. The first research work is targeted at predicting the future value of financial time series. More specifically, this study proposes to predict the future prices of stock prices based on their historical financial time series taken over a certain period of time. The main motivation for this research is to develop a new model of recurrent neural network in order to perform successful time series analysis for prediction and classification tasks.

To develop and test the concepts, the research has focused on financial time series, but once the methodology is confirmed to work well on these signals, the research can be expanded for use on other real-world signals. These types of experiments can be suitably applied to real-world problems. These applications can prove helpful in addressing some complex issues of time series signal prediction. The extensive amount of data sets used in this study give more insights into the functioning of complex systems, which were otherwise assumed to be black-box systems.

The proposed method in this study has been extended to the biomedical domain. In more specific terms, the research presented here is aimed at developing a recurrent neural network to classify the biomedical signals. It thus focuses on addressing the binary classification question of whether a patient will likely be normal or abnormal, e.g., in uterine EHG cases if a woman's delivery will be preterm or term.

1.5 Aims of the Thesis

The main aim of this thesis is to investigate and develop an efficient dynamic neural network approach in order to deal with time series analysis. This research work focuses on combining recurrent connections with a self-organised hidden layer inspired by immune algorithm. The recurrent connections improved the performance of the proposed model by having a "memory" of past information passed to the network through context units. The main objective of this dynamic neural network is for predicting and classifying a time series signal emerging from a complex system. Such systems are assumed to be a black box, wherein the properties of their components are unknown. This research work will attempts to investigate the ability and the performance of the proposed network based on two types of architectures for the purpose of analysing two types of real-time series signals. The main benefit of this network is that it has the ability to deal with variable length time series, and can still deal with the dynamic behaviour of the time series. Furthermore, it can be applied to classify between groups in time series.

This research has been applied into two different domains: financial time series, and biomedical time series, which is the uterine EHG. In the first part of the proposed experiments, financial data are utilised. Based on the promising results obtained with the financial data, the ability of the proposed network to be used as a forecasting tool can demonstrate. The second part of this research proceeds with possibility of extending the proposed neural network to be used with other domains. For this, the biomedical time series signals have been used for classifying preterm and term subjects in EHG signals. In this study, good classification results have been achieved. Various concepts were applied relating to pre-processing and transformation methods in the two types of signals that have been used in this thesis.

1.6 Objectives and Contribution of the Investigation

In order to investigate the research aims, some objectives have been set, as follows:

1. Develop a novel recurrent neural network architecture that can be used for the prediction and classification of real-time data. The proposed dynamic neural network is designed into two architectures. The first architecture is based on the Jordan recurrent neural network and is called Dynamic self-organised multilayer inspired by immune algorithm (DSMIA). The second architecture is developed based on the Elman recurrent neural network and is called Dynamic self-organised inspired by immune algorithm (DSIA).

2. Investigating the ability of the DSMIA network to deal with stationary and non-stationary time series.

3. Utilising the regularisation techniques in the DSMIA network.

4. Investigating the application of various recurrent neural network architectures for medical data analysis.

5. Implementing and evaluating different experiments to measure the performance of the proposed DSIA network to detect preterm classes from uterine EHG signals.

1.7 Thesis Structure

The remaining part of this thesis is organised as follows.

Chapter 2 will discuss the literature review about neural networks. A brief history and the main proprieties of neural networks will be addressed in this chapter. Furthermore, a detailed description of neural network architectures and their learning algorithms will be discussed. Chapter 3 will give a brief description of various types of recurrent neural network architectures. Furthermore, the chapter will discuss the various applications of recurrent neural networks in various domains. Chapter 4 will propose two dynamic neural networks, presented as an extension of the ordinary self-organised hidden layer inspired by immune algorithm. The two architectures of the proposed network as well as their properties and their learning methods will be presented. In Chapter 5 description about time series data and related studies that have been performed in analysing time series will be given. This chapter will deal with one type of time series, which is financial time series. It will review the fundamentals of financial time series forecasting, and address the difficulties and problems with this type of time series. Furthermore, Chapter 5 will include the background of the neural network and traditional forecasting methods' application in financial time series. Chapter 6 will discuss the application of recurrent neural networks in the biomedical domain. This chapter gives a brief introduction about medical time series. Then, it reviews the various techniques applied to analyse medical time series as well as the applications of recurrent neural network in medical time series analysis. A description of uterine Electrohysterography signals is also provided. Chapter 7 will describe the collection procedures and pre-processing methods used to assemble financial and uterine Electrohysterography signals. Furthermore, this chapter will present the simulation results for the various experiments that have been considered in this thesis. It also contains all the discussions from different experimental works. Finally, Chapter 8 will provide the conclusions and the directions for future work.

1.8 Chapter summary

Time series data are generated by very complex nonlinear dynamical systems. Therefore, analysis of such time series is very difficult. The challenge in time series prediction and classification is to discover the network model that would offer the best ability to deal with this type of data and to yield the best result. However, the neural network has some limitations in dealing with this type of data. Research work in this domain is continually leading to the introduction of numerous of neural networks for the purpose of analysing such dynamical systems. This thesis will propose one type of dynamic neural network known as Dynamic self-organised neural network inspired by immune algorithm. The performance of the proposed network will be examined in two types of time series domains, financial and medical data.

CHAPTER 2 ARTIFICIAL NEURAL NETWORKS

2.1 Introduction

An artificial neural network (ANN) is a system that processes information, which is inspired by the biological neural network in the human brain. In this chapter, the properties and the history of neural networks will be introduced. A number of neural network architectures have been designed; in this chapter, the feed-forward network will be presented. In addition, different type of learning algorithm will be explained at the end of the chapter.

2.2 Brief History

In previous decades, there have been great efforts to study and understand the information process in the human neural system and to simulate nervous-system learning in order to solve real-world problems. It has been recorded that, since 1901, great attention has been paid to understanding the human brain. For example, Rojas (1996) has mentioned that, from 1901 to 1991, almost 10% of the Nobel Prizes for Physiology and Medicine were awarded to scientists who contributed to the understanding of brain functions. It is worth mentioning that Cajal (1990) provided an important insight into the structural constituents of the brain when he introduced the best diagrams of neurons, which have been used ever since. The first artificial neural network was produced by Warren McCulloch and Walter Pitts in 1943 (McCulloch & Pitts 1943). They designed the first mathematical model of an ANN, which, although very simple, had substantial computing potential. Besides these examples, the ANN has been evolving since 1940 (Miller et al. 1992), where neurobiological researchers have worked together with the mathematics and computer science community to develop artificial neural networks (Rojas 1996). They have focused their attention on finding ways of training neural networks. Hebbian learning was developed in 1949, and was named after the neuropsychologist (Hebb 1949). In 1958, Rosenblatt extended the McCulloch and Pitts neuron. The Rosenblatt neuron is named the perceptron (Rosenblatt 1958).

In the mid-1980s, ANNs became very popular immediately after the back-propagation algorithm was developed by Rumelhart et al. in 1986 (Machado 1996a; Rumelhart & McClelland 1986). Events that followed this innovation led to thousands of papers being published on neural network applications. They have been borrowed by numerous disciplines (e.g. medicine, finance, physics, security, etc) in coming out with different hypotheses and theorems. They continue to attract new thoughts and ideas for improved performance, and a number of network architectures and learning algorithms are now being studied (Miller et al. 1992). They have been applied to perform many complex function approximations in different applications such as time series analysis, signal processing, pattern recognition, and image processing. In the literature, they have been put to practical use to solve various real-world problems related to areas such as medicine, science, engineering, business, etc (Dase & Pawar, 2010; Ispawi, Ibrahim, & Tahir, 2012; Sarbaz et al., 2011; Sriraam, 2011). These different applications have shown how extensively ANN can be utilised successfully to overcome problems, as compared to other standard statistical tools (Zhang, 2000).

There have been various studies and investigations conducted to improve and speed up the performance of neural networks by finding learning algorithms such as the Levenberg Marquradt Learning Algorithm (Hagan & Menhaj 1994). Besides, some researchers have tried to apply different techniques and approaches to develop the learning process in neural networks, such as using the momentum term (Miniani, 1990), variable learning rate (Jacobs 1988), and weight decay (Krogh & Hertz 1995).

2.3 Benefits of Artificial Neural Networks

Artificial neural networks have been considered as modelling tools and they have important properties. In the following section, these properties will be described.

First, neural networks can model linear and nonlinear relationships between variables (Scarborough, & Somers 2006). Nonlinearity means that the relationship between variables in a data set is not directly proportional, which means that slight changes in one variable can cause large changes in others (West 1985; Somers & Casal 2009). The property of nonlinearity is a very important and powerful aspect of neural networks. ANNs produce nonlinear models, which can be adapted to various problems. This property makes an ANN a more flexible model that is capable of dealing with real-world complex relationships (Zhang 2000). This flexible model has the power to learn the underlying connections between data samples. The neural network nonlinearity is due to the nonlinear activation function existing in neurons. It transforms input into output using a nonlinear function, thus resulting in true nonlinear parameters.

It is essential to take into account the fact that the relationships among variables in the data set will be linear as well as nonlinear; this will support the improvement of data analysis. Therefore, researchers have to consider using techniques that are capable of discovering both linear and nonlinear relationships. Some researchers have strongly recommended using nonlinear methods to analyse data. Somers in 2001 pointed out that significant connections among variables cannot be linear (Somers 2001). He focused on the application of neural networks in organisational research. His work has shown that the relationship between work attitudes and job performance is nonlinear, and the performance of nonlinear neural networks is more effective than linear regression methods (Somers 2001). Bettis in 1995 recommended the use of nonlinear methods to analyse strategy and adaptation of system dynamics (Bettis, 1995). In addition, ANNs are very beneficial in defining the upper boundary of analysis of variance when the relationships between variables are nonlinear. In general, ANNs present the prospect of new insights into the structure of relationships among variables.

Most ANNs are considered as universal approximation functions that are capable of approximating any continuous nonlinear function with the desired degree of accuracy (Zhang 2000; Hornik et al. 1989; Cybenko 1989; Roverso 2000). The approximation theory is based on approximate any function by evaluating a given set of values (Rojas, 1996). Universal approximation function theorem has been approved in a number of neural network architectures (Hornik et al. 1989; Cybenko 1989). Neural networks have many computational elements (neurons), each of which might use the nonlinear activation function. As a result, neural networks are able to model any general functions (Michie et al. 1994). Therefore, data with unidentified structure can be easily approximated and modelled by a neural network.

Furthermore, the ANN models function through a learning process. The neural network learning process is similar to the human nervous system, which is based on "learning by example". They are known as self-adaptive models, which adapt in order to approximate the behaviour of data (Ling et al. 2007; Detienne et al. 2003). Therefore, they have the ability to learn and adapt to various environments. Learning is defined as the capability to estimate the underlying behaviour in a specific set of data, which are called training data (Zhang 2000). ANNs can obtain any nonlinear mapping of training data through learning. The nonlinearity ability of neural networks is presented in the activation function that has been used to compute the neuron output. Thus, ANNs can be trained to represent any given problem behaviour. Hassoun (1996, p.1) revealed that "This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved but where training data is readily available". They have been designed with the capacity to adjust their synaptic weights in order to solve problems in different environments. Consequently, the weights will capture the information related to the problem from the training data. Thus, the knowledge of the ANN is retained across these weights. In addition, the neural network has the ability to generalise. This enhances the ability to predict output for given inputs that are not in the training set; thus, from the adaptive weights the network is able to generalise (Miller et al. 1992). Therefore, weights obtained from training data are applied to approximate targets of new data in the same environment, which are called test data.

Additionally, biological neural networks operate in parallel; thus ANN has a parallel processing capability, in which the calculation for each neuron is generally independent of all other neurons (Wesley-Smith, 2006). ANNs are able to discovers many competing hypotheses simultaneously by using parallel units, therefore ANN can be applied for parallel processing (Razmjooy et al. 2011) . ANNs can be used to solve complex problems, as used by Roverso for dynamic event recognition and fault diagnosis (Roverso 2002).

Finally, neural networks have the advantage of not using any assumptions about the underlying data structure and properties (Haykin 1998; Fausett 1994). Unlike standard statistical classification that needs prior information about data to perform classification, ANNs can model tasks and adjust their weights to the data without any previous information or specification on the underlying probability distribution (Zhang 2000).

2.4 Neural Network Architecture

An artificial neural network is a simulation model of the information processing of the human nervous system. This section will briefly describe some functions of the human nervous system that have stimulated the development of artificial neural networks.

From the beginning it is worth mentioning, as asserted by Rojas (1996), that the human brain has the ability to solve problems that no digital computer can yet professionally deal with. Nervous systems are very complex architectures that contain millions of elements called neurons (neural cells). Rojas asserted that both neurons involved in neural structure and the whole neural systems are considered self-organising systems that have the ability to deal with information in a variety of ways. Each of these types of neurons has been employed to perform a different function. These neural cells can create a response and output based on incoming signals. They transfer information by electrical signals. Signals run between neurons in a perfect manner (Rojas 1996). The fundamental nature of neural information processing is based on the relationship between electrical conduction of information in the cell and chemical transmission between cells.

The nervous system contains four components: Dendrites, Synapses, Cell body, and Axon. Each of these elements has its own function. The dendrites are branches on the ends of neurons, which pass the incoming information signals to the cell body. The cell body creates the essential chemicals to process the incoming signals from the dendrites. The output of the cell body is carried by the axon, which transfers the signals from the cell body out to other neurons. The synapses are contact points between one neuron and another, and they store neural network information. Thus, the direction of the information broadcast is controlled by synapses. The strength of the individual synapses and the organising of neurons are estimated by a complex chemical process (Rojas 1996).

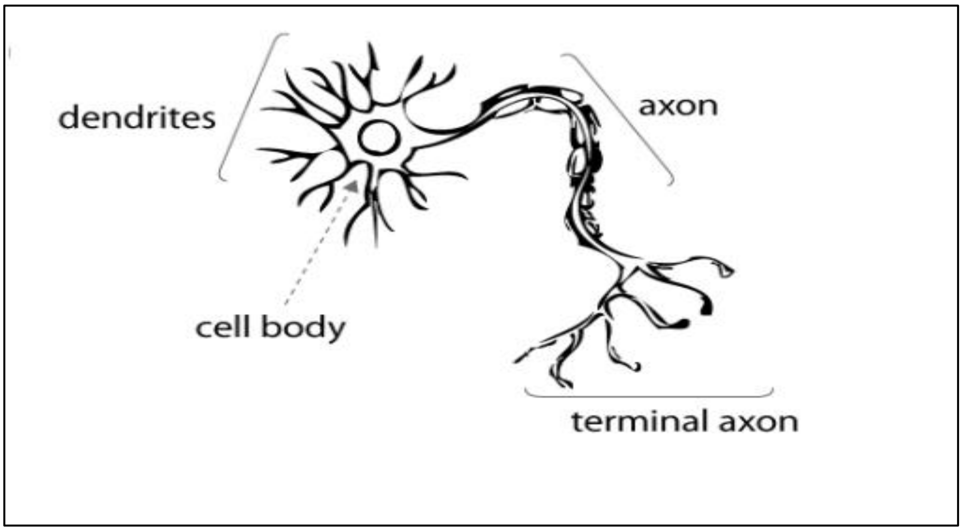


Figure 2.1: Diagrammatic representation of a neuron (Maltarollo et al. 2013)

Figure 2.1 illustrates a human neuron, where the neurons are attached through edges of the cell body. The synapses will be expressed in ANN models by embedding the weights. They will be adopted to contact neurons between the cell body and input or output neurons. They therefore measure the strength between neurons. They decide the importance of input signals and basically determine the network output. In addition, they can control the direction of the information broadcasting. Thus, ANNs will be able to capture underlying patterns from inputs and map them to outputs (Swingler 1996). A subgroup of neurons is organised in a layer. The first layer is the input layer while the last layer is referred to as the output layer. The hidden layers are the extra layers located between the input and output layers. The input neurons obtain impulses from external networks. The hidden neurons obtain their impulses from other neurons, and their outputs are transmitted to other neurons in the ANN. The output neurons present their output outside the network. Every neuron is a processing element with an activation function. The activation functions are referred to as squashing functions, which squeeze the amplitude range of neuron output to some limited value (Haykin 1998). The activation functions can be used in the neural network, as illustrated in Figure 2.3. Then, the neural network can modify the values of their weights in order to learn a specific function.

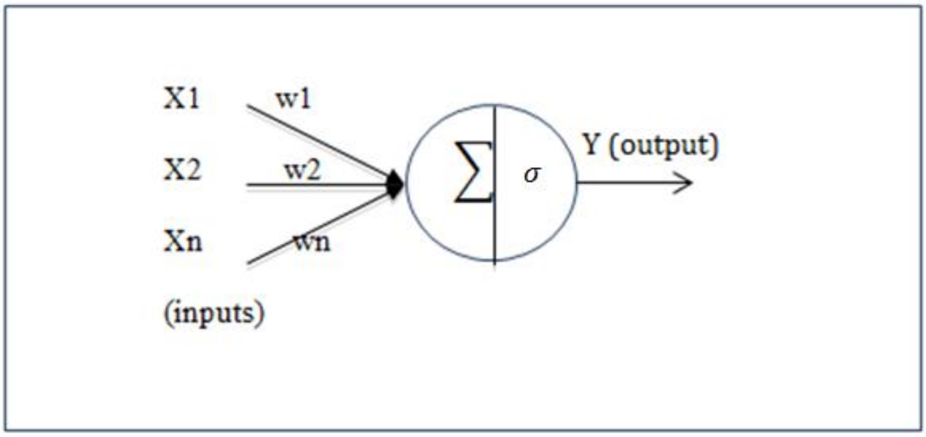


Figure 2.2: Artificial neuron

Figure 2.2 shows a simplified artificial neuron. Suppose (x_1, x_2, ….x_n) are input patterns for the neural network; the weight connect units i and j are represented by w_{ji} , *the* net_j are the sum of the input values x multiplying by the weight assorted with unit and computed as:

$$net_j = \sum_i w_{ji} x_i \qquad (2.1)$$

and the output from unit j is y_j which is the result of the activation function for net_j:

$$y_j = \sigma(net_j) \qquad (2.2)$$

Where σ is the transfer function. Common transfer functions used in the neural network are the logistic, the linear threshold, and the hyperbolic tangent, as shown in Figure 2.3.

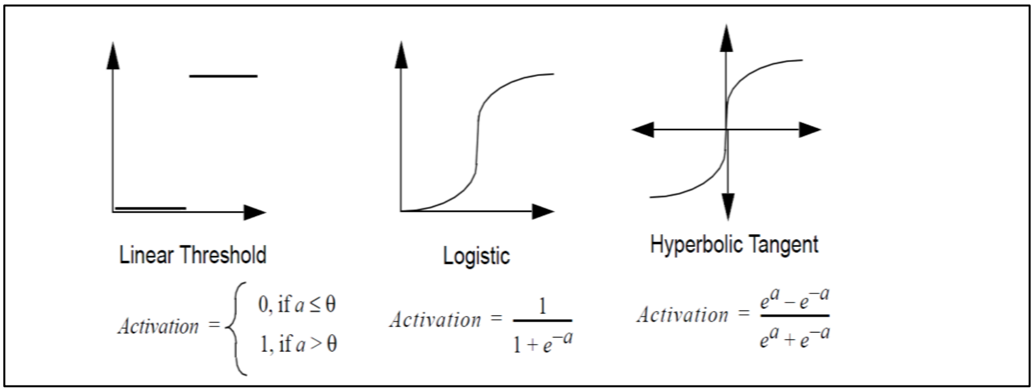


Figure 2.3: The activation functions

The linear threshold function has been used in the first neural network in Figure 2.3. It is based on the condition that the neuron is active if the impulse obtained is higher than a certain threshold. As for the logistic and hyperbolic tangent functions, they were used to make the function differentiable, and they differ only in the output range (0 to 1 for the logistic and -1 to 1 for the hyperbolic tangent).

Information on an ANN can be propagated by the neurons in different directions, and the connections between neurons can be presented in different ways. Therefore, a number of neural network architectures have been designed. In the next section, two types of ANN structures that are commonly used for classification and prediction will be described. The first one is the feed-forward neural network, which involves the information from the input layer being transmitted down through the network layer until it reaches the output layer. The second network type is called the recurrent neural network (RNN), as it incorporates recurrent links into its structure, such as the feedback connections.

2.4.1 Feed-Forward Neural Network

The feed-forward neural network is also called the multilayer perceptron (MLP). It is the most familiar type of neural network. In the literature, a number of neural network publications have been concerned with the feed-forward neural network. It has been applied successfully, compared with standard recognition tools. In addition to their history of successful application, MLPs are attractive because they were confirmed as being universal approximator function in 1989 by Cybenko and Hornik et al. (Hornik et al. 1989; Cybenko 1989). The universal approximation theorem asserted that a feed-forward network can approximate practically any nonlinear function (Gupta 2000).

The feed-forward network has a number of hidden layers. Each hidden layer has a number of hidden neurons. These hidden neurons provide neural networks with additional learning ability. They are able to learn more patterns and discover the hidden patterns in data. In addition, they can map the relationship between input and output neurons. Furthermore, the hidden layer in the MLP can map nonlinearities in data by using nonlinear transfer functions to compute the output of the hidden neurons.

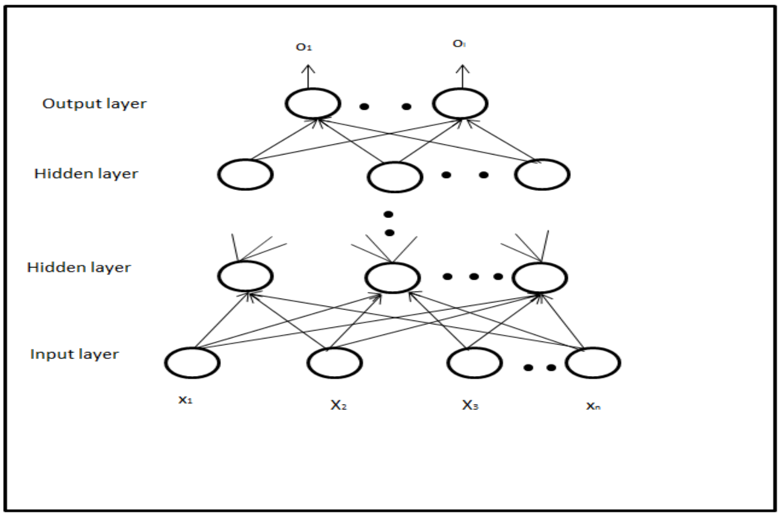


Figure 2.4: The feed-forward neural network

The basic fed forward neural network architecture is illustrated in Figure 2.4. The network consists of L number of layers: the first layer is called an input layer with N_I units, hidden layers with N_h units and an L layer is the output layer with N_o units. Let us suppose the inputs are $x=[x_1,x_2,....,x_{NI}]$, and weights are w; the network output will be $y=[y_1,y_2,...,y_{No}]$. The inputs are first passed to the input units in the input layer and then the outputs from the input units are sent to the hidden units in the hidden layer, which is the second layer, and so on, until finally the last layer of the hidden layers is reached, which is $(L-1)^{th}$ layer. The outputs of this layer are fed to the output units in the L^{th} layer. The calculation of the network is as follows:

$$u_i^1 = x_i, i = 1, \ldots \ldots, N_I \tag{2.3}$$

$$net_j^l = \sum_i w_{ji} x_i + B_j, j = 1, \ldots, N_h, l = 2, \ldots, L-1 \tag{2.4}$$

$$x_j^h = f_h(net_j^l) \tag{2.5}$$

Then the output value of the hidden layer will be calculated by sending the net_j to an activation function where f_h is the transfer function. Then the activation of the output units (y and l=1….N_o.) from the output layer are computed based on the hidden layer output x_j^h.

$$net_i^L = (\sum_j w_{ji}^o x_j^h + B_i), i = 1, \ldots., No \tag{2.6}$$

$$y_i = f_y(net_i^L) \tag{2.7}$$

The performance of the MLP neural network is affected by various factors such as type of transfer function, number of hidden layers or number of hidden neurons. The determination of the neural network architecture includes the selection of number of hidden layers and the number of hidden units in each hidden layer. Consequently, the use of different transfer functions or neural network architectures will create different results. For example, researchers have claimed that applying an ANN with more than one hidden layer can improve the required task (Hertz et al. 1991). However, other researchers have argued that complex nonlinear patterns can be effectively modelled by an ANN with one hidden layer (Bishop 1995; Ripley 1996). In terms of numbers of hidden neurons, no assumptions have been made about selecting an accurate number of hidden neurons for modelling tasks. Even if the required task is highly nonlinear, this does not mean that the neural network needs more hidden neurons (Gupta 2000). On the other hand, fewer hidden neurons might cause the neural network to learn the problem behaviour poorly and incompletely (Gupta 2000). However, the selection of the optimal numbers of hidden layers and hidden neurons for the required task is very challenging.

2.4.2 The Higher order Neural Networks

The higher-order neural network is a network with high order correlations of inputs to perform non-linear mappings with fewer layers. The higher order neural network exploits a higher combination of its inputs. There are different types of high order neural networks, such as functional link neural network, Pisigma neural network, and Ridge neural network.

Hussain et al. (2008) demonstrated that the Higher Order neural networks (HONNs) can successfully be used to predict financial time series. Their research involved various types of HONNs, such as functional link neural network (FLNN), Pi-Sigma neural network (PSNN), Polynomial neural network (RPNN) and novel polynomial pipelined network. These neural networks have been used to forecast several sets of exchange rate time series. Their result shows that the novel network can achieve best performance compared with other networks (Hussain et al. 2008).

2.4.2.1 Functional Link Neural Networks (FLNNs)

Giles and Maxwell (1987) first presented FLNNs as an addition to standard feed-forward networks. The FLNN architecture extends the feed-forward network, aiming to map non-linear relationships between input and output data. As the architecture is simpler, it is planned to decrease the computational cost incurred in the training phase, whilst demonstrating good approximation performance (Mirea & Marcu 2002). In spite of their otherwise linear nature, providing the input set is suitably descriptive, FLNN-architecture is able to offer a learning network with greater information capacity and complex learning ability (Cass & Radl,1996, Mirea & Marcu 2002).

FLNNs have created a popular strand of research. Fei and Yu (1994) showed that FLNNs have more powerful approximation capability than back-propagation networks, and form suitable models for system identification (Mirea & Marcu 2002). Cass and Radl (1996) used FLNN in process optimisation, discovering that the training speed is improved significantly compared to traditional MLP networks, without a loss in computational capacity.

However, FLNN architectures have some drawbacks. They suffer from weight explosion, with an exponential increase across the number of inputs.

The output of FLNN is specified as follows:

$$Y = \sigma\left(W_0 + \sum_j W_j X_j + \sum_{jk} W_{jk} X_j X_k + \sum_{jkl} W_{jkl} X_j X_k X_l + \dots\dots\dots \right) \tag{2.8}$$

where X and Y are the input and the output of the network, respectively. In this case, σ is a nonlinear transfer function, and w_o is the adjustable threshold.

2.4.3 Recurrent Networks

Since the main objective of constructing a neural network is to simulate the biological nervous network, it must be taken into account that the human neural system is actually dynamic. The human neural system has the ability to memorise past information from previous problems to use it to solve or deal with the next one. From the previous section it can be observed that there are no "memory" units in feed-forward networks that can save the output generated earlier by the network. Therefore, the standard feed-forward neural network must be extended with a recurrent connection to allow temporal dynamics into the neural network; this is known as a recurrent neural network (RNN). The RNN is considered to be a type of feed-forward network with integration of units called "context units"; these context units will hold the activation output produced by hidden or output layers. Chapter 3 will introduce the recurrent neural network in more detail.

2.5 Learning Algorithm

The ANN is considered a modelling approach, since the model can be designed by the learning from a specific set of data, which are training data. Hence, the knowledge in an ANN is based on the adaptive interconnection weights between neurons on each layer. The training process is an important property of ANNs. The samples of the training set are iteratively fed into the network, and each sample is considered information of the ANN; this information can be integrated into the network structure during the training process. While the training is taking place, the neural network adapts weights of each computing unit in each layer based on a learning algorithm. This adaptive processing will be repeated until the network learns to achieve the desired response (Rojas 1996). Accordingly, the learning operators will improve the neural network performance over time (Subasia et al. 2005). This is considered a correction step. Once the learning is completed, the neural network becomes a fast and accurate model of the original problem at hand. The trained neural network model can therefore be used on the test data to provide the optimal answer to the task it has learned. Different types of learning algorithms have been established to train neural networks. The oldest and best-known learning algorithm is Hebbian learning. The Hebbian rule is a method of computing changes in connection strengths between neurons (Hebb 1949). The Hebbian rule states that, if two neurons are active at the same time, the connection between neurons should be strengthened (Hebb 1949).

Neural networks can learn using supervised or unsupervised learning algorithms. Supervised learning neural networks are trained using a desired set of responses; this is known as learning with a teacher. The best-known supervised learning algorithm is the back-propagation learning algorithm, which is used to train multilayer perceptrons (MLPs). This is an attractive tool for prediction and classification tasks in many disciplines. On the other hand, in many cases the desired output and the right solution expected from the neural network are not provided. Therefore, unsupervised learning has been developed to deal with this kind of problem. Unlike with supervised learning, in unsupervised learning the desired response is unknown, so there are no teachers to help neural networks to learn. The best-known unsupervised neural network is the self-organising map (SOM). The SOM network is the most popular network and it can be applied for clustering tasks, visualisation and feature selection (Haykin 1998). The next sections will introduce these two types of learning.

2.5.1 Supervised Learning

In supervised learning, the neural network will be provided with input and output patterns during the training phase. In this type of learning, neural networks are taught to learn the pattern or behaviour of the input values in order to create an output that is already provided. Therefore, the learnt neural network will have the optimal weights, which will minimise the error function between neural network output and desired output on the training data set. The error function measures the difference between the network output and the desired output. Different error functions can be used to measure the performance of neural networks, such as the sum of squared errors for training data (Somers & Casal 2009; Werbos 1988). This error function has been used mostly in MLP.

2.5.1.1 Back-propagation Algorithm

As explained above, learning in neural networks means minimising the cost function by finding the optimal set of weights; this is achieved by the rule stating that, if the network output is not sufficiently close to the target, one should adjust the weights in the direction that will minimise the error. The most widely used learning algorithm for training a feed-forward multilayered network is called Back-propagation. This obtains the best set of weights for achieving the lower error by using gradient descent (Machado 1996; Rojas 1996). The main concept of this algorithm is to compute the effect of each weight in the network by using a training process.

During the training process two set of values are passed through the network:

Function values: the input values propagated from the input layer through the hidden layers until they reach the output layers, as shown in Figure 2.5.

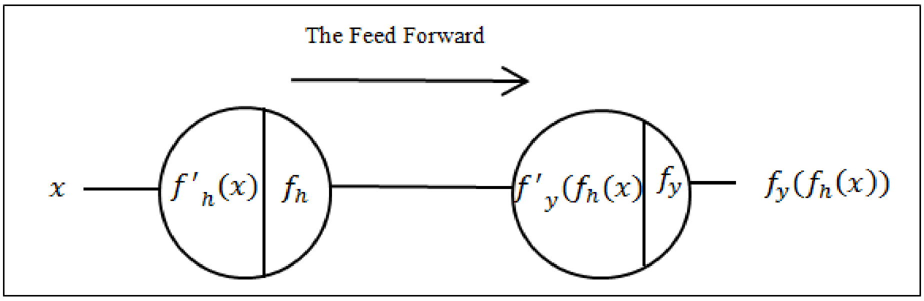


Figure 2.5: The Feed-Forward

Error values: the errors are computed through the back-propagation algorithm. The error is computed from the output layer and propagated backwards from the output layer, through the hidden layers, until it reaches the input layer, as represented by Figure 2.6. Therefore, each layer returns its error back to the previous layer.

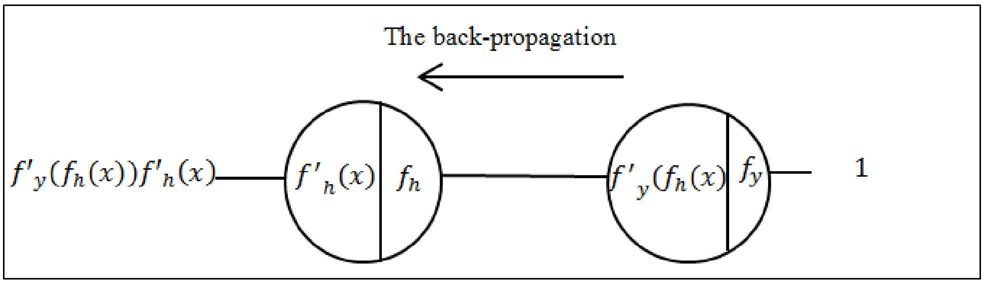


Figure 2.6: The process of back-propagation

The weights are modified based on the Delta rule. During training the network performance is evaluated by computing the error function. The output created by the network will be compared to the actual target, which is called the error function. It is quantified by:

$$e_k(t) = d_k(t) - y_k(t) \tag{2.9}$$

Where e_k represents the error between the actual target $d_k(t)$ and activation output $y_k(t)$ at time t. where k indicates the output unit. The weights are modified in order to make the network output close enough to the actual target. The error function to be minimised is:

$$E(t) = \frac{1}{2} \sum_k e_k(t)^2 \tag{2.10}$$

E will be the sum of *e(t)* over all patterns in the training data. Then the error is fed backwards through the neural network layers. Thus, the weights are adjusted at each layer, launched by the output layer. Each $W_{\hbar ji}$ is updated by the negative of the gradient to reduce the error. The weight change $\Delta W_{\hbar ji}(t)$ is calculated from the derivative of the errors with respect to the connections' weights as given by:

$$\Delta w_{ji}(t) = \alpha \frac{dE}{dW_{hji}}(t) \tag{2.11}$$

$$\frac{dE}{dw_{ji}} = \alpha \frac{dE}{dy_i} \times \frac{dy_i}{dnet_i} \times \frac{dnet_i}{dw_{ji}} \tag{2.12}$$

Where α is the learning rate and $W_{\hbar ji}$ is the weight from unit *j* to unit *i*, y_i is the output of the units in the *l* layer, and net_i is computed as : $y_i = f_y(net_i)$

$$net_i = \sum_j W_{hji} x_j \tag{2.8}$$

$$\frac{\mathrm{d}net_i}{dW_{hji}} = \frac{d(W_{hji} x_j)}{dW_{hji}} = x_j \tag{2.14}$$

Derivative of the error with respect to the activation and where y_i is referring to the output of the output layer which is named here as y, then $\frac{dE}{dy_i}$ will be:

$$\frac{dE}{dy_i} = -(d_i - y_i) \tag{2.15}$$

and if f_y is sigmoid function, then $\frac{dy_i}{dnet_i}$ will be computed as following:

$$\frac{dy_i}{dnet_i} = y_i(1 - y_i) \tag{2.16}$$

The value of the derivatives will used to decrease the error function by using the gradient descent as following:

$$W_{hji}(t+1) = W_{hji}(t) - \alpha \frac{dE}{dW_{hji}}(t) \tag{2.17}$$

The learning rate α is used to manage the learning process. It can be used to accelerate convergence to reach minimum error function; this can be done by increasing the learning rate. However, using a very large learning rate will lead the network to drop into the oscillatory traps of learning algorithm and it may pass the local minimum. Meanwhile, a small learning rate can be stuck in a local minimum of error function (Rojas 1996; Ghazali 2007). Therefore, the momentum term used to overcome this problem allows the decrease of oscillations in the training process (Rojas 1996), where μ is the momentum term.

$$\Delta w_{ji}(t) = \alpha \frac{dE}{dW_{hji}}(t) + \mu \Delta W_{hji}(t-1) \tag{2.18}$$

The error is minimised by using the gradient descent where the weights are modified along the negative direction of the gradient of error function E as illustrated in Figure 2.7, which shows the behaviour of error E with respect to one weights w. If the gradient of is E negative, then weight w must be increased. In contrast, the weight w must be decreased if the gradient of E is positive. Consequently, this will decrease the error at the next iteration (Holst 1997; Gupta 2000).

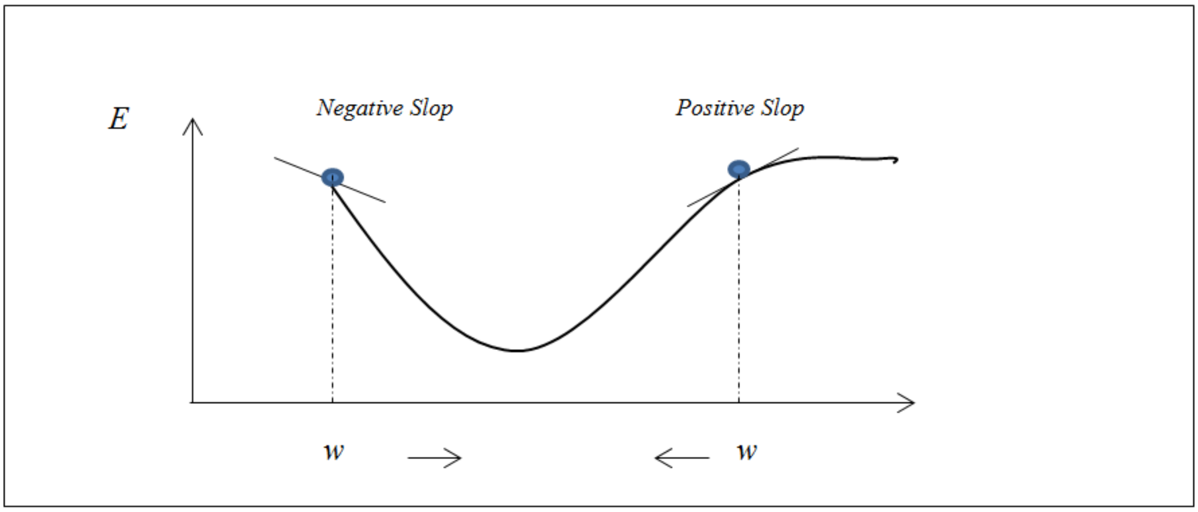


Figure 2.7: The principle of gradient descent

2.5.2 Unsupervised Learning

In unsupervised learning neural networks, there are no demands for the desired output. Unlike supervised learning, the neural network learns patterns and behaviour of data sets and adjusts their weights without extra help (Rojas 1996). This is known as self-learning. The network therefore has to create and develop its own output from the input by calculating the suitable connection weights.

Researchers have offered different methods for this type of learning, one of which is the self-organising map (SOM). This was designed by Kohonen in 1982, and his main objective in designing an unsupervised learning tool was to present a large set of inputs by a set of neurons. SOMs can be used to represent the topology nature of the data. It can incorporate multi-dimensional input vectors and can capture relations between them in a 2-dimensional matrix, as illustrated in Figure 2.8.

Self-organising neural networks attempt to learn in order to recognise hidden patterns in unlabelled data. The self-organising map clusters the input data into groups and discovers features implicit in the problem. The centres of clusters in the sample distribution are represented by each neuron in the SOM. The clustering problem is known as categorisation, where the class labels and the exact number of classes are unidentified, and the samples must be clustered together in some reasonable way. The SOM has been used in a number of practical applications in different disciplines (Salhi et al. 2009). It has been utilised successfully for pattern recognition and image analysis (Huang & Wu 2009).

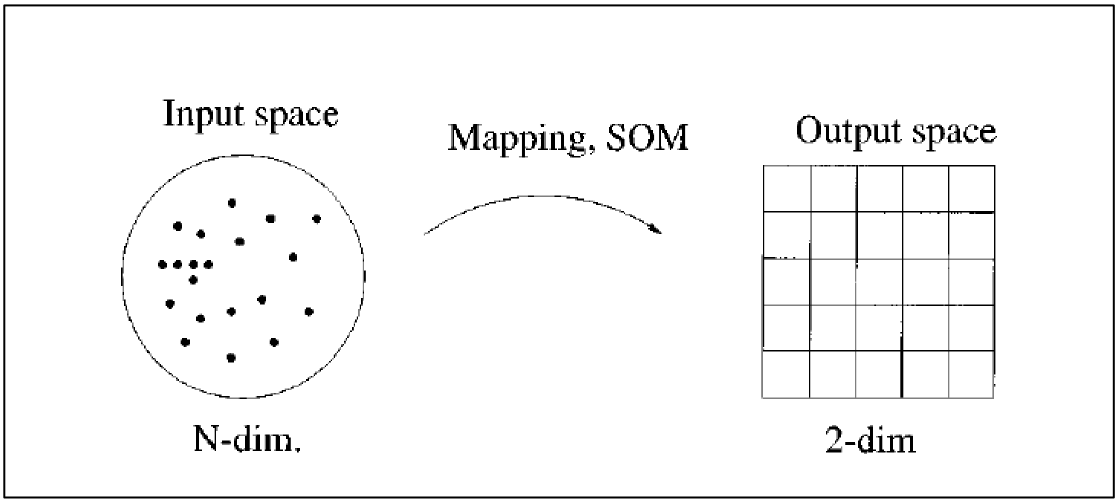


Figure 2.8: The Self-organising Map structure (Lagerholm et al. 2000)

2.5.2.1 Self-organised Map (SOM)

The training in SOM is a type of competitive learning. Competitive learning is based on the idea that the neuron with a weight vector that is similar to the input data is adjusted towards the input data. This neuron is called the Best Matching Unit (BMU) or winning neuron. The best matching unit (BMU) is the neuron that wins the competition, which is shown in Figure 2.9 as a red circle. The winner neuron and the neuron in the same area of the neighbourhood are adjusted.

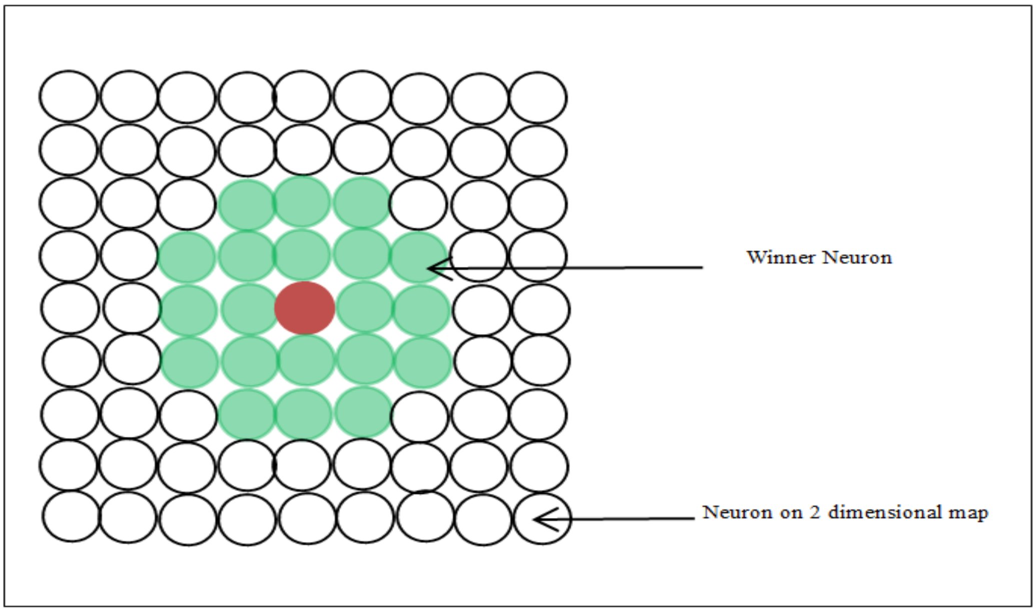


Figure 2.9: The SOM learning algorithm

For instance, let input data of vector $X=\{x1,x2,x3,\ldots\}$ represent "x_i" as input of unit *i*. Each input of *x* has been compared with neuron weights of the grid by finding its Euclidean distance to all weight vectors as follows:

$$E_i = \|x - W_{hi}\| \tag{2.19}$$

Then the winning *c* vector must be found, which minimised the measurement distance.

$$E_c = min(E) \tag{2.20}$$

which represents the best matching unit (BMU), where the weight of this units is much similar to the input units. Then the learning rule will be applied to adjust the weights of the BMU neuron and the neuron close to the BMU.

$$\Delta w_i = \gamma h_{ic}(x - W_{hi}) \tag{2.21}$$

Where γ is learning rate and h_{ic} is neighbourhood function, which minimise the distance between unit *i* and *c* on the map. The common neighbourhood function in the SOM is Gaussian function. $\hbar_{ic}$ is a Gaussian function of the Euclidean distance *d(i, c)* between units *i* and *c* in the map with σ Gaussian width:

$$h_{ic} = exp(-\frac{d(i,c)^2}{\sigma^2}) \quad (2.22)$$

Since the weight vectors of the neurons are modified iteratively for all the training data, this will allow these weights to capture the distribution of the input data due to the neighbourhood updating, and when a new input arrives every neuron competes to represent it. Thus, the SOM learns without a supervisor.

2.6 Chapter Summary

Neural networks have received great attention from different researchers in many applications. In the literature review, neural networks have been applied widely to solve a variety of real-world problems. They have the capacity to provide solutions without complete knowledge of the data structure; this makes them suitable to solve real-world problems. This chapter has provided a brief overview of the neural network properties and architectures. Furthermore, the two types of neural network learning methods have been highlighted. The next chapter will focus on one of the neural network architectures, which is the recurrent neural network.

CHAPTER 3 RECURRENT NETWORKS

3.1 Introduction

The recurrent neural network (RNN) can model sequential data in order to deal with temporal processing tasks such as signal processing, system identification, control, pattern classification and sequence pattern recognition. It is especially appropriate for dynamic problems. A recurrent neural network (RNN) is also known as a dynamic neural network, or a neural network for temporal processing. The main aim of this chapter is to provide an overview of the recurrent neural network (RNN). The different architectures of RNNs will be presented. Each type of neural network has its own strengths and capabilities that can be applied in various types of problems ranging from signal prediction and pattern recognition to data classification. The Jordan and Elman networks are considered the best-known recurrent networks (Mahdi et al. 2010; Szkoła et al. 2011). In this chapter, various recurrent neural network architectures including the Jordan and the Elman networks will be investigated, in addition to the recursive Self-organised neural network.

3.2 The Concept of Recurrent Neural Network (RNN)

RNN is a type of neural network with a memory capable of storing information from past behaviours (Haykin 1998). Typically, real-world data, such as financial time series or biomedical signals, are often constructed as a sequence of observations in time dimensions. According to Kim (Kim 1998), there are a number of methods that can be used to present time in neural networks. These include the creation of a spatial or explicit representation of a temporal pattern by extracting features from temporal data, putting time delays into layers or their connections such as time-delay neural networks (TDNNs), applying recurrent links in RNNs, utilising neurons with activations summing inputs over time, and the last method is a combination of all the previous methods.

The RNN enables the network to deal with static and temporal information in the input sequence, both for prediction and classification. In contrast to the feed-forward ANN, which only has connections from layer to layer moving forward from the input layer through the hidden layers to the output layer, the RNN has feedback loops. It has delayed feedback links that connect neurons on one layer back into the neurons of either the same layer or a previous layer. These connections will provide the network with the ability to incorporate the current information as well as previous information. In the RNN, the activation output of each unit is passed to other units.

$$net_j(t) = \sum_m W_{hjm} y_m(t-1) \tag{3.1}$$

$$y_j(t) = f(net_j(t)) \tag{3.2}$$

Where w_{mj} is weights that connect unit *m* to unit *j*. The activation output of each unit is *y(t)* and *t* referred to time.

3.3 The Properties of the RNN

The problem of sequence patterns is that the different patterns are related to one another. In other words, data patterns occur in sequence; therefore, the order of the patterns is very important. In order to model this type of pattern, the model system needs a memory to hold the past information. It has been proved that neural network models are able to capture the temporal nature of any time series signal by using feedback links such as RNN. The RNN performs perfectly on this type of problem compared with the feed-forward neural network. Many experiments have proved that the RNN can deal with sequence patterns more than the feed-forward neural network (Burrows & Niranjan 1994). These recurrent links will provide the feed-forward networks with the capability of dynamic procedure, which means that the neural network learning process is based on previous information in addition to current input data. One of the most important applications of the RNN is to model or identify dynamical patterns, which are based on time factors (Chung et al. 2009). The time factor is very crucial for some real-world time series signals. Thus, the requirement of a feedback connection is important. In a RNN, the connections between neurons can appear as directed cycles and loops, which let the network represent very complex dynamical behaviour. Furthermore, these connections attempt to discover an appropriate temporal representation that can capture the properties of the input sequence.

Furthermore, Kremer et al. (1995) mentioned that, since the 1960s, researchers have shown that recurrent artificial neural networks could approximate any finite state machine with the desired accuracy. Nonlinear hidden nodes provide the RNN with great approximation power. This allows the hidden neurons to integrate information over many time steps and store the information in an internal memory that saves past outputs, and uses them when creating the next outputs. This can be used to generate predictions with high accuracy (Sutskever et al.

2011). Moreover, once the RNN is combined with the nonlinear activation transform function, it is able to deal with and process complex spatiotemporal patterns (Forney & Anderson 2011). Another advantage of using recurrent networks is that they do not need to select the specific length of the time series that must be stored in the memory, as occurs with the tapped-delay methods (TDNN). Furthermore, they show their ability to map between input and output in classification task (Übeyli 2010; Petrosian et al. 2001). They obtained high accuracy in the classification compared to feed forward neural network. RNNs can be useful for different applications, including filtering, prediction, pattern classification, stochastic sequence modelling, and associative memory.

Time series prediction involves the use of the sequences of signals that vary during specific times to predict future values. Thus, problems such as predicting time series are very difficult to solve based on some sets of current inputs. Previous research has shown that recurrent neural networks are capable of generating better predictions by utilising previous and current values from the signal (Giles et al. 2001; Ghazali et al. 2009; Swarup et al. 2005). Recurrent links create memory to hold information of previous activation states (Hopfield 1982). Therefore, as Forney and Anderson (2011) asserted, RNNs will have the ability to learn tasks that require memory. These recurrent connections will improve the performance of the neural network, as Haykin stated (1998). However, the dynamic link in the neural network suffers from lack of stability of the network performance (Mishra & Patra 2008; Haykin 1998). Hence, stability of recurrent neural networks has been investigated extensively in number of studies (Voegtlin 2002; Zhang 2008; Mozer 1989; Campolucci 1998; Tsoi & Back 1994; Ghazali 2007; Atiya 1988; Barabanov & Prokhorov 2002; Kosmatopoulos & Christodoulou 1994).

3.4 Recurrent Neural Network Architectures

There are a number of recurrent neural network architectures based on the purpose of application or structure of networks. They can be divided into two types, which are fully or partially connected. In fully connected RNNs, each unit in input, hidden and output layer is connected recurrently, while in partial RNNs the feedback links are connected partially. A fully recurrent neural network as shown in Figure 3.1 has feed-forward besides feedback links in any order; each of these connections is trainable (Ghazali 2007). The first fully recurrent neural network structure to use the physical principle of storing information in a dynamically stable configuration was proposed by Hopfield (1982). Fully RNNs are general architectures and powerful for specific tasks (Campolucci 1998). Williams and Zipser (1989) have used fully recurrent network architecture, where the network has one hidden layer, for nonlinear adaptive filtering and pattern recognition. In spite of the advantages of using a fully recurrent network, it has some drawbacks: since each unit has connection weights, this means that, if the network consists of a set of n full connected neurons, ($O(n^2)$) weights connections must be fed back to all neurons. This will lead to a more complicated structure (Ku & Lee 1995). Furthermore, each of these weights is trainable, so the training process will be very slow and more complex (Tsoi & Back 1994). Fully connected RNN is very complicated when dealing with complex problems (Übeyli 2010). Although they are able to model very complex dynamical systems, simpler recurrent neural networks such as partial RNNs are preferable for some specific problems (Campolucci 1998).

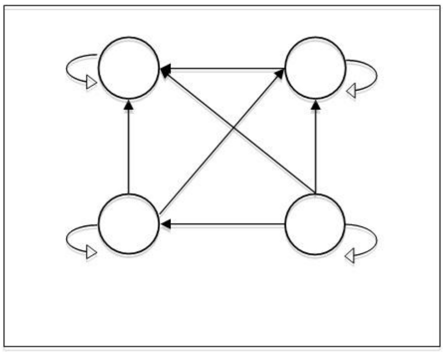


Figure 3.1: Fully Recurrent Networks

In a partial RNN, the network is extended from the feed-forward network by adding "context" units to the structure of the network, which can be used for storing to hold some information about previous states within the network (the previous network activation). This type of recurrent neural network has the ability to remember cues from the previous inputs. However, the connection on a partial RNN does not noticeably complicate the structure and the training procedure of the network. The partial RNNs enjoy simpler training than fully recurrent networks. Simpler recurrent neural networks make use of available prior information and knowledge, which can be better (Mozer 1989). Therefore, partial RNNs can be preferred in some dynamic systems more than fully RNNs (Tsoi & Back 1994; Ku & Lee 1995; Elman 1990; Mozer 1989; Campolucci 1998).

Since they can be used for time series prediction and modelling, a number of partially recurrent neural network architectures have been proposed in the literature. The most popular types of recurrent neural networks are shown in Figure 3.2.

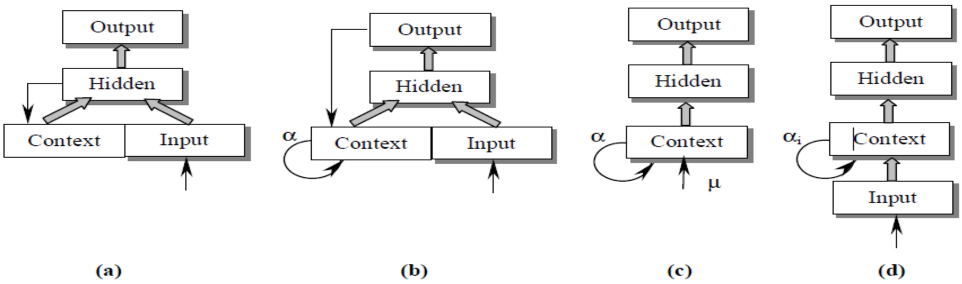


Figure 3.2: Different structures of recurrent neural networks (Hussain & Liatsis 2002)

Figure 3.2(a) represents the Elman recurrent neural network, 3.2(b) shows the Jordan recurrent neural network, 3.2(c) shows the model proposed by Stornetta et al, (1987) and 3.2(d) shows the model proposed by Mozer (1989). The next subsection will introduces these types of recurrent neural network architectures.

3.4.1 Elman Neural Network (ERNN)

The Elman Recurrent Artificial Neural Network (ERNN) was designed by Jeffrey Elman in 1990 in order to discover patterns in natural languages (Elman 1990). The ERNN architecture is similar to the feed-forward neural network with an input, hidden and output layer in addition to some context units. These context units store the past activation outputs from the hidden layer. Therefore, the recurrent connections in the ERNN are performed from the hidden layer to the input layer through the context-switching nodes.

The recurrent link allows the ERNN to identify and generate temporal patterns and spatial patterns. These links allow the output of the hidden nodes at time *t* to influence the output of hidden units at time *t+1*. This means that the hidden units do not only detect the actual input but also take information on their own activation output at the last time step by the context units (Mashhadany 2012). This is done when interrelations between the current input and the internal context units are processed to generate the hidden layer output and send the relevant past information in the context units (Mankar & Ghatol 2009).

The context units are linked to the hidden layer (as shown in Figure 3.2(a)). The main benefit of feeding back the output of the hidden layer is that the context units hold the same dimension and representation as in the hidden layer. The context units store previous information that has been represented in the previous hidden units. This means that the ERNN will try to learn the mapping between the input and the context units to the output. Furthermore, these context units provide the ERNN with a short-term memory; this memory will support the ERNN to have the greatest ability to learn problems. The feedback links help the ERNN to detect patterns (Ghazali 2007). Since their appearance, ERNNs have been used effectively on a number of practical problems (Forney 2011). In addition to their successes recorded in different applications, ERNNs have proved to have universal approximates functions (Kremer 1995). In other words, ERNNs can learn sequences generated from a finite state machine (Kremer 1995; Forney & Anderson 2011; Elman 1990; Huet 1993). The dynamic equations of the ELMAN networks are as follows:

$$y_i(t) = W_o * X_h(t) \tag{3.3}$$

$$X_h(t) = f(W_h X_i(t) + W_z Z(t)) \tag{3.4}$$

$$Z(t) = X_h(t-1) \tag{3.5}$$

Where W_z is the weight of the context unit, and *Z(t)* is the value of the context unit. ERNNs have been applied extensively for different applications including classification, regression, forecasting and generalisation of sequence data (E. M. Forney & Anderson 2011; Übeyli & Übeyli 2008; Szkoła et al. 2011; Mankar & Ghatol 2009). They have been used in different domains including medical fields, e.g. speech recognition, disease classification (Ilbay et al. 2011; Übeyli & Übeyli 2008; Übeyli 2009; Übeyli 2010). The results of these different studies confirmed that the ERNN achieved high accuracy in biomedical data classification. Furthermore, they have been used in economic fields such as financial prediction (Giles et al. 2001; Yümlü et al. 2005), and time series classification (Husken & Stagge 2003). Kremer presented a paper discussing the computational power of Elman (Kremer 1995). Ramadevi et al. (2012) presented a paper on finding the role of the hidden neurons in the ERNN. They used seven hidden layers; each layer had a number of hidden neurons. Their experiment used the network to classify Cavitation signals.

Wang et al. (2013) designed a novel hybrid optimisation algorithm Elman recurrent neural network. The improved Elman network was developed based on using two types of particle swarm optimisation algorithms: discrete particle swarm optimisation (DPSO) algorithm and improved particle swarm optimisation (IPSO) algorithm. A DPSO algorithm is used to determine the structure of a proposed network, while an improved particle swarm optimisation (IPSO) algorithm is used as a learning algorithm. This network combines the benefits of the two optimisation methods, which are DPSO and IPSO algorithms. The DPSO algorithm is employed to find the number of hidden units of the RNN, whereas the IPSO algorithm is adapted to adjust parameters (including weights, initial inputs of the context nodes) for each of the structures (particles) existing in the DPSO. The authors' main aim was to develop an algorithm to automatically determine the best Elman network structure as well as its optimal parameters. Their network was utilised to predict three time series: real-time data of a thermal system in a 600-MW power plant, Mackey-Glass time series and CATS time series. The result from their experiment demonstrated that this proposed network has achieved higher prediction accuracy and improved the generalisation aspect of the Elman network. The experimental results demonstrated the ability of the proposed algorithm to automatically select the structure of Elman (number of hidden nodes) as well as adapt their parameters efficiently (Wang et al. 2013).

3.4.2 Jordan Neural Network

The Jordan neural network is similar to the feed-forward network with feedback links from output units to a set of context units. It was designed by Jordan (Jordan 1990), who used the network to find its ability to learn sequential tasks in language processing. The main aim of designing the Jordan network was to make a neural network capable of showing temporal variations and temporal context dependence (Jordan 1990). It can play a valuable role in time series prediction and controlling the system. In this network, the recurrent links are presented from the output layer to the input layer, in which the input units hold a copy of the values of the external inputs, while the context units hold a copy of the values of the feedback link from the previous output units, in addition to self-feedback connections from the context units to themselves, as illustrated in Figure 3.2(b). Therefore, the output units are linked to context units in the input layer; thus the network outputs at time (t–1) will also be the input at time t. The context units in the Jordan neural network are called state units. The dynamic equations of the Jordan neural network can be determined as follows:

$$y_i(t) = W_o * X_h(t) \tag{3.6}$$

$$X_h(t) = f(W_h X_i(t) + W_z Z(t) \tag{3.7}$$

$$Z(t) = y(t-1) + \alpha(y(t-1) \tag{3.8}$$

α represents the strength of the self-connection, and *Z(t)* represents the context unit. Since the Jordan network holds a copy of the past time series input and its own forecasts, this property has provided recurrent networks with the ability to discover information outside a limited time period (Pissarenko 2002). The Jordan network has been applied in many applications including spoken language understanding (Mesnil et al. 2013).

Tellez (2013) presented a paper on improving the performance of the Jordan recurrent neural network for spoken digits recognition. Genetic algorithms have been used for optimising the performance of the Jordan recurrent neural network by finding the least number of hidden neurons necessary to get best performance. Three architectures of recurrent neural networks – Elman, Jordan and the combination of Elman and Jordan networks – were utilised in this paper in order to evaluate their performance with spoken Spanish digits. The result confirmed that the Jordan model gives the best recognition performance, which was 98.75% compared to Elman, which produced 93% (Tellez 2013). Another application of the Jordan network was presented by Silva et al. (2010). They used the Jordan network to reconstruct the missing data on a medical time series signal. They used a signal with a multivariate channel. In this paper, the Jordan network was trained to predict the missing data in order to recover the corrupted signal (Silva et al. 2010).

3.4.3 The Stornetta and Mozer Recurrent Neural Networks

There is another structure of recurrent neural network. The main difference between these recurrent neural networks and the above recurrent neural networks is that the context units are replaced on one single layer after the input layer. Therefore, network input only reaches the rest of the network via the context units.

Figure 3.2(c) shows another structure of a partially recurrent neural network, which was developed by Stornetta (Stornetta et al. 1987). This network pre-processes the current inputs and the previous values of the context units themselves and feeds them to the network through the context layer.

$$Z_i(t) = \mu x_i(t) + \alpha(Z_i(t-1)) \tag{3.9}$$

Where $Z_i(t)$ is the output of the context unit at time t, $x_i(t)$ is the current input at time t, μ is the input amplitude and $\alpha$is the decay rate. Therfore, the context layer is fed by the current input as well as part of the previous values of the context layer itself. Hence, the network will have current input and the past history of the network input. This type of network has been used for financial forecasting (Shin & Han 2000)

Figure 3.2(d) represents another recurrent neural network, which is designed by Mozer (1989). It has four feed-forward layers. The additional layer, which carries the context units, is called the context layer. It has feedback connections which allow the network to consider the context history. These feedback connections are trainable. Moreover, the units in the input and context layers are fully linked. The trainable context units perform as an integrator to keep the previous context states. Furthermore, the adjustable recurrent connections help the network to find a suitable decay rate to the context layer (Araújo & Darbo Jr. 1997). This can provide the network with the ability to learn and model the temporal domain of the current task (Huet 1993). The dynamic equation of this network is:

$$Z_i(t) = \alpha\big(Z_i(t-1)\big) + f(\textstyle\sum_j W_{hji}\, x_j(t)) \tag{3.10}$$

Where $Z_i(t)$ is the activity of the contxt unit i at time t, α is a decay weights, f is the sigmoid function, and $x_i(t)$ the input unit j. The weights from the input unit to the context unit are adjusted. The self-feedback connection gives the context units with inertia (Zhang 2008).

A number of learning algorithms were designed to train recurrent neural networks. Some of these algorithms were based on the gradient descent approach and the computation of the partial derivatives to train the recurrent neural networks. However, all of these developed learning algorithms are extended from the back-propagation learning algorithm (Rumelhart & McClelland 1986). These include real-time recurrent learning algorithm (Williams & Zipser 1989), and back-propagation through time (Doya 1995). The next section will introduce some developed recurrent neural networks and their learning methods.

3.5 Advanced Recurrent Neural Networks

There are a number of recurrent networks according to purpose of application or structure. In this section, some recurrent neural network architectures that have been used for financial or medical applications are introduced. Their learning algorithms and their applications will be highlighted.

3.5.1 Combined Jordan and Elman Neural Network – Elman/Jordan

This network was developed by combining the Elman and Jordan networks. This combined network was designed in order to develop the performance of the recurrent networks and to overcome their limitations. For example, the learning procedures in the Jordan network are very slow. Furthermore, this combination can improve the learning ability of the Elman network.

The context unit will hold a copy of the activity of the hidden layer as well as a copy of the output layer. The weights in this network are adaptive dynamically using a back-propagation learning algorithm. The context units in this network have made the network able to extract temporal information from time series. This extra information has a great impact on the network performance.

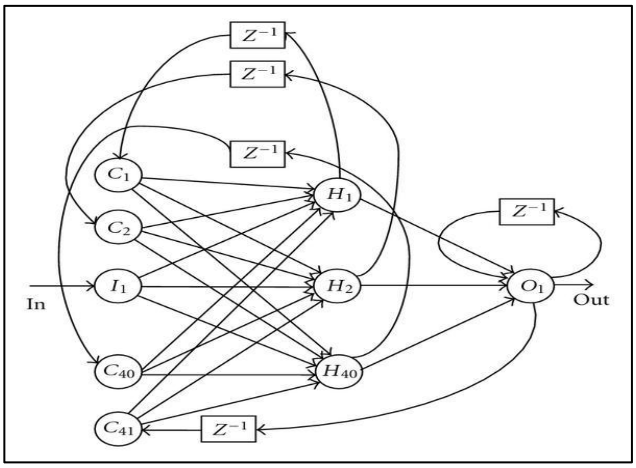


Figure 3.3: Structure of the trained Elman/Jordan neural network (Szkoła et al. 2011)

The dynamic equation of this network is:

$$y(t) = f\left(\sum_l w_l^{out} f\left(\sum_i w_{ij}^{n} x_i(t) + \sum_j W_{hj} x_{hj}(t-1) + w_o^{z} y(t-1)\right) + y(t-1)\right) \quad (3.11)$$

Where $x_{hj}(t-1)$ is the activation of the hidden unit *j* at time *t-1*, $x_i(t)$ is the input unit *i* at time *t*, and *y(t-1)* is the output of the network at previous time. The combination of the Elman and Jordan networks has been applied in different applications, speech recognition and signal filtering. Szkola et al. (2011) applied the combined Elman/Jordan to analyse speech signals in order to indicate the difference between normal persons and patients with larynx diseases. The features of patients' speech signals are extracted by the average mean squared errors obtained by Elman/Jordan for the original signal. The task utilises speech signals of patients from the healthy group and those with two types of laryngopathies, namely Reinke's edema (RE) and laryngeal polyp (LP). Their experiment involved asking patients to separately pronounce different Polish vowels. Their proposed network has shown some improvement on the learning ability of the neural network and time speed. Their approach can be utilised as an initial step in making decisions about normal and disease states (Szkoła et al. 2011).

Another application of the Elman/Jordan network was used without the self-recurrent connection in the output neurons. This study was presented by Mankar and Ghato (Mankar & Ghatol 2009), who used the Elman/Jordan network in order to remove noise from EMG signals. The noisy EMG signals were fed into the Jordan/Elman network. Their result showed that the combination of Jordan and Elman networks has successfully decreased the noise from the EMG signals. The performance was evaluated using mean squared error function (MSE) that had been computed between noisy EMG signals and the desired EMG signals; the network achieved 0.001 in MSE measure (Mankar & Ghatol 2009). Chowdhury et al. (2013) have also used this network for EMG noise removal. Their experiments confirmed that the performance of this network seemed to be a more exact achievement of the target pattern and it was faster. The main benefit of this combination is that it is capable of generalisation (Chowdhury, 2013).

3.5.2 Echo Recurrent Neural Network (ESN)

The recurrent network of an echo-state network involves an 'echo-state' characteristic (Verplancke et al. 2010). This 'echo-state' is utilised as a fading memory. The echo-state network was introduced by Jaeger et al. in 2001. The Echo state network (ESN) is considered as part of reservoir computing methods which are based on recurrent neural network (Verplancke et al. 2010). It involved two parts, as illustrated in Figure 3.4. The first part is a recurrent network with a number of units and weights that connect the units with each other, and this part is called the dynamic reservoir. The input units are applied to the dynamic reservoir as shown in Figure 3.4. The reservoir in the network performs as a fading memory, which is where the echo state name comes from (Verplancke et al. 2010). The second part is output units which are connected to the neurons of the dynamic reservoir (Mashhadany 2012; Jaeger 2001).

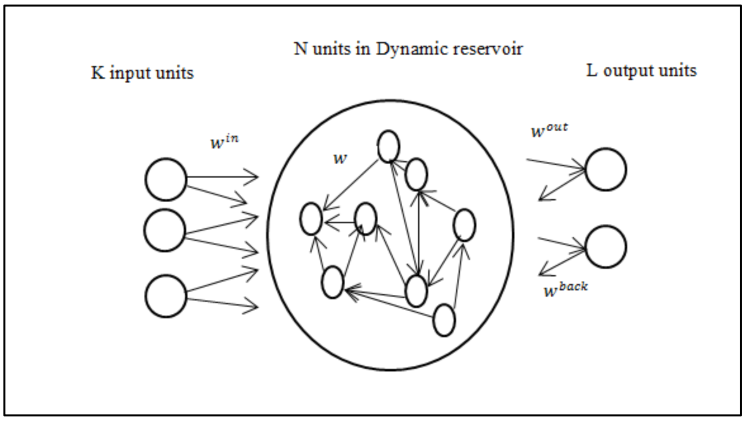


Figure 3.4: Echo State Networks

Let the network involve N_i input units, N_r reservoir units and N_o output units. The input unit i at time t is $x_i(t)$, the internal unit j in the dynamic reservoir is $u_j(t)$, and of output units is $y(t)$. The weight matrix that connects the input unit i to the reservoir unit j is w^{in} (t), the weights that connect the units in the dynamic reservoir are $w(t)$, and the weight that connects the internal units and input units to the output units is w^{out}. In addition, there are weights that reconnect the output units to reservoir units, w^{back}.

The activation of the internal units is computed as*:*

$$u(t) = f(w^{in}x(t) + wu(t-1) + w^{back}y(t-1)) \tag{3.12}$$

Where f is the transfer function. The output units are collected as:

$$y(t) = f(w^{out}(x(t), u(t), y(t-1))) \tag{3.13}$$

The weights in the dynamic reservoir are trainable by back-propagation algorithm. However, the recurrent weights on the ESN network do not adjust during training.

The weight connections in this network must be selected carefully and have to be small enough to avoid growing oscillations (Mashhadany 2012). The network output units are recurrently feed-back to the units of the dynamic reservoir. The main aspect of the ESN is that not all weight connections should learn during training, only the connections from the dynamic reservoir to the output units, which are the output weights w^{out}, are adapted. The main benefits that can be derived from using the ESN is that their recurrent weights do not change during training (Husain et al. 2008; Mashhadany 2012). This will reduce the complexity and time of the network training process. The main concept of the ESN is that, with a large pool of hidden neurons, the ESN will be able to represent dynamics behaviour (Schmidhuber et al. 2007).

ESNs are utilised in many applications as temporal pattern recognisers, pattern generators, predictors, and controllers. They are used for predicting movement data of people moving (Hellbach et al. 2008). They can be used for time series classification, being applied as a classifier in speech recognition (Skowronski & Harris 2006). ESNs are used to predict the Mackey-Glass time series (Jaeger, H. 2004). The first medical application of an echo network was by Verplancke et al. (2010), where an ESN was used to predict dialysis in intensive care unit patients (Verplancke et al. 2010) .

3.5.3 Dynamic Ridge Polynomial Neural Network (DRPNN)

Ghazali et al. (2009) have developed a new dynamic neural network architecture which includes a feedback connection in addition to the feed-forward Ridge Polynomial Neural Network. The Ridge Polynomial Neural Network is a generalisation of the PSNN network. The PSNN is a network with one hidden layer of summing units and product units in the output layer.

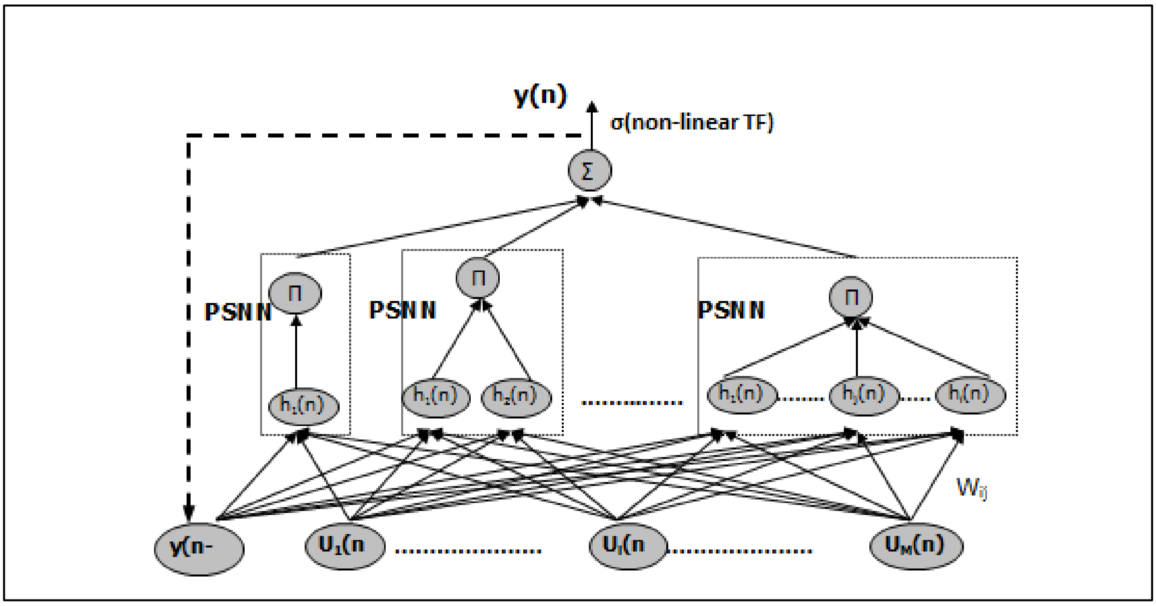


Figure 3.5: Dynamic Ridge Polynomial Neural Network of K-*th* (Ghazali 2009)

The RPNN network consists of a number of pi-sigma neurons of increasing orders. The architecture that has been used in the DRPNN network is the same as in the Jordan network; hence there was a recurrent connection from the output layer. The recurrent link provides DRPNNs with the capability of having a memory. This memory will help the DRPNN to exhibit a rich dynamic behaviour. Therefore, the network outputs are based on the initial values of external inputs, and the entire history of the system inputs (Ghazali et al. 2010).

Let *N* be the number of external inputs *x(t)* to the network and *y(t-1)* be the output of the DRPNN at previous time step. The inputs of the network are the combination between *x(t)* and *y(n+1)* and are referred to as *U(t),*

$$U_i(t) = \begin{Bmatrix} x_i(t), if\ 1 \le i \le N \\ y(t-1), if\ i = N+1 \end{Bmatrix} \tag{3.14}$$

The hidden layers are computed as equation:

$$h_m(t) = \sum_{i=1}^{N+1} W_{hmi} U_i(t) \tag{3.15}$$

Where the $p_i(t)$ of the output layer are computed as:

$$p_i(t) = \prod_{m=1}^{i} h_m(t) \tag{3.16}$$

$$p_i(t) = f(\sum_{i=1}^{k} p_i(t)) \tag{3.17}$$

Where k is the number of pi-sigma units used in DRPNN, and f is nonlinear activation function, and $p_i(t\ )$ is the output of each PSNN block. The cost function is the squared error between the real signal and the predicted signal, that is:

$$E(t) = \frac{1}{2}\sum_k e_k(t)^2 \tag{3.18}$$

$$e(t) = d(t) - y(t) \tag{3.19}$$

And where *d(t)* is the actual output. The learning algorithm in the network is the real-time recurrent learning algorithm (Williams & Zipser 1989). Therefore, the change for W_{hmi} of the weights matrix is computed according to the following equation:

$$\Delta W_{hmi}(t) = -\gamma \frac{\partial \mathrm{E}}{\partial \mathrm{W_{hmi}}} \tag{3.20}$$

Where γ is the learning rate the value of $\frac{\partial \mathrm{E}}{\partial \mathrm{W_{hmi}}}$ is computed as:

$$\frac{\partial E}{\partial w_{im}^{h}} = \frac{\partial E}{\partial y(t)} \frac{\partial y(t)}{\partial W_{hmi}} \tag{3.21}$$

$$\frac{\partial E}{\partial w_{im}^{h}} = e(t) \frac{\partial y(t)}{\partial w_{im}^{h}} \tag{3.22}$$

$$\frac{\partial y(t)}{\partial w_{im}^{h}} = \frac{\partial y(t)}{\partial p(t)} \frac{\partial p(t)}{\partial w_{im}^{h}} \tag{3.23}$$

$$\frac{\partial y(t)}{\partial p(t)} = f'(\textstyle\sum_{i=1}^{k} p_i)(\ \prod_{L=1}^{k} h_l(t)) \tag{3.24}$$

$$\frac{\partial p(t)}{\partial w_{im}^{h}} = w_{im}^{h}(t) \frac{\partial y(t-1)}{\partial w_{im}^{h}} + U_i(t)\delta_{mk} \tag{3.25}$$

Let $d_{im}(t-1)$ be as:

$$d_{im}(t-1) = \frac{\partial y(t-1)}{\partial W_{hmi}} \tag{3.26}$$

And δ is the Krocnoker delta, from Eq (3.24), (3.25) and (3.26) $\frac{\partial y(t)}{\partial W_{hmi}}$ is:

$$\frac{\partial y(t-1)}{\partial W_{hmi}} = f'(\textstyle\sum_{i=1}^{k} p_i)(\ \prod_{i=1}^{m} h_i(t))(\ W_{hmi}(t) d_{im}(t-1) + U_i(t)\delta_{mk}) \tag{3.27}$$

Then the update rule for $\mathrm{W_{hmi}}(\mathrm{t}+1)$is:

$$W_{hmi}(t+1) = W_{hmi}(t) + \Delta W_{hmi}(t) \tag{3.28}$$

The DRPNN was utilised to predict the upcoming trends of financial time series signals. Extensive experiments were presented for the prediction of financial time series. Their results showed that the DRPNN showed advantages in capturing chaotic movement in the financial time series. The experiment results showed that the DRPNN produced better prediction value in terms of the annualised return for the prediction of the exchange rate signals. The experiments demonstrated that the DRPNN is suitable for predicting nonlinear and non-stationary time series (Hussain et al. 2006; Ghazali et al. 2010; Ghazali et al. 2009). There is another application of this network to predict Standard & Poor's (S&P) 500 stock index future signals. The result showed the ability of the DRPNN to forecast S&P signals with lower forecast error (Ghazali et al. 2010).

3.6 Recurrent Neural Networks for Unsupervised Learning

There are a number of unsupervised neural network structures for temporal pattern recognition; these neural networks are considered to be modified versions of self-organising maps (SOMs). They extended SOMs with additional feedback connections that allow for natural processing of time series (Tino et al. 2005). These networks include the temporal Kohonen map (TKM) (Chappell & Taylor 1993), the recurrent SOM (RSOM) (Koskela et al. 1998), and the recursive SOM (RecSOM) (Voegtlin 2002).

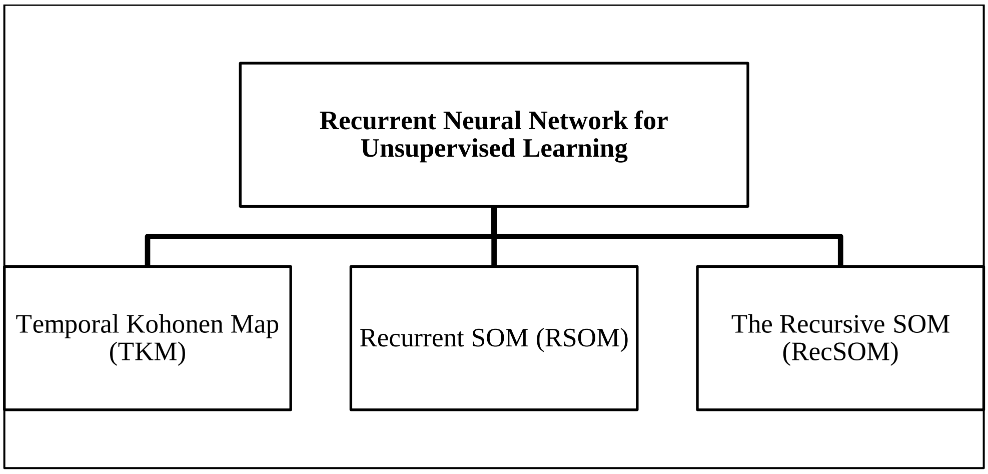


Figure 3.6: Recurrent unsupervised neural network types

The next sections will introduce the three types of unsupervised recurrent networks that have been used widely in many applications.

3.6.1 The Temporal Kohonen Map (TKM)

The TKM is considered to be the first proposed feedback SOM. It appeared in 1993, designed by Chappell and Taylor (Chappell & Taylor 1993). This network is extended from Kohonen's self-organising map. The TKM approach is based on establishing a leaky integrator in the output of each SOM unit. The leaky integrator is considered to be a low frequency linear filter. This is done by storing the output values of each unit and using them in the calculation of the next output value of that unit. Hence, it is considered different from the original SOM only in its output activities. In a TKM the final output of each unit is defined as

$$V_j(t) = a * V_j(t-1) - \frac{1}{2} * \left\|x(t) - w_j(t)\right\|^2 \tag{3.29}$$

Where *x(t)* is the input sequence and *j* refers to the unit on the map, the $V_j(t)$is the neuron activation, at time *t* and *a* is refers as time constant which is between (0 and 1). The best matching unit (BMU) *c* in the TKM is the one that maximises the activity output y:

$$V_c(t) = max\{V_j(t)\} \tag{3.30}$$

The learning rule of TKM is the same as SOM, which is computed as:

$$w_j(t+1) = w_j(t) + \gamma h_{jc}(x(t) - w_j(t)) \tag{3.31}$$

Where h_{jc} is the value of the neighbourhood function, which gives the excitation of unit *j* when the best matching unit is *c*. The common neighbourhood function in SOM is Gaussian function and γ is learning rate. A TKM network is presented in Figure 3.7 showing the current activation of the neuron unit is based on previous activation.

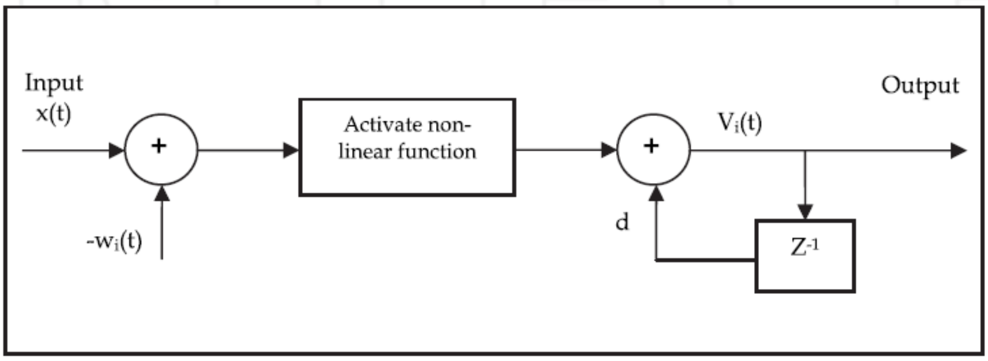


Figure 3.7: TKM network diagram (Sá et al. 2011)

In TKM according to Guimarães et al. (2002), only the magnitude of the output units will be saved, without any information about the direction of the error vector; this can limit the representation of sequences behaviour in the input data. Furthermore, representation capability is restricted just by the input data dimensionality. TKM cannot represent all automata (Strickert & Hammer 2005; Hammer et al. 2004). It has been presented by Koskela et al. (1998) that this network failed to learn temporal sequences of simple synthetic data.

The next version of feedback SOM is designed by moving the leaky integrators from the unit's outputs towards their entry. This network is called Recurrent SOM and it will be introduced in the next section.

3.6.2 Recurrent SOM (RSOM)

The Recurrent SOM (RSOM) was proposed in 1998 by Koskela et al. (Koskela et al. 1998) in order to defeat the TKM limitation. This network was designed by moving the leaky integrators from the output to the input as shown in Figure 3.8. As a result, the network memorises the magnitude and the direction of the error. The output for each unit is:

$$y_j(t) \;=\; \alpha[x(t) \;- W_{hj}(t)\,] + \;(1-\alpha)\,y_j(t-1) \tag{3.32}$$

$y_j(t)$ represents the leaked difference vector for unit *j* at time *t*, and *x(t)* is the input vector; the best matching unit c (BMU) in the RSOM is the one that minimises $y_j(t)$,

$$\mathrm{y_c(t) = min\{\|y_j(t)\|\}} \tag{3.33}$$

and the updated rule are based on minimising the error function E in equation (2.17):

The gradient direction of $E(t)$with respect to $w_j(t)$ is $y_j(t)$ and the weight update rule for $w_j(t)$ to minimise error $E(t)$ is

$$w_j(t+1) = w_j(t) + \gamma h_{jc}(t) y_j(t) \tag{3.34}$$

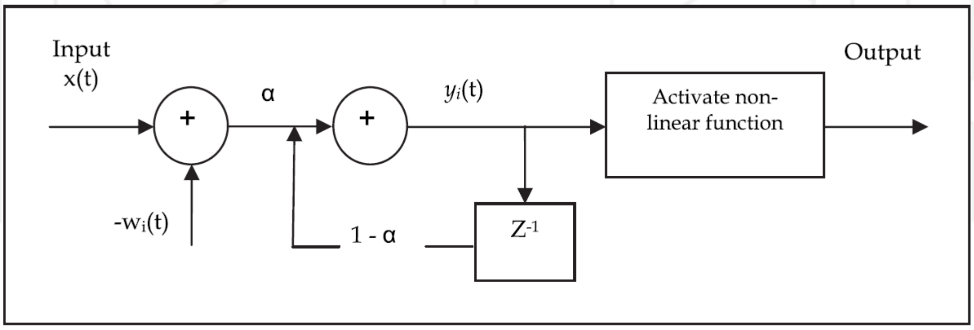


Figure 3.8: RSOM network diagram (Sá et al. 2011)

Varsta et al. (2001) presented a paper in order to compare RSMO and TKM. They investigated the difference between their learning properties. They asserted that the main difference between RSMO and TKM is the leaked integrators computed in RSOM by finding the difference vector as an alternative of using squared norm that has been used in the TKM. Furthermore, the updated rule in RSOM is based on approximates gradient descent to minimise the error function. Since the output value in RSOM is a vector instead of a scalar, the direction of the error can be captured and exploited in the modification of the weights (Koskela et al. 1998; Varsta et al. 2001). The RSOM has shown the ability to capture temporal context in many time series including Laser time series, synthetic, even biomedical signals such as EEG signals (Koskela et al. 1998; Varsta et al. 2001).

3.6.3 The Recursive SOM (RecSOM)

The recursive SOM (RecSOM) was designed by Voegtlin (2002). The RecSOM is a dynamical neural network which involves a set of input units which contain current inputs and some previous inputs (Voegtlin 2002). The architecture of the RecSOM model is shown in Figure 3.9.

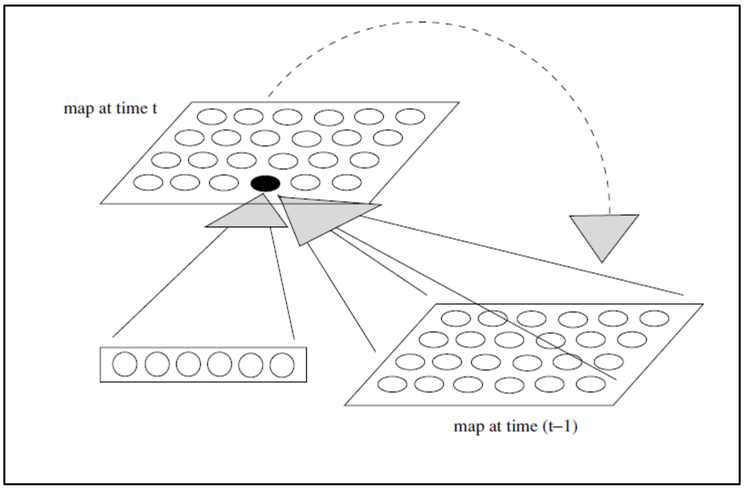


Figure 3.9: Recursive SOM architecture (Voegtlin 2002).

The RecSOM networks involve input neurons x(t) where t refers to time step, and the neurons in the map are referred to as $j \in \{1,2,...,N_h\}$; each neuron in the map has two weight vectors connected with it:

– W_{hji} linked with an N dimensional input x_i (t) feeding the network at time t. and $i=\{1,..,N_I\}$.

– wz_jlinked with the context units

The context units hold map activations $y_j(t - 1)$ from the previous time step.

The output of a unit i in the map at time t is calculated as

$$y_j(t) \; = \; exp(D_{hj}(t)) \tag{3.35}$$

$$D_{hj}(t) = \alpha \cdot \| \; x(t) - W_{hj}(t) \; \| + \beta \cdot \| \; y(t-1) - W_{z_j} \; \| \tag{3.36}$$

let α > 0 and β > 0 be model parameters which determine the influence of the input and the context, and norm is Euclidean distance. The best matching unit (BMU) is defined as the unit that minimises $D_{hj}(t\;)$, where c is an index of the best matching unit at time t,

$$c \; = \; argmin\left(D_{hj}(t)\right) \tag{3.37}$$

After finding c, both weights W_{hj} and W_{z_j} vectors can be updated using the same form of SOM learning rule:

$$\Delta W_{hj} = \gamma \cdot h_{jc} \cdot (x_i(t) - W_{hj}) \quad (3.38)$$

$$\Delta W_{zj} = \gamma \cdot h_{jc} \cdot (y(t-1) - W_{zj}) \quad (3.39)$$

and $0 < \gamma < 1$ is the learning rate. Neighbourhood function h_{jc} is a Gaussian function of the Euclidean distance *d(j, c)* between units *j* and *c* in the map with σ Gaussian width:

$$h_{jc} = exp(-\frac{d(j,c)^2}{\sigma^2}) \quad (3.40)$$

RecSOM is quite demanding because the RecSOM has higher storage capacity compared with other recurrent SOM models (Hammer et al., 2004). This is because temporal context is represented by the activation of the entire map in the previous time step (Strickert & Hammer 2005). The storage capacity can be enlarged by using more neurons in the map. This is because the RecSOM exceeds the simple local recurrence of leaky integrators of other models (Mourik 2006). As a result, the RecSOM can model very complex dynamic systems compared with other feedback SOM methods. As Hammer claimed, the RecSOM can characterise much richer dynamical behaviour (Hammer et al. 2004). Furthermore, the recursive SOM has shown its ability to categorise temporal inputs (Micheli 2003). In this research work, the RecSOM will be considered. The RecSOM training algorithm will be used to train the proposed network.

The RecSOM has been investigated by a number of researchers. It has been proven that RecSOMs are capable of dealing with sequential data with significantly superior memory depth and topography preservation (Mourik 2006). Mourik used the RecSOM to create a Markovian map in three types of sequences data: stochastic automaton, laser data, and natural language (Mourik 2006). It has been used to learn complex sequence patterns such as natural language text (Tino et al. 2005). Furthermore, Hunag et al. (2009) have used the RecSOM in order to recognise human actions. They used the network to learn adapted representations of temporal context associated with human action sequences which were encoded to the network. The data was a Weizmann human action data set. The result in this paper has been compared with other papers that have used the same data set. From their result, it was demonstrated that the RecSOM achieved promising results in the average recognition rate: 96.5%. Their experiments have demonstrated the effectiveness of the RecSOM for dimensionality reduction, clustering, and context learning in human action recognition (Huang & Wu 2009). Farkas and Crocker (2006) designed a network combining the RecSOM and Sequential Activation Retention and Decay Network (SardNet), which they called RecSOMsard. This network was applied to word prediction tasks. They evaluated the performance of the network by comparing its performance with various prediction models. Their result demonstrated that RecSOM produced better accuracy and faster training when used to learn data with a temporal structure (Farkas & Crocker 2006).

3.7 The Application of Recurrent Neural Networks

RNNs have been used in a number of interesting applications including pattern classification, such as automated biomedical signals classification which uses Elman recurrent networks (Übeyli & Übeyli 2008), time series prediction (Giles et al. 2001; Khoa et al. 2006) and adaptive noise filtering. Furthermore, Dijk (1999) indicated that a dynamic neural network can learn to identify the dynamic pattern of a phoneme (Dijk 1999). He demonstrated that a recurrent neural network can perform better recognition of a voice than a static neural network. A number of studies have asserted that RNNs have the ability to analyse and predict time series (Giles et al. 2001; Silva et al. 2010; Hüsken & Stagge 2003; Yao et al. 2013; Rhaman & Endo 2008; Williams & Zipser 1989; Kremer 1995). Furthermore, it has been used in financial time series forecasting (Tenti 1996; Hussain et al. 2006; Ghazali et al. 2010; Ghazali et al. 2009).This is because they can represent time dependencies in time series data better than a feed-forward neural network. In addition, a number of unsupervised recurrent neural networks have been used for sequential data (Giles et al. 2001; Swarup et al. 2005). For example, Giles et al. (2001) developed a neural network that uses a self-organising map and grammatical inference with recurrent neural networks in order to address difficulties with non-stationary prediction. Their network has implemented the prediction of daily foreign exchange rates. For experimental results, their network has achieved predictive accuracy of up to approximately 46% compared to MLP network. Furthermore, unsupervised recurrent neural networks can deal with difficult natural language-processing problems of position variant recognition (Mcqueen et al. 2003). RNNs have been proved to be powerful tools for modelling biological signals such as EEG signals (E. Forney & Anderson 2011). Furthermore, it has been used to classify the biomedical signals, which is been addressed in chapter 6.

3.8 Chapter Summary

Recurrent neural networks have received a lot of attention from the scientific society. They have been used in many applications such as speech recognition, forecasting, language understanding, medical diagnosis etc. Recurrent neural networks have proved their ability to deal with temporal processing, e.g. time series prediction, system identification, temporal pattern recognition and classification. They are able to learn the processing of sequential data. They have been recommended for use in time series analysis. The next chapter will introduce a novel dynamic neural network architecture developed during the course of this PhD study.

CHAPTER 4 DYNAMIC SELF-ORGANISED NEURAL NETWORKS INSPIRED BY THE IMMUNE ALGORITHM

4.1 Introduction

In this chapter, novel neural network architectures are proposed and presented. The proposed novel neural network is developed by incorporating feedback connections into the structure of a self-organised network inspired by the immune algorithm (SONIA) network. The self-organised network inspired by the immune algorithm (SONIA) is a type of feed-forward neural architecture developed by Widyanto et al. in 2005 (Widyanto et al. 2005). This network has been designed to improve the recognition and generalisation ability of the back-propagation neural network.

In order to apply the dynamic link to the SONIA network, the partially recurrent networks have been used in this research work. This type of recurrent neural network has feed-forward links as well as a selected set of feedback links, as explained in Chapter 3. The feedback connection provides a memory to the network that will help the network to remember information from the past without complicating the learning excessively. Two different types of partially recurrent neural network topology have been utilised in order to develop novel networks. The first type is dynamic DSMIA based on the Jordan recurrent neural network where the feedback links will receive data back from the output layer. The second type is dynamic DSIA based on the Elman recurrent neural network where recurrent links are expected to receive information from the hidden layer. In the next sections the SONIA network and the two proposed networks are presented. The main motivation of these networks is to provide memory for the feed-forward self-organised network inspired by the immune algorithm.

4.2 Artificial immune systems

The main task of a biological immune system is to defend our body from infectious agents (such as viruses or bacteria) usually known as *pathogens*. This task involves a great pattern recognition ability that enables the immune system to define the unknown cells coming into the human body. The biological immune system has many properties that make it suitable for applying in the computational field such as self-organisation, memory, pattern recognition, anomaly detection and adaptive system. The adaptive property in the immune system allows it to fight against any intruder that the innate system cannot remove and the immune system can, and also it can remember past incidents. The immune system consists of cells which are called B and T cells. These cells help to recognize and destroy specific substances, these cells are able to neutralise a predefined set of antigens. The antigen can be either a part of cells or an intruder of the organism itself. The immune system can be adaptive and this by extracted B cells in order to generate new antibodies and B cells to remove the infectious agent. (Timmis 2001).

These properties are receiving a great deal of attention from a scientific perspective (Read, Andrews & Timmis 2011). Therefore, the Artificial Immune System has been designed to inspire the biological immune system. The artificial immune system (AIS) was first designed by Timmis (2001). The AIS is based on simulating the behaviour and the relation between the cell body and the antigen that been created by the immune system. It has been developed during the last decade and has been applied in different applications such as clustering, data mining and optimisation. There are different types of AIS algorithms and most of them have been developed based on three immunological theories which are clonal selection, negative selection and immune networks (Timmis, 2001). All AIS algorithms simulate the behaviour and properties of immunological cells, specifically B cells and T cells. Different of AIS algorithms have deferent applications. They have been used for optimization process (Widyanto et al. 2005, Freschi et al. 2006, Jiao et al. 2008).

4.3 Self-organised Network Inspired by the Immune Algorithm (SONIA)

The SONIA network is a single hidden layer neural network, which uses a self-organising hidden layer inspired by the immune and back-propagation algorithms for the training of the output layer. The immune algorithm is simulated as the natural immune system, which is based on the relationship between its components which involve antigens and cells; this is called recognition ball (RB). The recognition ball in the immune system consists of a single epitope and many paratopes where the epitope is attached to the *B* cell, and paratopes are attached to antigens (Hussain & Al-jumeily, 2007). The *B* cell here will represent several antigens. Biologically, a *B* cell can be created and mutated to produce a diverse set of antibodies in order to remove and fight the viruses attacking the body (Shen et al. 2008).

Thus, the immune system can allow its components to change and learn patterns by changing the strength of connections between individual components. The inspiration of the immune system in the self-organised neural network will serve as a hidden unit created in the back-propagation network. For the SONIA network, the input units are called antigens and the hidden units are considered as a recognition ball (RB) of the immune system. The recognition ball is used to create hidden units. The relation between the antigens and the RB is based on the definition of local pattern relationships between input vectors and hidden nodes. These relationships help SONIA to easily recognise and define the input data's local characteristics, which increases the network's ability to recognise patterns. In SONIA, the mutated hidden nodes are designed to deal with unknown data, which is test data, to develop the generalisation ability of the network. In what follows a brief description of the structure of the SONIA network will be presented.

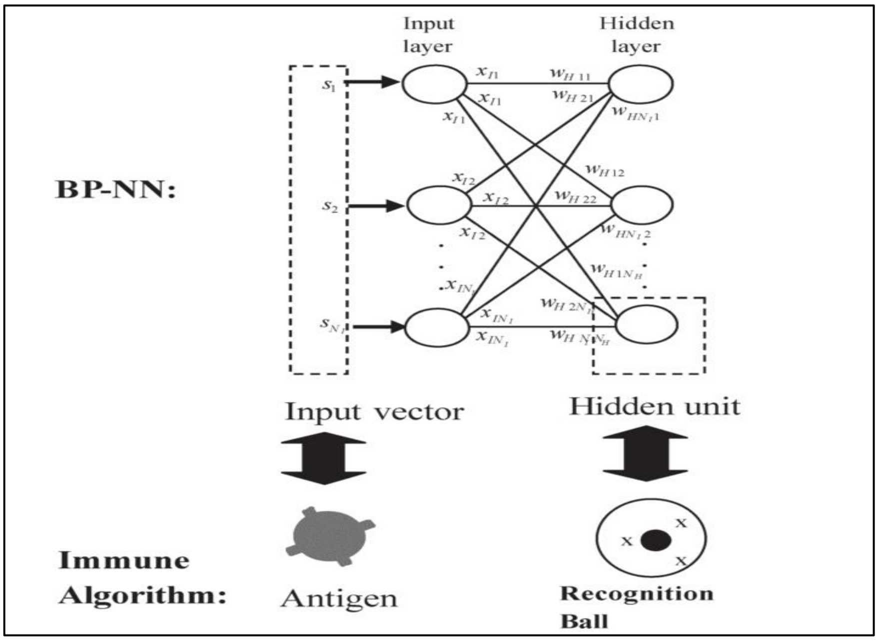


Figure 4.1: The structure of the proposed SONIA network (Widyanto et al. 2005)

Figure 4.1 shows the structure of the SONIA network. The network consists of a single hidden layer, which uses a self-organising hidden layer inspired by the immune algorithm and the back-propagation algorithm for the training of the output layer. In Figure 4.1, the input vector is represented as an antigen and the hidden unit is considered as a recognition ball. The SONIA network consists of three layers; for example, consider that the first layer has a number of input units *{1,….., N_I}*, the self-organised hidden layer has a number of hidden units *{1,…., N_h }*, and the output layer has a number of output units *{1,…, N_o}*.

The input x_i, *{i=1,….., N_I}* is normalised between 0 and 1, and the output of the first layer is $U_i \in [0,1]$; this will be fed as an input into the hidden layers. X_{Hj} is the output of the hidden layer which is computed by the Euclidean distance, as shown in the following equation:

$$D_{hj}(t) = \sqrt{\sum_{i=1}^{NI}(W_{hji} - U_i(t))^2} \qquad (i=1,.....,N_I, j=1,...., N_h) \tag{4.1}$$

The output of the hidden layer is calculated as follows:

$$X_{Hj}(t) = f_{ht}(D_{hj}(t)) \; (j=1,...., N_h) \tag{4.2}$$

D_{hj} refers to the strength of the connection between the i^{th} input units to j^{th} hidden units, and f_{ht} is a hyperbolic tangent sigmoid function in equation (4.2). This output will be used as the input for the output layers. The output can be computed as:

$$y_k(t) = f_{ot}(\sum_{j=1}^{N_h} W_{ojk} \, X_{Hj}(t) + \theta_{ok}) \tag{4.3}$$

W_{ojk} refers to the strength connection between the j^{th} hidden unit and k^{th} output units of the output layers. θ_{ok} is a bias of the k^{th} output units and f_{ot} is a nonlinear activation function which is a log sigmoid function. The overall aim of this training is to minimise the cost function that is:

$$e(t) = d(t) - y(t) \tag{4.4}$$

$$E(t) = \frac{1}{2}\sum e(t)^2 \tag{4.5}$$

Where *d(t)* and *y(t) are* the target and the network output at time *t*, respectively. Minimising the error value *E* is performed by updating the weights in the hidden and the output layers. The W_{ojk} and θ_{ok} which correspond to the output layer are updated by the back-propagation algorithm.

$$\Delta w_{ojk}(t) = -\gamma \frac{\partial E(t)}{\partial w_{ojk}(t)} \tag{4.6}$$

$$\Delta \theta_{ok}(t) = -\gamma \frac{\partial E(t)}{\partial \theta_{ojk}(t)} \tag{4.7}$$

4.3.1 *B* cell construction based on hidden unit creation

The weights $W_{\hbar ji}$ in the hidden layers are updated using *B* cell creation (Steurer 1993),where the hidden unit is considered as a recognition ball in the immune algorithm. In the initialisation procedure, the first hidden unit ($t1$, w_{h1}) is generated with $t1 = 0$, and w_{h1} is taken arbitrarily from the input vector. The procedure of the immune algorithm is used to create the hidden unit (Timmis 2001). This procedure will be repeated until all inputs have found their corresponding hidden unit (Widyanto et al. 2005):

For m=1 until N_I which is the number of inputs repeat the following procedure:

1. For $j=1$ to N_I calculate the Euclidean distance between m^{th} input and the centroid of the j^{th} hidden unit $j=\{1,\ldots,N_h\}$ by:

$$D_{\hbar j_{\hbar j}}(t) = \alpha\sqrt{\sum_{i=1}^{N_i}(u_i(t) - W_{\hbar ji})^2} \tag{4.8}$$

   Where $u_i(t)$ is the i^{th} input unit of the input vector and $W_{\hbar ji}$.

2. Find the short distance D_c

$$c(t) = argminD_{\hbar j}(t) \tag{4.9}$$

3. Compare the D_c; if it is below simulation level, s_1 where $s_1=[0\ 1]$, then the input has found its corresponding hidden unit. Then update the following parameters:
   $t_c=t_c+1$

$$W_{hji}(t+1) = W_{hji}(t) + \gamma D_c(t) \tag{4.10}$$

Where γ is the learning rate.

4. If the shortest is bigger than the stimulation level, s_1 adjust the following:

$$N_h=N_h+1 \tag{4.11}$$

Then generate a new hidden unit, and set the value of t_c to 0 and m to 1, then go to step 1.

The SONIA network was first used for financial time series prediction by Hussain et al.(Hussain & Al-jumeily 2007; Mahdi et al. 2009), and physical time series (Mahdi, Hussain, &

Al-Jumeily, 2010). Experimental results confirmed that the SONIA network could be applied successfully in financial time series prediction.

4.4 The Dynamic Self-organised Multilayer network Inspired by the Immune Algorithm

In this section, the novel neural network architecture will be discussed. The Dynamic Self-organised Multilayer network Inspired by the Immune Algorithm incorporates two different architectures based on the Elman and the Jordan RNN, where the reconnection on the Jordan network occurs from the output layer to the context unit, while in the Elman network the connection takes place from the hidden layer to the context unit. The structure of the Jordan neural network has been employed in this thesis to deal with prediction. In contrast, the Elman recurrent neural network has been used to design the second proposed network in order to deal with the classification task. The next two sections will introduce these networks and their mathematical functions.

4.4.1 The Proposed Dynamic DSMIA based on the Jordan Recurrent Neural Network

In this section, the dynamic self-organised network inspired by the immune algorithm based on the Jordan recurrent neural architecture (DSMIA) is described. The DSMIA network has a recurrent link from the output layer. The main motivation of the proposed DSMIA is to predict time series. Generally, it works by passing times series as inputs and the target is the next sequence. So, the network output consists of future values (forecast output), which will be the next sequence.

4.4.1.1 Properties and Network Structure of the DSMIA

In time series data, the observation at a particular point in time, as cited by Mozer et al. (1994), is based not only on the current inputs, but also on the entire history of the data set, and sufficient memory to store previous behaviour is highly recommended (Mozer 1994). Accordingly, the recurrent link must enrich the performance of this DSMIA model by having a “memory” of past behaviour distributed to the network through temporal context units. The proposed network uses both input units and context units, where the context units hold a copy from the network’s previous output. The use of recurrent connection on the proposed network will allow the network outputs to depend on the initial values of external inputs, as well as on the entire history state of inputs. Consequently, this will enhance the proposed DSMIA capability to deal with time-related patterns. The recurrent links of the proposed network are designed as a recursive unsupervised neural network (Voegtlin 2002). In the proposed network, the structure of the recurrent connection is the same architecture as the Jordan network (Jordan 1990), in which the output of the network is fed back to the inputs through the context nodes. As result, the model on the DSMIA network will be built based on the past time series inputs and its own prediction values.

The structure of the Jordan recurrent neural network is illustrated in Figure 3.2. The proposed DSMIA network is illustrated in Figure 4.2. The DSMIA network has three layers: the input layer, the self-organising hidden layer, and the output layer with feedback connections from the output layer to the input layer. The input layer holds copies of the current inputs as well as the previous output produced by the network. This provides the network with memory. As such, the previous behaviour of the network is used as an input affecting current behaviour; the output of the network is fed back to the input through the context units.

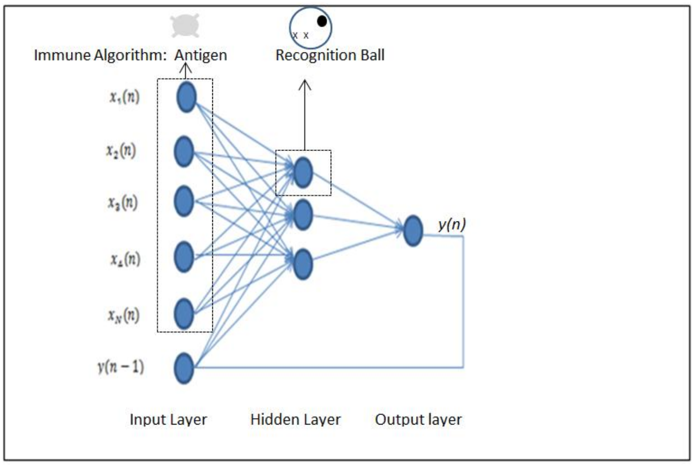


Figure 4.2: The structure of the proposed DSMIA network

4.4.1.2 The Dynamic Equations for DSMIA based on the Jordan Network

Suppose that N_I is the number of external inputs *x(t)* to the network, and $y_k(t-1)$ is the output of the unit k at previous time step *(t-1)* with N_o representing the number of outputs. In the proposed *DSMIA*, the overall input to the network will be the component of *x(t)* and $y_k(t-1)$ and the number of inputs of the network is N_I+N_o defined as U where the output of the input layer:

$$U_i(t) = \begin{cases} x_i(t), & i = 1,..,N_I \\ y_i(t-1), & i = 1,..N_o \end{cases} \tag{4.12}$$

The output of the hidden layer is computed as:

$$V_{\hbar j}(t) = \alpha\sqrt{\sum_{i=1}^{N_i}(W_{\hbar ji} - x_i(t))^2} \tag{4.13}$$

$$Z_{\hbar j}(t) = \beta\sqrt{\sum_{k=1}^{N_o}(W_{z_{jk}} - y_k(t-1))^2} \tag{4.14}$$

$$D_{\hbar j}(t) = V_{\hbar j}(t) + Z_{\hbar j}(t) \tag{4.15}$$

$$X_{Hj}(t) = f_{\hbar t}\left(D_{\hbar j}(t)\right) \tag{4.16}$$

Where $f_{\hbar t}$ is a nonlinear activation function, N_i is the number of external inputs, N_o is the number of output units, $w_{\hbar ji}$ is the weight corresponding to the external input while $W_{z_{jk}}$ is the weight corresponding to the previous output, and t is the current time step, while α , β are elected parameters with $0<\alpha$ and $0 < \beta$.

$$y_k(t) = f_{ot}\left(\sum_{j=1}^{NH} X_{Hj}(t) * w_{ojk}\right) \tag{4.17}$$

Where f_{ot} is a nonlinear activation function, which is the sigmoid function and w_{ojk} is the weight corresponding to output units.

4.4.1.3 Learning Algorithm

The first layer of the DSMIA is a self-organised hidden layer trained similarly to the recursive self-organised map (RecSOM) (Voegtlin 2002). In this case, the training rule for updating the weights of the context nodes $W_{z_{jk}}$ are also updated in the same way as the weights of the external inputs $W_{\hbar ji}$. This is done by first finding D, which is the distance between the input units and the centroid of the j^{th} hidden units:

$$D_{hj}(t) = \alpha\sqrt{\sum_{i=1}^{N_i}(X_i(t) - W_{\hbar ji}(t))^2} + \beta\sqrt{\sum_{k=1}^{N_o}(y_k(t-1) - W_{z_{jk}}(t))^2} \tag{4.18}$$

From $D_{\hbar j}(n)$, the position of the closest match will be determined as:

$$c(t) = argmin\{D_{hj}(t)\} \tag{4.19}$$

Where *c(t)* minimised $D_{\hbar j}(t)$, it is called the best matching unit (BMU), which is the unit that wins the competition. Then the weight from the external input vector and the context vector are updated as follows:

$$W_{hji}(t+1) = W_{hji}(t) + \gamma D_c(t) \tag{4.20}$$

$$W_{z_{jk}}(t+1) = W_{z_{jk}}(t) + \gamma D_c(t) \tag{4.21}$$

Where $W_{z_{jk}}$ is the weight of the previous output and $W_{\hbar ji}$ is the weight for the external inputs, and γ is the learning rate that is updated during the epochs.

4.4.2 The Dynamic Self-organised network Inspired by the Immune Algorithm (DSIA) based on the Elman Recurrent Network

In this section, the dynamic self-organised network inspired by the immune algorithm based on the Elman recurrent neural architecture is described. The DSIA network has recurrent links from the hidden layer. The main motivation of the proposed DSIA is for classification tasks. Generally, it works by passing some feature vectors as inputs and the target is the correct class label of these features. So, the network outputs represent the class.

4.4.2.1 Properties and Network Structure of the DSIA based on the Elman Network

The Elman Network architecture as discussed in Chapter 3 has been used for designing the structure of the DSIA. It consists of two parts, feed-forward part and a memory part, which is known as context units in the Elman network. The memory is stored the activation of the hidden units from the previous time step and fed back to the DSIA as additional input on the next time step. The feedback connection in the Elman neural network will enhance the classification performance (Arvind et al. 2010). In the learning phase, the feedback connection provides the network with more information during training. This will help the network to easily identify the important pattern. Furthermore, it will improve the generalisation ability (Hüsken & Stagge 2003).

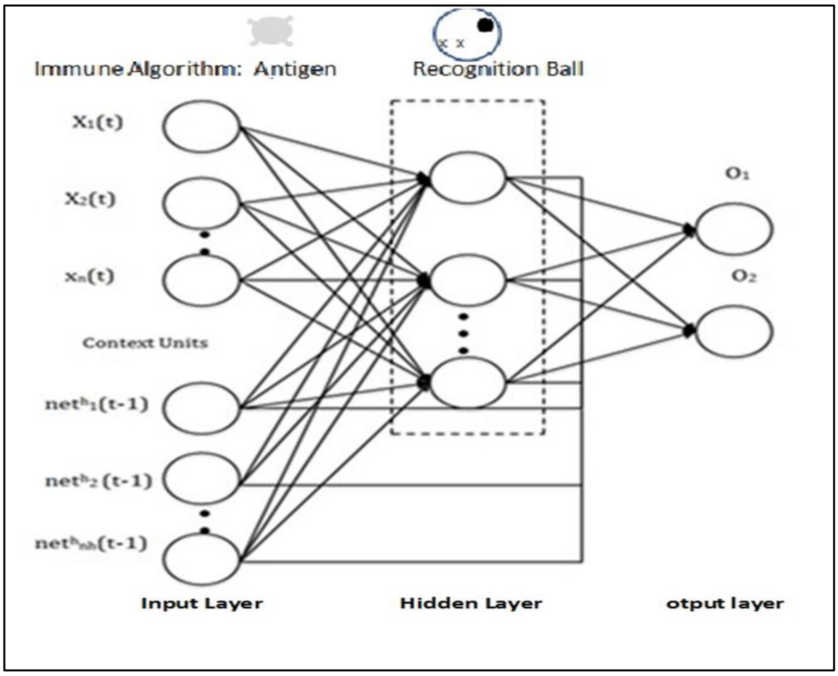


Figure 4.3: The structure of the proposed DSIA network

The proposed DSIA is illustrated in Figure 4.3. The DSIA network has three layers: the input, the self-organising hidden layer, and the output layer with recurrent connections from the hidden layer to the input layer. The input layer holds copies of the current inputs as well as the previous hidden unit produced by the network at previous time step *(t-1).*

This network will be suitable for the classification task; the features that are extracted from the time series signals are fed into the DSIA network as inputs. The output units of the output layer in the DSIA represent the classes of the input. In other words, the number of output units is equal to the number of classes. The output unit with the highest value determines the classification result. The network training procedures are based on reduction of error function, which is measuring by mean squared error (MSE).

4.4.2.2 The Dynamic Equations of DSIA based on the Elman Network

Suppose that N_i is the number of external inputs *x(t)* to the network, and $X_{Hi}(t-1)$ is the activation of the hidden unit *i* at previous time step *(t-1)* with N_h representing the number of hidden units. In the proposed DSIA, the overall input to the network will be the component of *x(t)* and $X_{Hi}(t-1)$, and the number of inputs of the network is N_I+N_h defined as U where the output of the input layer

$$U_i(t) = \begin{cases} x_i(t),\ i = 1, .., N_I \\ X_{Hi}(t-1),\ i = 1, .. N_h \end{cases} \tag{4.22}$$

The output of the hidden layer is computed as:

$$V_{hj}(t) = \alpha\sqrt{\sum_{i=1}^{N_i}(W_{hji} - x_i(t))^2} \tag{4.23}$$

$$Z_{hj}(t) = \beta\sqrt{\sum_{j=1}^{N_h}(W_{z_{jj}} - X_{Hj}(t-1))^2} \tag{4.24}$$

$$D_{hj}(t) = V_{hj}(t) + Z_{hj}(t) \tag{4.25}$$

$$X_{Hj}(t) = f_{ht}\left(D_{hj}(t)\right) \tag{4.26}$$

Where $f_{\hbar t}$ is a nonlinear activation function, N_i is the number of external inputs, N_h is the number of hidden units, $w_{\hbar ji}$ is the weight corresponding to the external input while $W_{z_{jk}}$ is the weight corresponding to the previous hidden unit, and t is the current time step, while α , β are elected parameters with 0<α and $0 < \beta$.

$$y_k(t) = f_{ot}\left(\sum_{j=1}^{N_\hbar} X_{Hj}(t) * w_{ojk}\right) \tag{4.27}$$

Where f_{ot} is the sigmoid function, N_h is the number of hidden units, N_o is the number of output units, and w_{ojk} is the weight corresponding to output units

4.4.2.3 Learning Algorithm

In the case of DSIA, each unit j on the map has two weights, $w_{\hbar ji}$ and $W_{z_j}$where $w_{\hbar ji}$the weights linking the map with input and W_{z_j} is the weights linking context unit which is the output of the hidden layer at the previous time step with the unit on the map.

$$D_{\hbar j}(t) = \sigma(\alpha\left\|x(t) - w_{hji}\right\| + \beta\left\|X_{Hj}(t-1) - W_{z_{jj}}\right\|) \tag{4.28}$$

$$X_{Hj}(t) = f_\hbar(D_{\hbar j}(t)) \tag{4.29}$$

With α >0 and β >0, || || denotes the standard Euclidean distance of vectors and $f_\hbar$ is a bipolar sigmoid function. The best matching unit is defined as the unit that minimised $D_{\hbar j}(t)$

$$c(t) = \text{argmin}\{D_{\hbar j}(t)\} \tag{4.30}$$

Then the learning rule is applied to update the weights of input units and context units:

$$W_{\hbar j}(t+1) = W_{\hbar j}(t) + \gamma D_c(t) \tag{4.31}$$

$$W_{z_j}(t+1) = W_{z_j}(t) + \gamma D_c(t) \tag{4.32}$$

Where W_{z_j} is the weight of the previous hidden unit and $W_{\hbar ji}$ is the weight for the external inputs, and γ is the learning rate which is updated during the epochs.

4.5 Chapter Summary

This chapter has introduced dynamic self-organised networks inspired by the immune algorithm which were presented as an extension of the feed-forward self-organised network inspired by the immune algorithm. In order to represent the dynamic networks, two recurrent architectures were used. The first network, DSMIA was extended by adding a feedback connection from the output layer. Using recurrent links from the hidden layer extended the second network, which is called DSIA. The properties of these two architectures and their mathematical processes have been presented. The next two chapters will illustrate the implementation of these networks with two different scenarios.

CHAPTER 5 APPLICATION FOR FINANCIAL TIME SERIES FORECASTING

5.1 Introduction

Time series analysis is a fundamental subject that has been addressed widely in different fields. It has been exploited in biomedical, economic and industrial data as well as financial time series. The analysis of financial time series is of primary importance in the economic world. This chapter aims to provide an overview of time series analysis. This chapter will deal with forecasting of financial time series and will involve some background and issues related to financial time series. Furthermore, it will address the application of neural network and traditional forecasting methods, specifically with regard to the forecasting of financial time series.

5.2 Time Series Data

A time series is a collection of observations of a particular problem measured during a period of time. In theory, it is known as a sequence of variables ordered in time (Schwaerzel & Bylander 2006). Mathematically, for any given system, a time series can be referred to as $x(t)$ or $\{x(t), t \in T\}$, and it contains two variables; the first one is the time variables *(t)* while the second one is the observation variables $x(t)$, where x can be a value that varies continuously with t, such as the temperature and stock market, etc.

In reality, there are many motivations for conducting time series analysis and modelling. It has recently gained much attention from scientists and researchers, whose interest has led to different types of time series in different applications worldwide. In industrial applications, time series can be used to monitor industrial processes (Roverso 2000; Mirea & Marcu 2002). Time series analysis also has important applications in economics. The main motivation of analysing financial time series is to gain the ability to identify and understand the internal structure that creates the data in time series. In other words, as Herrera (1999) asserted, it attempts to explore the underlying properties of sequences of observations taken from a system under examination. In addition, it helps find the optimal model to fit the time series data and apply this model to predict the future observations of data based on past data series (Ghazali et al. 2009). For example, financial market prediction by computations of the next value of trade sales each month (Widrow et al. 1994; Shachmurove & Witkowska 2000; Yümlü et al. 2005).

The main aspect of time series is actually that observation values are not created independently or ordered randomly; the data in time series are representing sequences of measurements arranged according to time intervals (Michael Falk, Frank Marohn, Rene Michel et al. 2011). Therefore, time variables are very important in time series analysis because they show when the measurements were recorded. Hence, Herrera (1999) asserted that the time values must be stored along with observations that have been taken, and they should be used with the time series as a second piece of information. Therefore, the model that will be used to fit and analyse the time series data must have the ability to process the temporal pattern of the time series.

Two main features characterise time series data: the stationary and non-stationary concepts. It is very important to identify these two concepts before time series analysis, and this will help to find the best mathematical model to deal with this type of data. The simplest way to observe stationary and non-stationary data is the plotting of the observations.

The concept of stationary in time series means that the probability distribution between data does not change when shifted in time. Hence, the statistical properties (e.g. mean, variance and autocorrelation) of the data are stable with respect to time (Pedersen 1997; Haykin 1998), such as climate oscillations (Mengistu et al. 2013).

In mathematics, stationary can be defined as follows, when the distribution of $(xt_1, \ldots, xt_n)$ is the same as the distribution of $(xt_{1+k}, \ldots, xt_{n+k})$ where $t_1, \ldots, t_n$ is refers to time step, and k is an integer (Aamodt 2010). The behaviour of any intervals in this series is similar to one another, even if the segments have been taken from the beginnings of the time series or the ends (Pollock 1987). Therefore, this type of time series is easy to model.

Non-stationary characterises another type of time series. It means that parameters of the information (e.g. mean and variance) of the data always change over time (Haykin 1998; Ghazali et al. 2009). Therefore, behaviours of the signals are changing from one interval to the next. Most real-world time series are non-stationary, such as financial time series data (Giles et al. 2001; Ghazali et al. 2009) or biomedical signals (Chendeb et al. 2010; Arafat 2003; Bazregar & Mahdinejad 2013). Non-stationary time series are difficult to deal with.

However, some models require the application of a pre-processed method in order to smooth out the noise and reduce the trend of the non-stationary data. Therefore, they can be transferred from non-stationary to stationary (Herrera 1999; Thomason 1999; Box et al. 1994).

5.3 Time Series Analysis

In the literature, there are a number of methods for analysing time series. In this thesis, two types of time series analysing tasks will be discussed: classification and prediction. These two tasks are quite different in their aims and processes. The target data of such tasks refer to different types of information. For classification tasks, the targeted data show the classes to which the data belong. Therefore, the task is to identify the class of signal. Furthermore, the values in the classification represent attributes or features. On the other hand, in the phase of time series prediction, the task is to predict future values of a signal from its past values. Therefore, the target data in this task are the future values. In this chapter the prediction task will be presented. The classification task will be presented in the next chapter.

5.3.1 Time Series Prediction

Recently, the ability to predict future observations has received great attention from various research disciplines, such as medicine, economics, speech recognition, etc. Researchers are interested in the ability to make well-defined decisions about certain situations (Ghazali et al. 2009; Chen & Leung 2005). These decisions must be created based on an understanding of their consequences. This can be done by predicting the consequences of decisions made or, in other words, predicting the future (Herrera 1999). Therefore, time series prediction is considered an important task for time series analysis. It is defined as the estimation of future values of the series based on some past observed values (Bishop 1995). The predictor is able to discover approximate functions between the input and the output values (Petridis et al. 2001; Yümlü et al. 2005). This function can be used to forecast the future value. The predictor can be achieved by using linear and nonlinear mathematical models.

During the last few decades, the importance of investigating and exploring times series data has been widely recognised in financial and other fields. This has led to the appearance of a number of prediction models. All predictors exist in order to discover functions or relationships between the past observations and then use these functions to predict the time series in the future (Herrera 1999). The prediction can be expressed mathematically as follows. Let *x(t)* represent a time series where $t=0,1,....N_t$. In order to predict the value of *x* at time *t+1:*

$$\mathrm{x}(t+1) = f\big(x(t);\ x(t+\tau), \dots\dots x(t+(d-1)\tau)\big) \tag{5.1}$$

Where d<Nt are sampled data values, *d* is time steps back from *t*, and τ refers to a delay time. The embedding dimension of the time series denotes the number of degrees of freedom. However, the differences between these models are based on the techniques that have been used to approximate the unknown function *f*.

In the early days, most of the prediction time series models were linear; these include the Autoregressive (AR) model, which is based on the simple principle that time series data are highly correlated; therefore, previous observations of the data series are used to predict future observations (Box et al. 1994). The AR model of a random process *x(t)* in time *t* is defined by the following:

$$x(t) = \sum_i^p a(i).x(t-i) + \varepsilon(t) \tag{5.2}$$

Where *a* are the coefficients of AR, *m* is the order of the model, *ε(t)* is random error.

The moving average (MA) measures the mean of a set of previous observations in the time series in which the mean values are used to predict future observations of the time series (Dunis & Williams 2003). The MA model of a random process *x(t)* in time *t* is defined by the following:

$$X(t)=\mu \sum_i^m a(i)\varepsilon(t-i) \tag{5.3}$$

Where μ are are the mean values of the series, and *m* are the order of the model MA.

The Autoregressive-moving-average model (ARMA) is based on both the AR and MA (Box et al. 1994). The type of time series assumed in the previous models is stationary. The random process *x(t)* is an autoregressive moving average process (ARMA) *(p,q)* in time *t* and is defined by the following:

$$x(t) = \sum_i^p a(i).x(t-i) + \varepsilon(t) + \sum_i^q a(i)\varepsilon(t-i) \tag{5.4}$$

Where p q, denote to p autoregressive and q moving average. For non-stationary time series data, the autoregressive integrated moving average (ARIMA) was developed by Box and Jenkins (1976) and proposed to fit this type of time series. This model is an extended model of ARMA, consisting of autoregressive (AR), integrated (I) and moving average (MA) parts (Box et al. 1994). In ARIMA, the time series must be integrated before the forecasts are created so that the predictions are expressed in values matching the input data (Alnaa & Ahiakpor 2011). The ARIMA model assumes that the data can be stationary after differencing (Sfetsos 2000). It has been considered the basis for time series analysis and has been widely used in financial predictions (Porter-hudak 1990; Ho et al. 2002).

Despite the wide applications and easy implementation of the traditional forecasting models, their ability to understand time series is very limited (Box et al. 1994; Herrera 1999; Faraway & Chatfield 1998) as they suffer certain limitations. They assume that the relations between data in time series are linear, and they are utilised under stationary conditions (Herrera 1999; Box et al. 1994; Faraway & Chatfield 1998). In practice, relations in most time series are complex and nonlinear in nature (Pedersen 1997; Huang, Lai and Nakamori 2004). For example, time series of stock markets are complex, nonlinear, dynamic and chaotic. Traditional linear prediction methods are very poor at capturing the optimal prediction functions and therefore need to be improved (Pedersen 1997). Furthermore, there is an increasing requirement to find more robust and powerful prediction methods that can overcome the traditional prediction models' limitations.

Consequently, the use of nonlinear flexible methods as predictors is in great demand. These extend the power of time series analysis to cover systems with nonlinear behaviour (Herrera 1999). As shown in Chapter 2, models based on ANNs have been known as nonlinear flexible models that are able to model any relationship and behaviour of the time series (Somers and Casal 2009; Castiglione 2008; Mirea and Marcu 2002; Zhang et al. 2001). Therefore, ANNs can overcome the limitation of the statistical methods. On the other hand, the employment of adaptive learning models that are able to generalise the model to predict new data, such as neural network models, can thereby increase the probability of producing correct predictions (Herrera 1999). Moreover, in real-life research, no information is available on the system that generates time series data sets (Castiglione 2008). Therefore, the predictor model must have the ability to discover the correct internal presentation and capture the hidden pattern of time series (Hill, T. & Lewicki 2007; Ho et al. 2002; Chen & Leung 2005; Pedersen 1997).

As a result of these properties of ANNs, they have been taken into account in forecasting economical and other types of time series. In order to deal with time series problems, ANNs need to be modified in such a way that the input data presented to the neural network are drawn from a number of sequential previous inputs, rather than single inputs, and the rest of the network is implemented and learned just like other problems based on the adaptation of weights to minimise the forecasting error (Hill, T. & Lewicki 2007). The ANNs most widely utilised as time series predictors are MLP (Cao & Tay 2003b; Rout et al. 2012), and Elman neural network (ERNN) (Zhang et al. 2013; Forney & Anderson 2011; Elman 1990). A number of studies have proved that using ANN in modelling and predicting time series can produce acceptable results. The next section will introduce one type of time series: financial signals.

5.4 Financial Time Series

The analysing of financial time series has an economic importance. It is a promising and crucial task for any future investment used for making decisions in different areas, such as businesses and financial institutions (Kamruzzaman, 2004). Financial time series involve different time scales such as intraday (high frequency), hourly, daily, weekly, monthly, or yearly. The distance between variables in financial time series is influenced by real economic activity (Espinoza et al. 2009). The effect of this activity has been represented by a mixture of high and low values in financial time series charts (Leondes 2010). Thus, the prediction aims to forecast these activities. The variations in financial time series could be trend, cyclic, periodic and day-to-day variations. The trend represents a recognisable long-term regular variation in a time series. The cyclic and periodic variations are monitored either by the business cycles in the economy or by seasonal patterns. Short-term and day-to-day variations seem to be random and difficult to forecast; however, they are usually considered as a resource of the financial trading gain and loss (Leondes 2010).

Financial data analysis usually provides the fundamental basis for decision models (Kamruzzaman & Sarker 2003) to achieve good returns, which is the first and the most important factor for any investor. This can help to improve companies' strategies and decrease the risk of potentially high losses (Krollner 2011). Furthermore, it can help investors to cover the potential market risk to establish some techniques to progress the quality of financial decisions. Different approaches have been utilised to study and analyse the financial data, which include: technical analysis and fundamental analysis.

Technical analysis is a method based on studying the change in price movement of a stock and using this study to forecast future stock price movements. So, it can predict the right time to buy or sell a stock. Each type of price movement would be translated as a time series signal to either buy or sell a stock. The information about a stock signal can be presented on charts that contain technical data like price, volume, and highest and lowest prices per trading. It is known as univariate signals. This method is a very common method used to predict price. It is very simple to use for forecasting future values (Aamodt 2010; Huang, Lai, and Nakamori 2004).

On the other hand, fundamental analysis attempts to determine the factors that affect price movement directions. It studies issues that can affect the performance of the company's business and its future prospects. It is considered to be a physical study of a company in terms of its product sales, manpower, quality, infrastructure, etc. Therefore, it demonstrates the company's financial statement and current economic activity (Ghazali 2007). The fundamental data are considered as multivariate signals. They include information about the current market statement, in addition to other information referred to (Hellstrom & Holmstrom 1997). This type of data is more difficult to deal with as it involves a more complete picture of the financial environment (Aamodt 2010; Huang, Lai and Nakamori 2004).

5.4.1 Financial Time Series Prediction

Financial time series prediction has been widely addressed by many researchers. Its most important feature is enabling administrators to look to the future and thus take precautions, which helps to develop companies more strongly and safely (Ghazali et al. 2009). Furthermore, forecasting helps to show the right decision and route those financial and business institutions must follow to achieve their goals. Prediction can help to improve the design of a company's aims. Thus, financial data prediction has attracted many financial organisations and companies (Hammerstrom 1993; Shachmurove & Witkowska 2000; Deistler 1996) . The forecasting of the stock market is one example of financial time series prediction. The fluctuations of the stock market are produced by complex activity and their moves are translated into a blend of gains and losses that are represented in time series (Ghazali 2007). The aim of stock market forecasting is to forecast the future values of the price. This process will provide essential information about the actual stock price movement direction and its trends. This information will benefit the investors, enabling them to make the right choice about buying/selling strategies (Ristanoski & Bailey 2011); in addition, it will help build a profitable trading strategy. Therefore, the importance of analysing financial time series in the economic world is significantly increasing.

However, financial time series prediction is considered to be a challenging task because of the structure of financial time series data (Castiglione 2008). As explained by Deistler (1996), many financial time series involve complex behaviours. Financial time series involves various characteristics which are listed below:

- Noise: it contains high levels of noise especially when the variation type in financial time series is day-to-day variation (Hussain et al. 2008) due to random external driving forces.
- Incomplete: in any time series, there are usually some missing data.
- Non-stationary: the non-stationary characteristic in financial data indicates that the trends in their means and variances vary over time (Ghazali et al. 2009).
- Nonlinearity: the nonlinearity in the financial data indicates that the relations between some observations are nonlinear (Bansal et al. 1993).
- Outlier's data: this means that there are some values in financial time series that do not appear to be consistent with other values on the same time series. These values are referred to as irregular values.

5.4.2 The Problem with Financial Time Series Prediction

Financial time series modelling and predicting have fundamental importance to several practical areas. Financial time series such as the movements of market prices is not random and predictable (Man-chung et al. 2000). Financial data are naturally dynamic, nonlinear, nonparametric, complex, and chaotic (Tan et al. 2005). Financial time series such as the stock market are facing dramatic changes, as well as rapid information exchange all the time (Lin & Yu, 2009). Hence, the prediction of economic activity in the future is extremely challenging (Ahmadifard et al. 2013). On the other hand, there are several economic factors that complicate the process of predicting change in large financial data sets, including: institutions' performance and policies, general economic conditions such as stock prices of other countries, gross domestic product, bank rate, exchange rates, interest rates, current account, money supply, employment, general economic conditions, commodity price index, bank rate, bank exchange rate, investors' expectations, institutional investors' choices, movements of other stock markets, psychology of investors, etc. (Ahangar et al. 2010; Kurihara 2006; Agrawal et al. 2010). Furthermore, Yao et al. (1996) identify that there are a number of interrelated factors influencing the direction of the price movement on the stock market, besides the economic factors, which are political factors and psychological factors affecting both powerful decision-makers and individuals or consumers. Even traditional economics studies tell us that microeconomic drivers can affect demand consumption patterns and vice versa; thus, forming a very complicated, highly interrelated system is very challenging (Yao et al. 1996; Ghazali 2007). Therefore, a number of difficulties can be faced by researchers when handling time series forecasting (Schwaerzel & Bylander 2006). Hence, the selection of an appropriate model for solving the time series prediction problem has been considered by many scholars and public investors (Zhang et al. 2001). The main purposes of using these models are to discover some rules and hidden information in stock price fluctuation by analysing trading behaviours. Then, these models will be used to accurately predict the future values of the time series. Numerous techniques have been applied to perform financial time series forecasting, as detailed in the next section.

5.4.3 Conventional Prediction Methods for Financial Time Series

The early studies on time series prediction used traditional statistical models. In the literature, there are a variety of linear time series models. The most standard models are described in detail by Box and Jenkins (1978).

In spite of the extensive applications of linear models in financial forecasting, these models suffer from some limitations in capturing some types of economic behaviour such as non-stationary, or economic performance, at specific periods of time (Clements et al. 2004; Kamgaing 2005). These models adopt the idea that the underlying relationship among the past and the future values of a time series is linear, and some of them assume that times series are stationary (Herrera 1999; Cao & Tay 2001; Chen et al. 2006).

However, financial time series show nonlinear characteristics, and specific dynamic behaviour (Dablemont et al. 2007). The volatility in time series is not constant over time (Dunis and Williams, 2002; Yao and Tan, 2000; Hellstrom and Holmstrom, 1998). Thus, from the perspective of forecasting, there is plenty of reason to look at nonlinear prediction models. Clements et al. (2004) published a paper to address the reasons for considering the nonlinearity model for forecasting financial time series. Furthermore, they show that the forecasting ability of linear models is not accepted. It is therefore not surprising that nonlinear methods such as neural networks are receiving an excessive amount of attention in financial time series prediction literature.

5.4.4 Neural Networks in Financial Time Series

A number of different artificial intelligent (AI) methods have been developed and used to overcome these limitations of traditional forecasting methods. Various AI techniques such as artificial neural network architectures (ANNs) have been proven to be extremely successful for predicting nonlinear and non-stationary time series (Kamruzzaman 2004; Dunis & Williams 2002; Castiglione 2008; Lin & Yu 2009; Chen et al. 2006: Tawfik et al.,2014). This section will provide a literature review of using neural networks on financial forecasting.

The first application of a neural network for stock market predictions was established by Kimoto et al. in 1990; they applied a neural network to forecast the Tokyo stock exchange index (Kimoto et al. 1990). Since Kimoto et al.'s initial research, neural networks have been considered in the financial domain. During the last decades, a number of neural network architectures have been studied in this regard. According to Zekic (1998), neural networks have different applications to financial time series analysis, which include classification of stock market, recommendation of trading, predicting price changes of stock indexes, stock price forecasting and modelling the time series of stock markets. ANNs have been used in various tasks such as forecasting the exchange rates between currencies (Walczak 2001; Yao et al. 1996; Giles et al. 2001; Yao & Tan 2000) and forecasting the sign of price increments (Castiglione 2008). Other studies have demonstrated that neural networks can capture the underlying information and rules of the movement in currency exchange rates (Yao & Tan 2000; Chen & Leung 2005; Yao et al. 1996; Ghazali et al. 2009). Dase and Pawar (2010) presented a literature review on using ANNs to forecast world stock markets. ANNs have proved extremely successful in predicting nonlinear and non-stationary time series

(Kamruzzaman 2004; Chen et al. 2006). The most common neural network is the traditional multilayer perceptron neural network with back-propagation learning. A number of studies have compared the performances of traditional forecasting models with neural networks, including Bansal et al. (1993), Yao et al. (1996), Vojinovic et al. (2001) Dunis and William (2002; 2003), Pissarenko (2002), Aryal and Yao-wu (2003), and Kamruzzaman (2003; 2004). Empirical results reported in various studies indicate that the performances of neural networks are better than linear regression techniques using financial evaluation functions or forecasting error measures (Bansal et al. 1993; Ahangar et al. 2010; Deniz and Karım 2013; Dunis 2012; Giovanis 2010). Kamruzzaman et al. (2003) have investigated the performance of three ANN models that aimed to forecast foreign exchange rates; they were the standard MLP network trained with the back-propagation learning algorithm, the scaled conjugate gradient network, and Baysian regression. The experiments attempted to predict six foreign currencies against the Australian dollar. The results also showed that ANNs outperform ARIMA statistical techniques. Aryal and Yao-Wu's (2003) study compared the performance of an MLP network with statistical techniques such as ARIMA to forecast the Chinese construction industry. Their result showed that the cost function root mean square error (RMSE) achieved by the MLP is 49% lower than the ARIMA. Bagherifard et al. (2012) have confirmed that neural networks perform better than the ARIMA model in financial time series prediction in terms of statistical or financial matrices.

Furthermore, ANNs forecast future currency with approximately up to 60% accuracy (Steurer 1993; Walczak 2001), and consequently are being widely used to model the behaviour of financial data and to forecast future values for time series (Widrow et al. 1994; Hammerstrom 1993). Empirical results were reported in Bansal et al. (1993), which indicated that the performance of neural networks is better than linear regression techniques using financial evaluation criteria such as the payoff measure. Furthermore, the performance of ANNs may be better in terms of traditional predictive accuracy such as the proportion of response variation (R2), especially when the relationships within the data set are nonlinear (Bansal et al. 1993). Vojinovic et al. (2001) studied the ability of the Radial Basis function neural network model (RBF) to forecast the daily US/NZ closing exchange rate. Their result proved that the RBF neural network is able to predict the directional change of exchange rate with 76% accuracy, while the traditional linear autoregressive model made only achieved 21% accuracy of predictions. Dunis and Wiliams (2002) compared the performance of a neural network with a number of traditional prediction models including moving average convergence/divergence (MACD) strategy, ARMA, logit estimation and Naïve strategy. Their study was based on predicting the foreign exchange rate on EUR/USD time series. Their results proved the ability of neural network to forecast financial time series. The interesting point in all those works is that different neural network models achieved a very high accuracy in forecasting the market in comparison with the traditional models.

Huang et al. (2004) published a survey of research using ANNs for forecasting foreign exchange rates. Their study also demonstrated the factors that can affect the performance of the neural network. These factors are the selection of data, pre-processing data, selection of the input and output variables, structure of the neural network, and other factors. ANNs can stand noise and chaotic features in time series better than most of the other models (Masters 1993). Therefore, ANNs can deal with the complexities and characteristics of financial time series.

The advantage of using a neural network as a prediction tool is that it includes automatic learning of dependencies on data and it can be adapted to any type of data (Bagherifard et al. 2012). The training process on neural networks can help to learn complex relationships between input and output variables without requiring any assumption of the nature of the relationship. The training algorithm in a neural network is able to discover and model hidden dependencies, then use them to forecast future values (Jayarekha & Nair 2009). In addition, ANNs have the ability of generalisation: their ability to forecast future values of unseen data (testing data set) in which they have not been trained. Utilising a neural network to forecast financial time series such as a stock market has a number of advantages including: saving investors' time when making financial decisions, and aiding them to decrease investment loss and risk caused by stock market fluctuation (Lin & Yu 2009). In 1991, the banks started utilising neural network models to help them make conclusions about the loan application of financial prediction (Pissarenko 2002). Several large investment banks (including Goldman Sachs and Morgan Stanley) currently have departments dealing exclusively with the neural network for their business investment models (Shachmurove & Witkowska 2000). Unfortunately, the most difficult task in time series prediction is to produce high-quality predictions.

Several models and techniques have already been developed to enhance the forecasting ability of neural networks, in order to understand the complex characteristics and natural dynamics in financial time series. Hussain et al. (2008) applied various types of HONNs to predict financial time series, such as functional link neural network (FLNN), Pi-Sigma neural network (PSNN), Polynomial neural network (RPNN) and novel polynomial pipelined network. These neural networks have been used to forecast several sets of exchange rate time series. Their result shows that the novel network can achieve best performance compared with other networks (Hussain et al. 2008). Mahdi et al. (2009) used the self-organising multilayer perceptron inspired by immune algorithm (SMIA) to forecast ten financial time series. Their results show that the SMIA network achieved profit using stationary data. However, when using non-stationary data, the neural network failed to generate any profit. Furthermore, their experiment results for the stationary signals demonstrated that the SMIA achieved better performance on the profit compared to the MLP neural network.

Despite the great applications of the feed-forward neural network, it does not consider the dependence of an individual input on the inputs processed previously (Pacifici 2010). Time series are generated by dynamic complex systems (Herrera 1999). This means that their state changes over time. Prediction is considered a temporal signal processing problem. Therefore, using a temporal model such as a recurrent neural network can achieve the prediction task with the desired accuracy. The recurrent neural network can address the temporal relationship of the inputs by maintaining an internal state (Giles et al. 2001). It is considered an alternative method of time series forecasting through the use of dynamic memory (Zhang et al. 2013). The memory of the previous values generated by the neural network is kept in the context layer; the values are then fed back to the neural network to predict the next values (Pedersen 1997). As result, the predicted values will rely on the current values as well as the previous neural network outputs. This can help the neural network to learn the dynamic information involved in time series.

Zhane et al. (2013) attempted to compare the performance of a seasonal autoregressive integrated moving average (SARIMA) model with three neural networks (MLP, RBFNN and ERNN) in order to forecast typhoid fever incidence in China. Their study proved that ANNs achieved better values in three different cost functions – mean absolute error (MSE), mean absolute percentage error (MAPE), and mean squared error (MSE) – than the traditional prediction model. Ghazali et al. (2009) designed a novel neural network called a dynamic ridge polynomial neural network. They proved that their novel neural network performed very well in financial time series forecasting. Tenti (1996) investigated the forecasting ability of three variations of RNNs to forecast exchange rates. Three types of were Jordan, Elman recurrent network. In the third RNN the context units hold the input layer's contents. He demonstrated that RNNs are useful for forecasting foreign exchange markets. He asserted that these different architectures of RNN are equally applicable to other types of financial time series.

5.5 Chapter Summary

Forecasting financial time series is considered to be one of the most challenging problems in economic societies. A number of studies have shown that neural networks have the greatest ability to forecast time series in general. Different studies have confirmed that neural networks are considered to be promising tools for predicting financial time series. These studies have proven their ability compared to traditional static methods.

The next chapter will introduce another type of time series signal. It is a biomedical signal: uterine Electrohysterogram (EHG) signals. These signals are recorded from women during their pregnancy. The next chapter will show the difficulties and the issues related to this type of times series.

CHAPTER 6 MEDICAL TIME SERIES ANALYSIS

6.1 Introduction

The development of medical information systems is playing an important role in medical societies. The aim of such development is to improve the utilisation of technology in medical institutions (Shortliffe et al. 1990). Expert systems and different Artificial Intelligence techniques for classification have been used to improve decision support tools for medical societies. One of the most widely used tools in classification techniques is the neural network, which has been used to identify different types of diseases and illnesses.

In the literature, many research works have addressed the problem of classification of medical data. The medical issue that is considered in this research investigation is Preterm cases classification from EHG signals.

This chapter concentrates on using EHG classification to identify whether delivery will be preterm or term. It will provide a brief overview of the medical time series analysis, and also present a literature review of using artificial neural networks, specifically the recurrent neural network, for medical time series classification.

6.2 Medical Time Series

Medical time series has received a great deal of attention from medical specialists and scientists, whose concern has led to the existence of different types of time series in the medical area. For example, medical scientists attempt to monitor biological signals from patients using an electroencephalogram (EEG) to record brain activity waves for the purpose of detecting diseases related to brain functions (Subasia et al. 2005; Übeyli 2009; Lehnertz et al. 2001; Szkoła et al. 2011; Shoeb 2009). They also use an electrocardiogram (ECG) to trace electrical activities of the heart in order to detect heart disease (Übeyli 2010; Balli & Palaniappan 2010; Chowdhury et al. 2013). Electromyograms (EMGs) have been used to record the electrical activity of muscle contractions, and these signals have been recorded from various parts of the human body in order to understand the body’s behaviours under normal and pathological conditions (Konrad 2005) (e.g. wrist, uterus, the human forearm, femoris muscle, Gait analysis, etc) (Graupe 2010; Diab, El-Merhie, El-Halabi 2010; Moslem et al. 2012; Miller2008; Chowdhury et al. 2013b; Ilbay et al. 2011). In other cases, patients’ speech signals are recorded in order to recognise specific types of disease such as larynx diseases (Szkoła et al. 2011). Furthermore, time series analyses can be used to assess progresses of the patient's status over time (Machado 1996). The analysis of the medical time series aims to investigate biomedical signals. Such investigations are based on analysing the variation of observations taken with the measurement tools with respect to time (Chiang 2010), and identifying the differences between these signals. These differences can help to classify between normal and abnormal signals to identify diseases.

6.3 Medical Time Series Classification

Classification is the process of finding out which object belongs to which class. There are different types of tasks from an extensive range of areas that can be referred to as classification in the medical field, such as diagnoses and detection. The main task with which this thesis is concerned is the medical diagnosis task. However, classification in medical time series signals is considered a challenging task. The most important and difficult question is how to represent the signals to be classified (Holst 1997; Haykin 1998; Chen et al. 2010). This is considered problematic because variables in medical time series are highly correlated and related to the time domain (Verplancke et al. 2010). Therefore, the requirement to capture both spatial and temporal information of medical signals is highly important. Furthermore, the time series signals operate at a very high dimension, which makes them difficult to classify. It is necessary to utilise some feature extraction methods that transform large time series signals into a small number of features that optimally discriminate that set of signals from other signals and allow a classifier to group the signals with the related class (Miller 2008). For a classifier to be computationally able to distinguish between different signals recorded in different situations or for different people, certain steps must be taken into consideration, including pre-processing signals, and feature extraction.

6.4 Pre-processing of Medical Signals

In reality, most of the recorded signals that represent time series in different applications contain noises. These noises may be due to measurement error or temporary incident, or they may be related to problems with the recording tools (Herrera 1999). For example, the biomedical signal of a patient may be interrupted by the patient's movements or breathing, or by the patient electrocardiogram (ECG) (Herrera 1999; Rabotti 2010; Rosa et al. 2007; Chowdhury et al. 2013a; Rabotti et al. 2008). Hence, the signal characteristics are buried away in the noise (Liu et al. 2005). Therefore, the researchers need to filter these signals to remove or at least reduce these noises in order to measure the true properties of the series (Baghamoradi et al. 2011). Filter techniques play an important role in extracting the signal of interest and removing the unwanted effects of noise. The literature describes a number of filtering methods that have been designed, such as the band-pass filter, which allows specified frequencies to pass. For example, Balli and Palaniappan (2010) have used a band-pass filter to remove high-frequency content and baseline noise on the ECG signals. It has been used on EMG signals to filter different parameters (Rosa et al. 2007; Moslem, Karlsson, et al. 2011; Rabotti et al. 2008; Eswaran et al. 2002; Kavsek and Pajntar 1999; Phinyomark et al. 2012). An EEG signal (Akrami et al. 2005; Übeyli 2009; Shaker 2005) has its own optimal parameter to be used with the filter. For example, the most relevant information in EEG is contained in the range of 20-500 Hz (Chen et al. 2010; Konrad 2005) while heart rate effects can be eliminated with a low of 100 Hz. However, there are no perfect filters to remove unwanted artefacts (Fergus et al. 2013). Fele-Zorz et al. (2010) showed that 0.3-3 Hz is the best for classifying between preterm and term delivery (Fele-Zorz et al. 2008). Nevertheless, the frequency range of the motion noise is 1–10 Hz (Chowdhury et al. 2013).

6.5 Feature Extraction

Biomedical signals can be classified using assets of attributes or features. The features can be computed using feature extraction methods. Feature extraction is known as a conversion process of signals to features, and these features can characterise the properties of the signal (Wang et al. 1997; Übeyli 2009). A number of techniques have been used to represent and extract features from biomedical signals in order to improve the classification (Forney and Anderson 2011; Subasia et al. 2005). Various frequency and temporal analysis measures have been utilised to represent signals (Song 2010). Time domain analysis can be conducted using statistical characteristics including autoregressive coefficients (AR), zero crossing, waveform length, and root mean square (RMS) (Rosa et al. 2007; Chen et al. 2010; Fele-Zorz et al. 2008; Phinyomark et al. 2012). For frequency analysis, the methods include the Fourier transform technique, which divides signals into sinusoidal components of different frequencies (Shaker 2005). This is done by applying discrete Fast Fourier Transform (FFT) to the signal and then finding the signal amplitude, power and energy (Sarbaz et al. 2011; Chiang 2010).

Although the field of spectral analysis has been dominated by the use of the Fourier transform method, some studies have shown that features extracted in the frequency domain are considered the best for recognising mental tasks based on EEG signals (Liu et al. 2005), and they have been used to classify uterine EMG signals into labour or non-labour and term or preterm classes (Moslem et al. 2011; Bazregar and Mahdinejad 2013; Fele-Zorz et al. 2008; Garfield et al. 2005). However, that method suffers from high noise sensitivity (Majumdar 2009). Fele-Zorz et al. (2008) used Power Spectral Densitt (PSD) to generate features of the uterine signals. PSD actually represents only the assessed power across a range of frequencies (Forney & Anderson 2011). However, this method produces good results, but the Fourier functions do not sufficiently deal with non-stationary signals (Liu et al. 2005; Chowdhury et al. 2013b). In reality, most biomedical signals are highly non-stationary signals such as EMG (Tsuji et al. 2000; Moslem et al. 2010).

Another method that can be used for such a process is wavelet transform, which is based on the time-frequency domain (Shaker 2005). Wavelet transform allows the transformation of signals into smaller waves (Sifuzzaman et al. 2009), which is often used in EMG and has produced good results (Eswaran et al. 2002). Although the frequency and time analysis methods provide the benefits of physical interpretation and convenient computation (Chen et al. 2010), these types of methods have suffered from the ability to obtain both spatial and temporal information from biological signals (Forney and Anderson 2011; E. M. Forney & Anderson 2011). They ignore the essential nonlinear dynamics behaviour of signals (Übeyli and Guler 2007; Balli and Palaniappan 2010). Successful classification depends on utilising signal representation that captures the crucial information from signals needed to estimate the correct classes. In reality, most biological and biomedical signals have nonlinear dynamic properties with complex behaviour (Meng et al. 2010; Chen et al. 2010; Meng et al. n.d.; Tsuji et al. 2000). These types of signals are considered chaotic (Guler et al. 2005; Übeyli et al. 2008; Übeyli & Guler 2007). Therefore, nonlinear dynamic measures can be used as clinically useful parameters to capture salient information about signals (Rezek & Roberts 1998). These nonlinear methods are able to measure complexity (Abasolo et al. 2006; Costa et al. 2003; Costa et al. 2005; Pincus et al. 1993; Radhakrishnan & Gangadhar 1998; Radhakrishnan et al. 2000; Rezek & Roberts 1998; Richman & Moorman 2000; Xu et al. 2007; Zhang et al. 2001).

During the last few decades, the use of nonlinear analysis techniques to determine the character of biological signals' properties has increased significantly. A number of measures are available to discover nonlinear characteristics in signals. In the literature, they have been used in nonlinear biosignal analysis such as EEG (Übeyli & Guler 2007), ECG (Übeyli 2010) and EMG (Fele-Zorz et al. 2008; Diab et al. 2012; Hassan et al. 2009; Balli & Palaniappan 2010; Meng et al. 2010). Among the nonlinear features are Lyaponov exponents, which are quantitative measures for estimating the chaos in a signal (Fele-Zorz et al. 2008). The approximate entropy is another nonlinear feature. Various studies have proved that these features can properly represent the biomedical signals and that, by using these features, a good distinction between classes can be achieved (Fele-Zorz et al. 2008; Balli and Palaniappan 2010; Übeyli 2010; Diab et al. 2012). One study confirmed that the progress of the labour can be evaluated using sample entropy features (Vrhovec 2009). Moreover, there are other nonlinear feature measures such as correlation dimension. These can be used to distinguish between term and preterm labour, as proved by Fele-Zorz et al. (2008). However, it should be taken into account that each feature represents a different classification power for different problems. Therefore, feature extraction methods are playing an important role in improving the discriminative performance of classifiers (Moslem et al. 2011). For example, in neural networks the selection of (ANN) inputs is the most essential factor in designing the neural network model based on pattern classification because the best classifier will perform poorly if the inputs are not selected carefully (Übeyli 2010).

In some cases of feature extraction, the features vectors will be high dimensional, which will increase the computational difficulty. This is related to the presence of irrelevant and noisy features that can affect the classification performance. For a high dimensional data set, the commotion time for measuring and calculating similarity between data becomes more complex, and has led to misunderstandings of the data structure. Therefore, the high dimensionality of the extracted feature vectors must be reduced using feature selection techniques. This can be achieved by finding new space that is a lower dimension space than the dimension of the original data. It is used to improve the ability of the ANN in classification (Khemphila & Boonjing 2012). It has been confirmed that ANN classification accuracy can be improved by reducing the number of unnecessary features that are recorded from patients. Feature selection has two types: (1) feature transformations, where the data are projected onto a lower dimensional space, such as principal component analysis (PCA) and independent component analysis (ICA); and (2) selecting the set of inputs that best represents a given pattern and can be utilised based on certain statistical features such as the mean or the maximum or minimum of the feature values.

6.6 Recurrent Neural Network for Pre-Processing

Human biological signals are naturally characterised by complexity, dynamism and the inherent nonlinearity. These characteristics of biological signals make their analyses very difficult. Since variables in medical time series are related to the time domain (Verplancke et al. 2010), therefore, the necessity of using techniques that are able to deal with both spatial and temporal information on medical time series data is highly essential. The recurrent neural networks have enjoyed very important properties which make them good for pattern recognition. One of their powerful properties is finite state machine approximation, which makes recurrent neural network learning both temporal and spatial patterns (Forney & Anderson 2011). This kind of network is very useful for real-time application like biomedical signal analysis. Some of the various RNNs developed were designed in order to detect patterns in biological signals. Certain research work has proved that RNNs are a very powerful tool for modelling biomedical signals such as EEG signal (Forney & Anderson 2011). Forney and Anderson (2011) have shown that the Elman RNN is able to detect mental tasks. It has shown its ability to forecast the EEG signal. Their process was based on Classification via Forecasting (CVF). Each EEG signal is recorded from a person while he/she imagines mental tasks. ERNN has been trained to forecast the signals of each of these imagined mental tasks. The forecasting errors of ERNN are fed to the classifier as features; then the label of class is selected with the ERNN model that obtained the lowest forecasting error. This experiment has been performing very well, and has achieved up to 93% classification accuracy. This is related to the dynamical link on the ERNN, which holds some of the temporal information from the EEG signal.

There are some existing studies which are based on the investigation of the recurrent neural network's ability to model systems to generate time series signals of some diseases. Sarbaz et al. (2011) investigated the ability to produce a model for basal ganglia structure. The basal ganglia are a collection of nerve cells in the brain which are strongly interconnected. The researchers' proposed models exist in order to generate gait time intervals for healthy patients and those with Parkinson's disease. They have used the Elman recurrent neural network to build their models. Their experiments concluded that the recurrent neural network is capable of simulating conditions of healthy patients and those with Parkinson's disease. Furthermore, RNNs have the ability to generate the behaviour of different persons as healthy or unhealthy patients (Sarbaz et al. 2011).

The recurrent neural network has also been utilised for filtering and signal detection in order to reduce the noise associated with biomedical signals. Erfanian and Mahmoudi (2005) utilised a nonlinear adaptive noise canceller (ANC) filter for ocular artefact cancellation. Their motivation was based on filtering EEG recordings by removing noises that affect the eye blink or eye movement. Their proposed work uses an ANC filter based on using the recurrent neural network to remove the Electrooculogram (EOG) interference from EEG signals. The EEG signals were recorded at frontal site F3, and the EOG signals were recorded by using two pairs of electrodes; the first pair was placed above and below the eye to record the vertical Electrooculogram (EOG), and the second pair was placed at the left and right outer canthi to record the horizontal EOG. Then, EOG signals were added to the EEG signals to obtain the main signals. After that, the main signals were passed to the nonlinear recurrent neural network as inputs. The recurrent neural network filter achieved a signal to noise ratio (SNR) of 27 dB. Their experiments confirmed that the nonlinear ANC filter based on the recurrent neural network has a high ability to detect the EOG to EEG signals and to perfectly remove ocular artefacts from the main signals (Erfanian & Mahmoudi 2005).

Another RNN-based approach was investigated in 1996 by Cheron et al. Their main objective of using dynamic recurrent neural networks was to find the connection between the arm kinematics and the muscle EMG activity. The neural network that was used in their study consists of fully interconnected neurons and their experiments show that this RNN is perfectly able to identify muscle activity EMG signals. It can identify the complex mapping between the upper-limb kinematics and the muscle EMG activity during complex actions (Cheron et al. 1996).

Chowdhury et al. (2013) presented a number of experiments using neural networks to reduce the noise on the EMG signals. In their study, a number of neural network were applied in EMG signals to check their performances based on mean square error and correlation coefficients. Their result showed that the recurrent neural network is able to reduce noise on EMG signals with MSE 0.009, compared to MLP which achieved 0.01. From their experiments, the authors suggested that using recurrent neural networks as noise removal methods is highly useful.

6.7 Classification

The importance of classification techniques in the medical community, especially for diagnosis purposes, has gradually increased. The important reason for improving medical diagnosis is to enhance the human ability to find better treatments, and to help with the prognoses of diseases to make the diagnoses more efficient (Akay 2009), even with rare conditions (Machado 1996). The classification task involves the following: each object in a data set is represented by a number of attributes or features, and each of these objects can be determined according to a number of classes to which it belongs. The features can be assembled into an input vector *x*. The classifier will be provided by a number of previous objects (training set), each involving vectors of feature values and the label of the correct class. The aim of the classifier is to learn how to extract useful information from the labelled data in order to classify unlabelled data. Various methods have been employed for the classification task. They are categorised into two groups: linear and nonlinear classifiers.

The linear classifiers are represented as a linear function of input feature *x*.

$$g(x) = wTx + b \tag{6.1}$$

Where *w* is a set of weight values and *b* is a bias. For two classes, problem c_1 and c_2, the input vector *x* is assigned to class c_1 if g(x)>=0 and to class c_2, otherwise. The decision boundary between class c_1 and c_2 is simply linear. In the previous studies, several traditional linear classifiers were designed and applied to perform classification in different areas such as Linear Discriminant Analysis.

Nonlinear classifiers involve finding the class of a feature vector *x* using a nonlinear mapping function (*f*), where *f* is learnt from a training set *T*, from which the model builds the mapping in order to predict the right class of the new data. The most popular nonlinear classifier is the neural network. As a classifier, the ANN has a number of output units, one of each probable class. Nonlinear neural networks are able to create nonlinear decision boundaries between dissimilar classes in a non-parametric approach (Haykin 1998; Chen et al. 2010). Zhang (2000) asserted that neural networks have the power to determine the posterior probabilities, which can be used as the basis for establishing the classification rule.

ANN achievements have covered different types of medical applications such as the analysis of EEG signals (Guler et al. 2005). Diab et al. (2010) have used the ANN to classify uterine EMG signals for preterm deliveries and deliveries at term according to their frequency domain (Diab, El-Merhie, El-Halabi 2010). Tsuji et al. (2000) have used the ANN to classify non-stationary EMG signals during continuous motions over a short period of time.

6.8 Recurrent Neural Network for Medical Classification

Classification in medical time series signals is considered a challenging task. In recent years, the analysis of dynamic behaviour in biological and biomedical signals has received great attention (Dingwell 2006). In the literature, a number of studies have sought to improve the classification accuracies of these signals by employing different pre-processing, feature-extraction approaches and ANN architectures in order to improve diagnosis (Forney & Anderson 2011). The difficulty of classifying signals could be solved by a dynamic system such as a recurrent neural network (RNN) (Szkoła et al. 2011).

In the literature, it has been shown that RNNs have the ability to discover the hidden structure of the medical data. Existing studies have indicated that RNNs have the ability to perform pattern recognition in medical data and have obtained high accuracy in the classification of medical data (Übeyli and Übeyli 2008; Verplancke et al. 2010; Ilbay et al. 2011; Übeyli 2009; Petrosian et al. 2001). In addition, it has been shown that RNNs have the ability to give an insight into the feature used to represent biological signals (Übeyli 2009). Therefore, the employment of a dynamic tool to deal with time series data classification is highly recommended (Hüsken and Stagge 2003).

The most popular recurrent neural network used for medical classification tasks is the Elman neural network (ERNN) (Elman 1990). The Elman network has the ability to detect and classify temporal constructions (Elman 1990). This network has been used widely in medical classification. According to some studies, the recurrent links of the RNN help to remember the past information without causing any complexity in the learning process (Guler et al. 2005).

In the last couple of years, various medical applications based on RNNs have been implemented. One of the most prominent applications of the RNN is pattern recognition, such as automated diagnostic systems (Übeyli and Übeyli 2008). The RNN can utilise nonlinear decision boundaries and process memory of the state, which is crucial for the classification task (Guler et al. 2005; Petrosian et al. 2001; Petrosian et al. 2000). A number of studies have confirmed that RNNs have the ability to distinguish linear and nonlinear relations in the signals. In addition, they have proven that RNNs enjoy signal recognition abilities (Petrosian et al. 2001). Different research works have attempted to investigate the ability of RNNs to classify biological signals (e.g EEG, ECG and EMG). The procedure for signal classification is performed in two stages: extracting features which were used as input to the RNN classifier, then classifier techniques can be performed.

Currently, most research work is based on using recurrent neural networks for EEG signal classification (Koskela et al. 1998), and others have addressed the utilisation of recurrent self-organising map (RSOM) to EEG signals for epileptic patients. It has been applied to detect the activity of epilepsy on EEG signals. The features that have been used in this experiment are spectral features. Wavelet transform was used to extract signals from each window, and sixteen energy features from the wavelet domain have been computed for each window. The RSOM network has been run to classify the EEG signal to normal or epileptic activity. Their results show that the RSOM achieved a better clustering result than the SOM. The authors conclude that using context memory for detecting the EEG epileptic activity has enhanced the classification performance of the SOM (Koskela et al. 1998).

Another study was presented based on using the Elman network to classify mental diseases on EEG signals combined with wavelet pre-processing. Petrosian et al. (Petrosian et al. 2001; Petrosian et al. 2000) investigated the ability of RNN employed with wavelet pre-processing methods for diagnosis of epileptic seizures in EEG signals (Petrosian et al. 2000) and for the early detection of Alzheimer’s Disease (AD) in EEG signals (Petrosian et al. 2001). For the diagnosis of epileptic seizures in EEG signals analysis task, Petrosian et al. (2000) examined the ability of recurrent neural networks (RNN) combined wavelet transformation methods to predict the onset of epileptic seizures. The recurrent neural network was trained based on the decoupled extended Kalman filter (DEKF) algorithm. In Alzheimer’s disease in the EEG signals detection task, the RNN has been used to distinguish between AD and healthy groups. In that study, the authors have used a network training algorithm based on the Extended Kalman Filter (EKF) algorithm. The signals were obtained from ten healthy persons and ten early AD patients. The EEG signals were recorded using nine channels with two min length and with 512 Hz sampling rate. EEG has been recorded to monitor the subject during the eyes closed resting state. The Fourier power spectra methods have been used to analyse the row EEG signals. The band-pass FIR filter has been used to filter each EEG signal into four sub-groups: delta, theta, alpha and beta. Furthermore, the fourth level wavelets filter has been used on raw EEG signals. In the study, the inputs of the RNN were the original channel signals and the derived delta, theta, alpha and beta for each signal as well as their wavelet filtered sub-bands at levels 1-6. From their experiments, the best RNN result was achieved using parietal channel P3 raw signals as well as wavelet decomposed sub-bands at level 4 as inputs. The RNN achieved a high performance to classify AD with 80% sensitivity and 100% specificity. Petrosian et al. (2001) have proved that the combination of RNN and wavelet approach has the ability to analyse EEG signals for early AD detections.

Guler et al. (2005) have also investigated the diagnostic ability of the recurrent neural networks (RNNs) to detect EEG signals of epileptic seizures. The EEG signals used in that experiment were recorded from five healthy volunteers with their eyes open, five epilepsy patients in the epileptogenic zone during seizure-free interval and epilepsy patients during seizures. Lyapunov exponent methods have been applied to extract features. One hundred and twenty-eight Lyapunov exponent features have been calculated for each EEG segment. The statistical methods were used to reduce the dimensionality of the features. This was done by computing maximum, minimum, mean and Standard deviation of the Lyapunov exponents for each EEG signal. The result achieved in this study confirmed that RNNs are able to classify EEG signals better than MLP. The classification accuracy percentages of the RNN were approximately 97% for the healthy subjects, 96.88% for seizure-free epileptogenic zone subjects and 96% for epileptic seizure subjects, while MLP classified the healthy subjects, seizure-free epileptogenic zone subjects and epileptic seizure subjects with 92%, 91% and 90.63%, respectively (Guler et al. 2005).

Another study attempted to evaluate the diagnostic accuracy of the recurrent neural networks (RNNs) by utilising eigenvector methods to extract features of EEG signals of epileptic seizures (Übeyli 2009). Consequently, the data involved five groups: two containing healthy people and three containing people with an epilepsy diagnosis. Each set contains 100 single-channel EEG signals of 23.6 s period. The signals were filtered using a Band-pass filter with 0.53–40 Hz. This research used three eigenvector methods (Pisarenko, multiple signal classification, and Minimum-Norm) to calculate the power spectral density (PSD) of signals. Frequencies and power levels of signals have been obtained by these eigenvector methods.

After extracting the features, the feature selections are proposed by finding the logarithm of the PSD values of each eigenvectors method. Then two types of neural network have been used to classify the signals, the MLP and the Elman recurrent neural network (ERNN). The result indicates that the ERNN has succeeded in classifying the EEG signals. This network provided the best classification performance, with an accuracy of 98%. The ERNN outperformed the MLP, which had an accuracy of 92% (Übeyli 2009). From these experiments, it has been concluded that the margin between eigenvector methods and RNN approach can be used to discriminate between the other biomedical signals.

Another biomedical signal that has been used to investigate the RNN ability for classification is ECG signal. Übeyli and Übeyli's (2008) study has also used a RNN to diagnosis ECG signals of partially epileptic patients. The RNN has been applied to classify non-arrhythmic ECG waveforms as normal or partially epileptic. The ECG signals involve two types of beats, normal and partial epilepsy, and they were collected from the MIT-BIH Database which was created by the Massachusetts Institute of Technology (Al-Aweel, Krishnamurthy , Hausdorff , Mietus , Ives , Blum , Schomer 1999). The features were extracted by using wavelet coefficient and Lyapunov exponents. Also, in the ECG experiment the authors have used statistical methods to reduce the dimensions of the extracted features. The trained ERNN obtained high classification accuracy of 98% compared to MLP, which achieved 93%.

In addition, Übeyli (2010) used the ERNN to distinguish the differences in beats on electrocardiogram (ECG) signals. An ECG signal involves four beats (normal beat, congestive heart failure beat, ventricular tachyarrhythmia beat, and atrial fibrillation beat). The ECG signals contain 48 signals with 30 min length. In the experiment, the electrode was placed on the subject's chest. The band-pass filter was used to digitalise signals at 360 Hz. The features were extracted used a nonlinear dynamic method which is Lyapunov exponents. The Elman recurrent neural network with Levenberg-Marquardt leaning algorithm was applied in order to classify the ECG signals. The result of the study has confirmed the ability of the ERNN to classify ECG signals with 94.72% accuracy (Übeyli 2010).

Another biomedical signal that has been used to examine the RNN capacity for classification is EMG signal (Arvind et al. 2010). This study focused on the automated detection of Parkinson's disease (PD) by using a recurrent neural network to classify EMG signals. The Elman recurrent neural network (ERNN) has been used to classify the health and PD states.

The EMG signals were recorded from the extensor carpi radialis muscle during rest and activated motion. The resting motion signals were obtained from abnormal PD patients, and muscular contractions signals were obtained from healthy persons. The signals' duration was 30 min with sampling frequency of 100 Hz. In order to distinguish the EMG signals, the authors have used the power spectral density features. The statistical measures, which are mean and maximum of PSD, were computed as features. From their experiments, their result shows that the ERNN can classify EMG signals with 95% classification accuracy (Arvind et al. 2010). Other studies have attempted to classify different types of conditions related to human muscles. For example, Ilbay et al. (2011) used the Elman recurrent neural network (ERNN) for automated diagnosis of Carpal Tunnel Syndrome (CTS). It has been applied to patients suffering from various Carpal Tunnel Syndrome symptoms such as right CTS, left CTS and bilateral CTS. In this experiment, the study collected EMG signals from 350 patients who suffered from CTS (left, right and bilateral) symptoms and signs. Nerve conduction study (NCS) was applied by using surface electrodes to record the EMG signals on both hands for each patient. NCS measures how fast electrical signals can be sent through nerves. Therefore, they are able to diagnose Carpal Tunnel Syndrome and the results of this test are used to evaluate the degree of any nerve damage.

Furthermore, the RNN has been used to classify the signals recorded from the Doppler system. For example, (Übeyli and Übeyli (2008) evaluated the diagnostic ability of the Elman recurrent neural network (ERNN) employing Lyapunov exponents to classify arterial disease. In that study, signals were collected from a Doppler ultrasound. The Doppler ultrasound method has been used to evaluate blood flow in both the central and peripheral circulation (Evans et al. 1989). The main motivation of the study was to obtain nonlinear dynamic features from the ophthalmic arterial (OA) and internal carotidarterial (ICA) Doppler ultrasound signals. One hundred and twenty-eight Lyapunov exponent features have been calculated from each OA and ICA Doppler signal segment (256 discrete data). However, these features have been reduced using statistical methods to represent signals for classification. The trained ERNN in this feature obtained high classification accuracy with 97% for OA Doppler signals and ICA Doppler signals.

The next sections will introduce the medical issue that has been considered in this research work, which is Preterm cases' classification from EHG signals.

6.9 Introduction to Preterm

One of the most challenging tasks currently facing the healthcare community is the identification of premature labour (Maner 2007). Premature birth occurs when the baby is born before 37 weeks of pregnancy. A term birth occurs when the baby is born after the 37-week gestation period.

The number of preterm births is increasing gradually; it badly affects healthcare development. The increase in preterm labour contributes to rising morbidity. It has been recorded that, in 2011, the percentage of babies born as preterm was 7.1 % in England and Wales, according to the Office for National Statistics (2013). Approximately 50% of all perinatal deaths are caused by preterm delivery (Baker and Kenny 2011), with those surviving often suffers from health problem caused during birth.

6.9.1 The Negative Effects of Preterm Labour

Preterm birth has a great impact on new babies' lives, including health problems or increased risk of death. One million preterm babies die each year according to the World Health Organisation (WHO) (2012). An earlier delivery has a significantly negative impact on babies' later lives. Preterm infants are usually born at low weights of less than 2500 grams compared with full-term babies (Garfield and Maner 2007). In their future lives, they might suffer from more neurological, mental and behavioural problems compared with full-term infants (Maner 2007). In other cases, preterm birth leads to increased probability of asthma, hearing and vision problems; some preterm infants may have difficulty with fine-motor and hand-eye coordination (WebMd 2012). An early delivery impacts the development of the kidneys and their function in the infant's future life (Heijden 2012). Furthermore, 40% of survivors of extreme preterm births may be affected by chronic lung disease (Greenough 2012).

On the other hand, preterm births also have a negative effect on families, the economy, and community (Fergus et al. 2013). According to the World Health Organisation, more than three-quarters of premature babies can only be saved with very high-level effective care (WHO 2012), which results in more infant hospitalisations and a lot of healthcare expenditure. Preterm infant needs intensive care, which will raise the cost of hospital care to $1500 a day (Garfield and Maner 2007). Furthermore, the reduced gestation duration increases the number of days spent in hospital. As a result, preterm births have a negative economic effect (Shi et al. 2008). According to Mangham et al. (2009), in 2009, in England and Wales the total cost to the public sector of preterm births was valued at £2.95 billion. However, attempting to have a better understanding of preterm deliveries can help to create the right decision and prevention strategies to reduce the negatives effects of preterm deliveries on babies, families, societies and healthcare services (Muglia and Katz 2010; Fergus et al. 2013; Iams 2003).

6.9.2 The Factors for Preterm

Significant progress has been made in understanding the process of labour, and research on premature labour has attempted to discover the risk factors (Chen et al. 2011; G. Fele-Zorz et al. 2008). A number of investigators have found many factors leading to preterm delivery. According to Baker and Kenny (2011), approximately one-third of preterm deliveries occurred because of the membranes rupturing prior to labour. Another third might happen due to the increasing of spontaneous contractions (termed preterm labour or PTL) (Fergus et al. 2013; Greenough 2012). Lastly, preterm birth can occur because of medical indication towards the best interest of the mother or baby. Moreover, there are still doubts about which of these factors can increase the risk of preterm birth. On the other hand, there are some reasons for preterm labour which ultimately may or may not end in preterm birth (Baker & Kenny 2011). These reasons may relate to the mother’s illnesses, congenital defects of the uterus and cervical weakness (Rattihalli et al. 2012; Goldenberg et al. 2008; Goldenberg et al. 2008). Other factors of preterm labour could be related to health and lifestyle of the mothers; these factors include uterine abnormalities, short cervix, recurrent antepartum haemorrhage, illnesses and infections, any surgery, underweight or obese mother, diabetes, stress, smoking, social deprivation, long working hours/late nights, alcohol and drug use, and folic acid deficiency. However, in some situations, the cause of preterm delivery is undetectable (Diab, El-Merhie & El-Halabi, 2010).

6.9.3 The Prediction of Preterm Birth and Labour

The medical community has made significant progress in alleviating the effects of preterm birth and preterm labour, improving the care of immature babies. Effective diagnosis of preterm labour could contribute to appropriate lifestyle and medical interventions in saving the lives of many infants (Muglia & Katz 2010). It can help medical specialists to stop the labour’s progress or give effective treatment to decelerate it, or at least make careful preparation for preterm infants (Shi et al. 2008; Garfield & Maner 2007). The prediction of preterm birth and diagnosing preterm labour obviously has important consequences for healthcare and the economy. During pregnancy, early prediction of preterm birth can help to select the relevant necessary treatments (Rabotti et al. 2007). However, incomplete understanding of the physiology of the uterus and parturition means that premature labour prediction is a difficult task (Fele-Zorz et al. 2008; Leskošek & Pajntar 2002). The reason for this may be that the initial symptoms of preterm labour occur commonly in normal pregnancies (Iams 2003). There is some misclassification in regard to recognising full-term and preterm labour; approximately 20% of women who are identified as reaching full-term labour actually deliver prematurely (Iams 2007). Therefore, there is great demand for early accurate diagnosis of preterm labour.

A number of clinical methods have been implemented to predict labour, such as uterine contraction measurements using an elastic belt (external tocography), cervical change test, salivary estriol, fetal fibronecti, and intrauterine pressure. Unfortunately, these methods have not proved to be accurate in predicting preterm delivery (Maner 2007). The importance of understanding uterine activity and the processes underlying labour in order to diagnose the true path of labour and predict the delivery time has contributed to the appearance of different techniques to monitor the contractions, including the internal uterine pressure (IUP), tocodynamometer, and Electrohysterography (EHG). A tocodynamometer records contractions by using a pressure transducer, and an external belt fastened around the abdomen. The IUPC measures the pressure inside the uterus. IUP is done by an intrauterine catheter to measure the most complicated deliveries. This type of measurement is considered unsafe because it can damage the foetus (Rabotti 2010; Rabotti et al. 2008). However, the use of an external tocodynamometer only provides information that is related to contraction frequency (Rabotti 2010). Furthermore, the problem of using tocodynamometry in detecting preterm labour is that it does not measure the contraction duration and amplitude, which ends up in giving poor analytical results for predicting preterm delivery (Eswaran et al. 2002; Rabotti 2010).

The Electromyography (EMG) technique is considered to be a helpful and effective method to detect preterm labour. EHG is very efficient measurement to record electrical activity, because it measures the contraction directly, rather than the physical response to contractions, which may get lost amongst other physical noise and disturbance (Leman et al. 1999). Hence, Electromyography is considered in this thesis. The next section gives a brief introduction to EMG and especially Electrohysterography (EHG).

6.9.4 ElectroHysteroGram (EHG)

It has been recorded that electrical activity of the uterus muscle has been known for a long time, since at least the late 1930s (Bozler 1938). However, it is only in the last twenty years that formal techniques have been available to record these activities (Cheung 2012). The activity is recorded as signals. The method that has been used to record such signals in a time domain is called Electrohysterography (EHG). EHG is a technique for measuring electrical activity of the uterus muscle during pregnancy, through uterine contractions (Garfield et al., 2005; Marshall, 1962). EHG is one form of electromyography (EMG), the measurement of activity in muscular tissue.

The uterine muscle is like skeletal muscles. In smooth muscles, as Rabotti (2010) asserted, the way the contraction occurs is by the process of propagation of electrical activity over the muscle cells in the appearance of an action potential (AP). The spreading of electrical activity in the action potential (AP) through the myometrium cells causes uterine contractions. Therefore, EHG is the measurement of the AP propagating through the myometrial cells. Figure 6.1 represent the contraction that happens on muscle. The woman's body will slightly increase the number of electrical connections (gap junctions) between uterine cells (Fergus et al. 2013).

Figure 6.1: Schematic representation of the smooth cell contraction (Rabotti 2010)

From a medical point of view, the strengthening and increasing of uterine contractions over time is a sign of imminent labour and birth (WelcomeBabyHome 2006), and shoots up particularly in the last four days before delivery (Lucovnik et al. 2011). During parturition, the increasing of the contractions will help the body to prepare for the final stage of labour and parturition (Cheung 2012; Fergus et al. 2013). They will help to shorten the cervix and force the foetus to descend into the birth canal. Therefore, the main function of uterine contractions is to generate the force and synchronicity that are necessary for true labour.

Over the last few decades, EMG has been used in two ways: the older method is an invasive one involving the insertion of needle electrodes into the uterus; however, this method is painful and uncomfortable for patients. Hence, it is unwanted. The second method is a non-invasively one which places electrodes on the woman’s abdominal surface. Many experiments have used non-invasive EHG signals in order to study the pregnancy process and predict labour in both humans (Maner 2007; Fele-Zorz et al. 2008) and animals (Shi et al. 2008; Marque et al. 2007).

EHG signals have been recorded by placing bipolar electrodes on the abdominal surface. These electrodes are spaced out at a horizontal, or vertical, distance of 2.5cm to 7cm apart. The numbers of electrodes that have been used for recording EHG have been chosen differently in various studies. One study utilises two (Doret 2005) while other studies managed to use EHG signals that recorded from four electrodes (Fele-Zorz et al. 2008; Cheung 2012; Fergus et al. 2013). Other studies used sixteen electrodes to obtain EHG signals (Moslem et al. 2011; Diab, et al. 2011; Moslem, Khalil, et al. 2012; Moslem and Khalil 2011b), and a high density grid of 64 small electrodes was used in Rabonetti et al. (2010).

The results of these different studies have confirmed that EHG records are different from woman to woman, depending if she is in true or false labour and whether she will deliver term or preterm (Hassan et al. 2010). Therefore, EHG can be used to predict and diagnose preterm birth.

In the literature, a number of research studies have confirmed the importance of EHG recordings to analyse the uterine contraction during pregnancy and parturition (Rabotti et al. 2008; Buhimschi and Garfield 1996). Analysis of the EHG provides a strong basis for understanding and identifying the progress of labour (Devedeux, et al., 1993; Fele-Zorz et al. 2008; Gondry, Marque & Duchene 1993; Maner et al. 2003). Furthermore, Gondry et al.

(1993) recognised uterine contractions from EHG records as early as 19 weeks of the pregnancy period.

6.9.5 Literature Review of Preterm Labour

Many research studies have used EHG for prediction or detection of true labour in term and preterm cases. Many pattern classification techniques have been used to classify groups of patients according to their EMG parameters. The different studies that focused on classifying the EHG signal will be presented in this section.

Fele-Zorz et al. (2008) presented a study that compared linear and non-linear signal processing techniques to efficiently classify EHG signals in order to separate different classes of EHG records into term and preterm classes. They presented a set of data called The Term-Preterm EHG Database (TPEHG), and used a statistical analysis method to study the differences in EHG. Their database set was used in this thesis and will be described in Chapter 7. Fergus et al. (2013) have used same uterine EHG signals which were collected from Physionet but with different filter parameters. They used 0.3 to 1 HZ. Leman et al. (1999) used a statistical analysis approach to observe the disparity in the EHG parameters and the possibility of such parameters in allowing discrimination between preterm and term classes. In contrast, Diab et al. (2007) have used two methods of classification. The first method is based on unsupervised statistical classification (*USCM*) combined with the pre-processing method of Wavelet Transform while the second method is based on the Autoregressive model (AR) and k-nearest neighbour model (KNN). The basis of USCM is the Fisher Test and k-Means methods. The pre-processing techniques that were used in their study were AR modelling and wavelet transform.

The objective of their study was to classify contractions from 16 women and divide them into three groups. The first group (G1) contained women who had their contractions recorded at 29 weeks, and then delivered at 33 weeks; the second group (G2) had their contractions obtained at 29 weeks, but they delivered at 31 weeks; and the third group (G3) were recorded at 27 weeks and delivered at 31 weeks. The first classification task is to classify G1 and G2; the second classification task is to classify G2 and G3. The study result demonstrated that the wavelet transform, combined with *USCM*, could achieve a classification error of 9.5% when discerning G1 against G2, and 13.8% when classifying G2 against G3. In contrast, *AR* accompanied with the *k-NN* achieved classification of 2.4% for G1 against G2 and 8.3% for

G2 against G3. Therefore, it can be concluded from their results that the *AR* and *k-NN* methods achieved better results than the *USCM*. Furthermore, the classification accuracy of G1 against G2 was always lower than the equivalent G2 to G3 classifications. This suggests that it is easier to distinguish between pregnancies recorded at different stages of gestation than it is to distinguish between the times of delivery.

In the literature, a support vector machine (SVM) has been used to classify pregnancy and labour contractions (Moslem et al. 2012; Moslem & Khalil 2011b). Their works are based on classifying contractions into labour or non-labour, by using different sites on the abdomen. Their approaches were tested on a multichannel EMG signal recorded from 16 electrodes. The power of the contractions was used in these studies, and the median frequencies were extracted from the signals corresponding to each channel. In Moslem and Khalil (2011a), Moslem and Khalil (2011b) and Diab et al. (2012), the researchers have proved that the classification of multichannel uterine signals can be improved by using the fusion rule. The global accuracy of the studies using SVM combined with the Data Fusion Approach was 78% to 88%. Furthermore, SVM has been used to prove that reducing the number of channels can also increase the classification accuracy to diagnose between pregnancy and labour contractions, as shown by Diab et al. (2012); their result demonstrated that the combination of four channels from 12 channels of uterine signals yields the best classification accuracy of 84%.

6.9.6 Artificial Neural Networks in Medical Applications

Artificial neural networks have been used extensively in medical diagnosis application (Al-shayea 2011). They have been applied for a number of different tasks in biomedical research. They are used for diagnosis, prognosis, or clustering of medical data (Machado 1996). ANN is a very active research field in medicine, and many studies have attempted to investigate the application of neural networks in typical disease classification such as Alzheimer’s Disease (Joshi et al. 2010), cancer (Singh & Gupta 2011) and heart disease (Khemphila & Boonjing 2011). ANN can also be used as a pre-processing method of medical data. In this section of the thesis, summaries of ANN application of classifying uterine EHG signal are presented.

Neural networks for medical classification are simply based on presenting a set of attribute values (features) as input and these values might be continuous, e.g. “temperature”, or discrete, e.g. “gender/ sex, age”, and the neural network concludes whether the person is normal or abnormal. In our case, the preterm cases will be considered as abnormal and term cases will be normal.

ANNs have been used in a large number of studies to classify uterine EMG signals (Diab et al. 2012; Shi et al. 2008; Baghamoradi et al. 2011; Chen et al. 2011; Marque et al. 2007; Diab, El-Merhie, El-Halabi 2010; Garfield and Maner 2007). The use of EMG with ANN has shown its ability to diagnose term and preterm births (Baghamoradi et al. 2011; Diab, El-Merhie & El-Halabi 2010; Marque et al. 2007). Other studies have used ANN to identify true labour (Diab, et al. 2011; Shi et al. 2008; Maner 2007; Charniak 1991; Doret et al. 2005), regardless of whether they were term or preterm. In Shi et al. (2008), it was shown that ANN can efficiently identify true labour in term and preterm on animal EMG signals and uterine pressure data. In term births, the classification has been made with 100% accuracy; however, the preterm delivery was 92% for the labour group and 86% for the non-labour group.

Maner et al. (2007) used the Kohonen ANN on human uterine signals to distinguish between term and preterm and the labour/non-labour classes. The study comprised 134 term and 51 preterm pregnant women. The global classification accuracy was 80%. Diab et al. (2011) have applied ANN on EMG signals collected from 16 electrodes placed on the abdomen in order to classify each channel into labour/non-labour classes. The results have proved that the classification accuracy of each of the 12 sensors differs from one to another. However, the informative combination from different sensors can improve and strengthen the classification result. Diab et al. (2011) have used the data fusion rule to make decisions about each channel; after the fusion procedure, the ANN classification has obtained better results than a single channel could have provided.

Baghamoradi et al. (2011) used cepstral check analysis to differentiate between term and preterm births. They used the TPEHG database (Fele-Zorz et al. 2008) to estimate classification accuracy. The uterine signal in this experiment was recorded from four electrodes. The features used in this study were thirty cepstral coefficients and sample entropy, which were calculated from each channel. The sequential forward feature selection and Fisher’s discriminant have been applied before the classification task to select the optimal features. A multilayer perceptron neural network classified the records into term and preterm. The selection of informative features leads to improved classification accuracy from 53% to 73%.

Lu et al. (2007) used a MLP to classify between term and preterm signals. The signals were filtered using wavelet packet transform. The feature was average wavelet packet energy of the whole signal. Their results demonstrated that MLP can classify term and preterm labour signal with 64% accuracy.

Maque et al. (2007) used MLP to diagnose the risk of contractions leading to preterm labour; their result showed that the early diagnosis of preterm labour can be detected very early, in the 27th week of pregnancy. Another study has also used ANN to discover the risk elements leading to preterm birth (Chen et al. 2011). A number of studies used single layer, whereas others used multilayer networks. However, the overall accuracy of using ANN to predict preterm birth on these different studies was from 72% to 97% (Cheung 2012; Chen et al. 2011; Diab, El-Merhie & El-Halabi 2010;Moslem, Diab, et al. 2012).

6.10 Chapter Summary

Medical Time series analysis has recently gained much attention from scientists and researchers in medical society. This chapter presents the issues that are related to medical time series classification including pre-processing, filtering, feature extraction and classification. Neural networks have been applied widely to analysis of biomedical signals. They are ideal for recognising diseases. There are a number of advantages of using the ANN as a classifier since it has the ability to generalise; it does not require a prior understanding of the data pattern. Furthermore, many existing research works have demonstrated the success of recurrent neural networks to pre-processing or classification of medical data. Chapter 7 will present the simulation results and discussion of our research work.

CHAPTER 7 THE SIMULATION RESULTS

7.1 Introduction

In this chapter, the simulation results using the proposed neural network architectures will be presented. The dynamic self-organised neural network inspired by the immune algorithm using the concept of the Jordan network architecture is utilised for the prediction of financial time series such as the exchange rate time series and the oil prices.

The dynamic self-organised neural network inspired by the immune algorithm using the concept of the Elman network architecture is utilised for the classification of term and preterm subject in EHG signals.

7.2 Financial Time Series

Ten noisy financial time series were studied and used to evaluate the performance of the DSMIA neural network architectures. The results of the proposed DSMIA have been benchmarked with the other standard neural network architectures. Figure 7.1 shows the proposed schematic for forecasting financial time series.

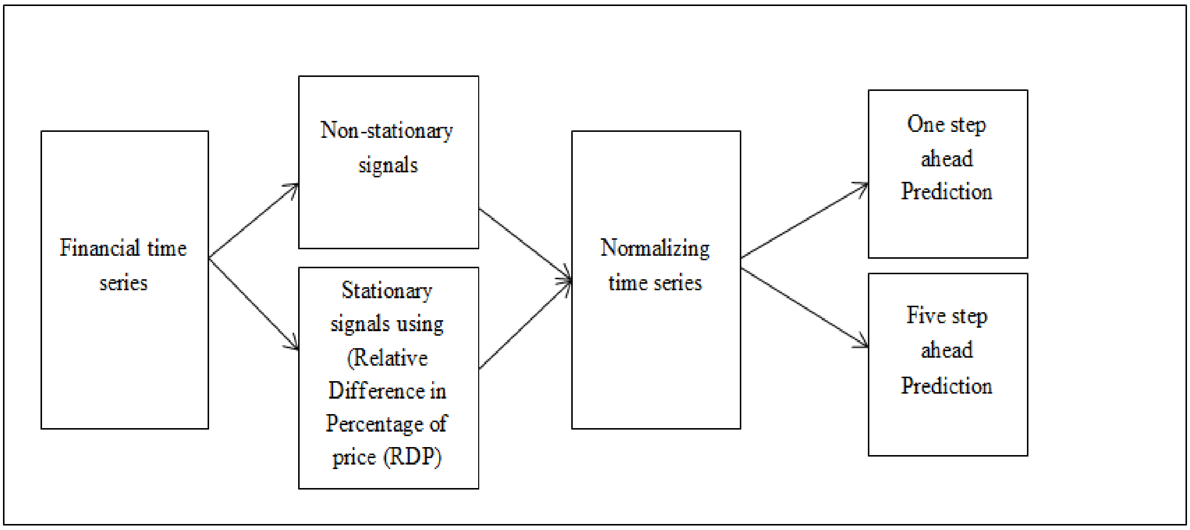


Figure 7.1: Proposed method for the prediction of financial time series

7.2.1 Experimental Designs

The procedures of financial time series predication are considered to be challenge. A number of steps must be taken into consideration before applying neural networks for prediction.

These steps are the selection of the input-output variables, the choice of data, the initial weight state, the stopping criterion during the learning phase, the selection of the activation function, selection of the best parameters such as learning rate, momentum term, and the number of nodes in the hidden layers (Ghazali 2007; Huang, Lai, and Nakamori 2004). All these factors are very important for improving the accuracy of the forecasting. This section will provide an overview of the pre-processing and methodology that has been used to apply the proposed DSMIA for financial time series prediction.

7.2.2 Financial Time Series Data Sources

Three different types of financial time series are applied in this research work: the exchange rate prices, stock opening and closing prices, and the oil price. The exchange rate time series and the stock prices are daily time series for the period from 1st July 2002 to 11th November 2008, giving 1,605 trading days, as shown in Table 7.1. The oil price data is monthly data and covers the period between 1st January 1985 and 1st November 2008, with a total of 389 trading months. The source of the data can be found at http://www.economagic.com/ecb.htm.

Table 7.1: Time Series Data Used

No	Time Series Data	Total
1	US dollar to EURO exchange rate (USD/EUR) 01/07/2002 - 13/11/2008	1607
2	US dollar to UK pound exchange rate (USD/UKP) 01/07/2002 - 13/11/2008	1607
3	Japanese yen to US dollar exchange rate (JPY/USD) 01/07/2002 - 13/11/2008	1607
4	Dow Jones Industrial Average stock opening price (DJIAO) 01/07/2000 - 11/11/2008	1605
5	Dow Jones Industrial Average stock closing price (DJIAC) 01/07/2000 - 11/11/2008	1605
6	Dow Jones Utility Average stock opening price (DJUAO) 01/07/2000 -11/11/2008	1605
7	Dow Jones Utility Average stock closing price (DJUAC) 01/07/2000 -11/11/2008	1605
8	NASDAQ composite stock opening price (NASDAQO) 01/07/2000 -12/11/2008	1606
9	NASDAQ composite stock closing price (NASDAQC) 01/07/2000 - 12/11/2008	1606
10	Oil price of West Texas Intermediate crude (OIL) 01/01/1985 -01/11/2008	389

Since most of the published papers about financial time series prediction have focused on exchange rate prediction (Ghazali et al. 2009; Mahdi et al. 2009; Al-Aweel, Krishnamurthy, Hausdorff, Mietus, Ives, Blum, Schomer 1999; Dunis & Williams 2002), this research has used exchange rate time series, as shown in Table 7.1. The foreign exchange market is considered to be the largest market, with more than $1 trillion traded every day (Yao & Tan 2000; Huang, W., Lai, K.K., and Nakamori 2004). The US dollar is the most significant currency in the market and it has been used as a reference currency. Another time series used in this research is the West Texas Intermediate (WTI) crude oil prices. Crude oil is well known as a central source of energy. The future oil price has a great impact on governments, industries and companies' activities (Alexandridis & Livanis 2008).

7.2.3 Modelling DSMIA for Prediction

In this section the structure of the neural network models will be explained. This section will focus on the determination of the number of variables that have been used as input and output for the neural network, and the pre-processing and scaling methods that have been used in this thesis. Furthermore, the section will introduce the quality measures that have been used to evaluate the performance of the neural network for prediction.

7.2.3.1 Data Preparation

The selection of a suitable forecasting horizon is the first step that must be taken before financial forecasting can begin. From a trading principle, Cao and Tay (2003a) asserted that a long forecasting horizon could avoid over-trading resulting in extreme transaction rates. However, if the forecasting horizon is very long, that might increase the complexity of the forecasting procedure. On the other hand, predictors have claimed that the forecast horizon must be sufficiently short as the persistence of financial time series is for a limited period (Ghazali 2007). However, using a very short forecast horizon might not be sufficient and might decrease the forecasting accuracy; this is due to the large amount of noise existing in the financial time series (Jurik 1999). As Thomason (1999a) recommended in his work, the optimal choice of forecasting horizon for the daily data is five days. From the trading and prediction view, this experiment applies two forecast horizons: one day ahead and five days ahead predictions

7.2.3.2 Data Pre-processing

Financial time series are a non-stationary, high noise type of data. The noise variables in data are identified as harmful variables in neural network learning (Mahdi 2010). Therefore, it is crucial to have a pre-processing method to deal with data before passing them to the neural network. The original raw non-stationary signals are transformed into stationary signals before sending them to the neural network, by a transformation technique known as Relative Difference in Percentage of price (RDP) (Thomason 1999), which is used by a number of researchers in this field (Sundareshan et al. 1999; Cao & Tay 2003a; Ghazali et al. 2009; Mahdi et al. 2009). It creates a five-day measure of the relative difference in price data. In this transformation, the signal is expected to be more symmetrical and closely follow the normal distribution; consequently it can improve the prediction process (Cao & Tay 2003a). Ghazali et al. (2009) mentioned that RDP helps to reduce the influence of trends on financial time series, smoothing the data and helping to reduce noise. The input variables are computed from four lagged RDP values based on five-day periods (RDP-5, RDP-10, RDP-15, and RDP-20) and one transformed signal (EMA15), which is computed by subtracting a 15-day exponential moving average from the raw signals (Cao & Tay 2003a). The reason for selecting a 15-day average is that, according to a number of studies (Ghazali 2007; Cao & Tay 2001; Ristanoski & Bailey 2011), the best length of the moving day should be more than the prediction horizon. The main reason for applying an exponential moving average of 15 is to maintain useful information contained in the original signal, which might be removed by the RDP method. Furthermore, using EMA to produce input variables can improve the prediction performance (Ghazali 2007; Cao & Tay 2001; Cao & Tay 2003a). The computation of the inputs is presented in Table 7.2.

Table 7.2: The data pre-processed into a stationary series where α is the weight factor which is experimentally determined in these experiments as 0.85, *P(i)* is the values of signal for the *ith* day, and h is a horizon of one or five step ahead prediction.

	Indicator	Calculations
Input variables	EMA15	$P(i) - \overline{EMA_{15}(i)}$ $EMA_n(i) = \frac{\alpha^0 p_i + \alpha^1 p_{i-1} + \alpha^2 p_{i-2} + \ldots + \alpha^{n-1} p_{i-n+1}}{\alpha^0 + \alpha^1 + \alpha^2 + \ldots + \alpha^{n-1}}$
	RDP-5	$(p(i) - p(i-5)) / p(i-5) * 100$
	RDP-10	$(p(i) - p(i-10)) / p(i-10) * 100$
	RDP-15	$(p(i) - p(i-15)) / p(i-15) * 100$
	RDP-20	$(p(i) - p(i-20)) / p(i-20) * 100$
Output variable	RDP+*k*	$(p(i+k) - p(i)) / p(i) * 100$ $\overline{p}(i) = \overline{EMA_3(i)}$

Another pre-processing method applied to the data is scaling. This method has been used in order to reduce the range differences in the data as well as to decrease the computational time. All input and output variables are scaled between upper and lower bounds of the network transfer function. The scaled method is done by using the minimum and maximum normalisation method as follows:

$$N(x) = (max_2 - min_2) * \left(\frac{x - min_1}{max_1 - min_1}\right) + min_2 \quad (7.1)$$

Where $N(x)$ is referring to the normalised value and max_1, min_1 are the minimum and maximum values of the original signal. max_2, min_2 refer to the wanted minimum and the maximum values of the new scaled series and *x* is the original values of the signal.

In this experiment the sigmoid transfer function has been used in the output layer. Therefore, all input and output values of RDP are scaled into the range of [0.2, 0.8], which is same interval that has been used in Mahdi et al. (2009). Consequently, the neural network output values will be closer to the endpoints of the output layer activation function.

7.2.3.3 Selecting Inputs and Output Variables

The amount of input units must be selected carefully. According to Huang et al. (2004), there is a step that must be taken when using a neural network to predict financial time series. This step is determining what and how many variables should be used for the input and output of the neural network. As has been discussed before, the type of data used in this research are univariate data; so, variables taken into account are historical data. Therefore, neural network inputs for this type of data are represented as lagged values and the output values are corresponding to the future value. The input layer will hold the time series data points of *N* days, and the output layer will produce the prediction values for next days *"(N + 1)*th" day. As has also been discussed before, using too many past periods as input will lead to much difficulty in training the ANN, whereas too few periods may not be enough to train the ANN. In this research work, the number of inputs is set to five, as recommended by a number of studies (Mahdi et al. 2009; Ghazali et al. 2009). Since the number of the output units in time series forecasting is related to the forecasting horizon, two types of forecasting horizon have been used in this research, the one day ahead and five days ahead predictions.

7.2.3.4 Selection of Neural Network Parameters

In terms of selecting the number and size of the hidden layers in the ANN and other neural parameters such as learning rate and momentum, it has been recommended that trial and error is needed to determine the optimal structure of the neural network. The best way to evaluate the performance of the ANN learning is to split the raw data not only into training and test sets, but also a separate validation set. Therefore, the time series was divided into three parts, the first 50% of the data are used for the training set; the second 25% for the validation set, used to estimate the neural network parameters, and the third 25% is selected for testing the performance of the network. The testing period is kept for final performance evaluation and comparison. This has been done in order to evaluate the accuracy of the model for understanding the past, present and future data sets. The initial weights are selected between [-0.5, 0.5]. The momentum term and the learning rate parameters are selected experimentally. The best values for these parameters are based on the training data set. The values of α and β are model parameters that respectively impact the effect of the input and the context unites, which are considered to be vital parameters for the network performance. The selection of α is a sign of the value of the current entry and β parameters determine the importance of the previous output. To insure the stability, the β value must be very small, according to Voegtlin (2002). From the experiments, the effects of the context vector must be small; thus the value of β parameters must be small, which is selected as β = [0.1 0.3], while α = [0.95 2].

The main concern with financial time series prediction is not to evaluate the predictive capability of the network, but to focus on the profitable value that the neural network achieved, consequently, the neural network structure achieving the best accuracy of financial functions on the test data set is chosen as the best model.

7.2.4 Performance Measures

There are different evaluation functions that have been applied to estimate the network performance; some of them are related to financial measurement and some of them are statistical methods. The performance of the proposed network is measured with four financial metrics (Dunis & Williams 2003) and five statistical metrics (Cao & Tay 2003b) which measure the accuracy of the prediction signal.

7.2.4.1 The Statistical Measures

These have been used to evaluate the performance of the neural network prediction model, which includes the functions listed below:

1. Normalised Mean Square Error (NMSE)

   NMSE is an estimator of the overall deviations between target and predicted values. The lower NMSE values show that the prediction signals closely follow the trend of the actual target.

$$NMSE = \frac{1}{\sigma^2 * N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (7.2)$$

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y}_i) \qquad (7.3)$$

Where N is the total number of data patterns, y and $\hat{y}$.

2. Mean Square Error (MSE)

MSE is computed by finding the square of the error between the target and predicted values. This measure is very popular for evaluating the forecasting ability of a neural network.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (7.4)$$

3. Mean Absolute Error (MAE) Mean

The MAE produces the mean absolute error value of the deviation between the actual and forecasted values. The smaller the value of MAE, the closer the predicted time series to the actual values.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}| y_i - \hat{y}_i |^2 \qquad (7.5)$$

4. Correct Directional Change (CDC)

CDC determines the ability of a prediction model to accurately forecast the subsequent actual change of a forecast variable.

$$CDC = \frac{1}{n}\sum_{i=1}^{n} d_i , d_i = \begin{cases} 1 & if \ (y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0, \\ 0 & otherwise \end{cases} \qquad (7.6)$$

5. Signal to Noise Ratio (SNR)

It is a measurement used to compare the amount of information on a desired signal to the amount of background noise. The highest ratio of SNR means the signal levels are higher than the noise level. SNR is measured in dB.

$$SNR = 10 * \log_{10}(sigma) \qquad (7.7)$$

$$sigma = \frac{m^2 * n}{SSE} \qquad (7.8)$$

$$SSE = \sum_{i=1}^{n}(y_i - \widehat{y)_i} \qquad (7.9)$$

$$m = \max(y_i) \qquad (7.10)$$

7.2.4.2 The Financial Functions

Since the main aim of financial time series forecasting is to achieve the trading profits rather than to evaluate the forecasting accuracy, the financial measures must be used to evaluate the performance of the neural network for predicting the financial time series. In order to measure profit generated from the network's prediction, a simple trading strategy has been used in this research, which is based on: if the network forecasts a positive change for the next k-day price for non-stationary or stationery signal, a buy signal is sent, otherwise a sell signal is sent if the network forecast negatives change for the next day.

The application of these measures may be a better standard for determining the quality of the forecasts (Dunis & Williams 2002). These measurements are listed below:

1. Annualised Return (AR)

   AR measures the ability of neural networks to be used as traders. It is a scaled calculation of the observed change in time series value. The higher value of AR shows the better prediction model.

$$AR = \frac{\mathrm{Pr}ofit}{All\ profit} * 100 \tag{7.11}$$

$$Pofit = \frac{252}{n} * CR, \qquad CR = \sum_{i=1}^{n} R_i$$

$$\mathrm{R_i} = \begin{cases} +\left|y_i\right| & if\ (y_i)(\hat{y}_i) \geq 0, \\ -\ \left|y_i\right| & otherwise \end{cases} \tag{7.12}$$

$$All\ profit = \frac{252}{n} * \sum_{i=1}^{n} abs(R_i) \tag{7.13}$$

2. Maximum Drawdown (MDD)

   The maximum drawdown (MDD) measure is another financial evaluation which refers to the maximum drop of the asset price within a given time period. It measures less risk or less loss, so a high value is required (Vecer & York 2006).

$$MD = \min\left(\sum_{t=1}^{n} \left(CR_t - \max\left(CR_1, ..., CR_t \right) \right) \right) \tag{7.14}$$

$$CR_t = \sum_{i=1}^{t} R_i, t = 1,...,n$$

$$R_i = \begin{cases} +|y_i| & if \ (y_i)(\hat{y}_i) \geq 0, \\ -|y_i| & otherwise \end{cases} \tag{7.15}$$

3. Annualised Volatility (AV)

   VOL is the function that estimates the investment risk and profit possibilities; so, a small value of volatility is considered to be a better result.

$$VOL = \sqrt{252} * \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(R-\bar{R})^2} \tag{7.16}$$

4. Sharp Ratio (SR)

   It is a function to calculate the risk-adjusted return. SR determines the relation between the Annualised Return and the volatility evaluation. In this measurement, a high value is demanded. The higher value of SR shows the higher the return and the lower the volatility.

$$SR = \frac{AR}{AV} \tag{7.17}$$

7.2.5 Regularised DSMIA Network (R-DSMIA)

In this section, the regularisation technique has been used on DSMIA to develop the performance of the proposed network. The regularisation has been addressed to improve the generalisation and to solve the over-fitting problem; also, the elimination of the weights leads to a reduction in the complexity of the training network and makes the learning process easier (Siwek & Osowski 2001). the procedure of regularization is process that attempt to smoothing the cost function and reducing variance by pruning some weights besides keeping the functional capability necessary to solve the problem (Larsen et al., ,1998). This is the procedure of regularisation. Regularisation is the technique of adding a penalty term Ω to the error function, which can help obtain a smoother network mapping. It is given by:

$$E_{reg} = E_{std} + \lambda\Omega \tag{7.18}$$

Where E_{std} represents one of the standard error functions such as the sum-of-squares error and the parameter λ controls the range of the penalty term Ω in which it can influence the form of the solution. The network training should be implemented by minimising the total error function Ereg (Bishop 1995). One form of regularisation is called weight decay (Krogh & Hertz 1995). It has been shown that it can help to avoid over-fitting the network to training data; as such, improving the network (Duda et al., 2001). This form is based on the sum of the squares of the adaptive parameter in the network.

$$\Omega = \frac{1}{2}\sum_i w_i^2 \tag{7.19}$$

The idea is that every w_i weight, once updated, is simply decayed or shrunk as follows:

$$w^{new} = w^{old}(1 - \lambda) \tag{7.20}$$

Where $0<\lambda< 1$. The weight decay is performed by adding a bias term to the original objective function E_{std}; thus the weight decay cost function is determined as follows (Steurer 1993):

$$E_{reg} = E_{std} + (\lambda/2)\, B \tag{7.21}$$

Where λ is the weight decay rate and B represents the penalty term. The simplest form of calculating the penalty term B is:

$$B = \sum W^2_{ij} \tag{7.22}$$

Where w_{ij} is the weight connections between the i^{th} units and j^{th} nodes in the next layer. In the R-DSMIA network the weight decay was used to adjust the weights between the hidden nodes and output units. The change of weights using the weight decay method could be calculated as follows:

$$\Delta w_{ojk} = -\eta\frac{\partial E}{\partial w_{ojk}} - \eta\lambda w_{ojk} \tag{7.23}$$

Where, Δw_{ojk} is the updated weights that connected hidden units and output units. This will improve the neural network performance. Since the number of weights to be computed in the DSMIA network can be quite large in some problems, there is a need to eliminate some weights from the network without losing the functional ability required to solve the problem. The R-DSMIA network is used to examine the effect of the regularisation technique and to enhance the performance of the DSMIA network in time series prediction.

7.2.6 Simulation Result and Analysis

In this section, the simulation results using DSMIA with other benchmark networks are presented. In this research work, the networks were tested in two different sets of signals, stationary and non-stationary. One and five step ahead predictions of financial time series were utilised. In the case of the non-stationary signal, all the data presented in Table 7.1 are passed directly to the neural network. On the other hand, for the stationary signal, the original signals have been transformed using RDP.

The main interest in those experiments is to consider the profitability value of the network and consequently the network that generates the highest percentage of Annualised Return (AR) is considered the best model. In contrast, for the Annualised volatility (AV) a small value of volatility is considered as a better result. In the sharp ratio (SR) measurement, the high value is demanded. The main reason for considering the financial measurement to evaluate the predicting models is that, from a trading aspect, the models must be generating profit. Therefore, it is significantly important for the predicting model such as a neural network to predict the correct direction change of signals.

7.2.6.1 Prediction of Stationary Signals

This section represents the experiments result of the stationary prediction of the ten financial time series signals. The original ten signals have been transformed to stationary signals. The simulation results for one step ahead and five step ahead predictions are discussed.

One step ahead Prediction using Stationary Signals

Tables 7.3 to 7.8 summarise the average result of 30 simulations obtained on testing data sets from ten signals using six types of neural networks. These networks are MLP, SONIA, DSMIA, R-MLP, R-SONIA and R-DSMIA. The simulation results are based on achieving the best profit values, which are generated by using different structures of neural networks. The best neural network model for financial forecasting is selected based on generating the highest percentage of annualised return.

Simulation Result

In the case of evaluating the percentage of Annualised Return (AR), the proposed network successfully made the best profit return when compared with MLP and SONIA networks, as is presented in Tables 7.3 to 7.8. The result of the Annualised return from Tables 7.5 and 7.8 showed that the proposed DSMIA and R-DSMIA achieved the highest profit on return compared to all networks, except the US/EU exchange rate and the NASDAQC stock closing price. In the case of predicting the US/EU and NASDAQC signals, R-SONIA has achieved the highest profit return, as shown in Table 7.7.

In addition, the comparison between the performance of the DSMIA network and regularisation technique with the R-DSMIA has achieved the best profit return on five time series namely NASDAQO, DJIAO, DJIAC, DJUAO and OIL compared to DSMIA. For the rest of the signals, DSMIA has achieved a better result than R-DSMIA network.

In terms of the other financial measures such as the maximum drawdown and volatility, it can be observed that DSMIA achieved the highest value of maximum drawdown when predicting the USD/UKP exchange rate, DJUAC stock closing prices, and OIL time series compared to MLP and SONIA networks. The MLP and SONIA produced better results in four time series: USD/EUR, NASDAQO, NASDAQC, and DJIAC.

In addition, the comparison between the performance of the DSMIA network and regularisation techniques with DSMIA network based on the maximum drawdown value, from Tables 7.5 and 7.8, it can be observed that R-DSMIA has got the best profit return on six time series: USD/UKP, USD/EUR, NASDAQO, DJIAO, DJIAC, and OIL compared to DSMIA. For the rest of the signals, DSMIA has achieved a better result than the R-DSMIA network.

For the volatility values, the comparisons between the neural networks are based on achieving the lowest value of volatility. It can be observed that the proposed DSMIA network and R-DSMIA have lower values than other networks expect for USD/EUR time series. The R-SONIA has achieved better value on volatility when predicting the USD/EUR time series.

In term of evaluating the Sharp Ratio measure, the higher value of SR is desired. From Tables 7.3 to 7.8, it can be noticed that the best value of SR was produced by the proposed DSMIA and R-DSMIA networks, except for USD/EUR, NASDAQC and OIL time series. For the Signal to Noise Ratio criterion, the higher value of SNR demonstrates that the predictor has read signals with less noise re-write. Tables 7.3 to 7.8 show that the proposed DSMIA and R-DSMIA obtained the best values when used to predict USD/UKP, DJIAC, DJUAO and

DJUAC time series. The R-SONIA has produced the best SNR value on JPY/USD, USD/EUR, NASDAQO and OIL time series. The MLP network achieved the highest value of SNR on DJIAO time series.

For the correct directional change, the R-MLP produced the best value of CDC when forecasting USD/UKP, JPY/USD, NASDAQC, DJIAC, DJUAO and DJUAC time series. The SONIA achieved the best value of CDC on USD/EUR and Oil time series. However, from Table 7.7, it is clear that the R-SONIA obtained the best value on two types of stock opening prices, which are NASDAQO and DJIAO signals, compared to other networks.

In the case of forecasting error measures NMSE, MSE and MAE, it can be observed that the DSMIA and R-DSMIA networks outperform all other networks with the lowest forecasting errors, while MLP produced the highest value of forecasting error measures except for DJIAO time series. These results confirm the forecasting ability of the proposed networks. Figure 7.2 illustrates the result of Annualised return measures, which has been forecasted by six neural network architectures.

Table 7.9 represents the number of hidden units that have been used on neural networks to generate the better prediction results for the AR. The simulation results indicated that using 12 to 22 hidden nodes in DSMIA and R-DSMIA networks can obtain the best result of profits. On the other hand, MLP needs five to eight hidden units to produce the best average results. However, it can be observed from Table 7.9 that SONIA requires a higher number of hidden nodes, which is between 12 to 25 units.

Table 7.3: The result of one step ahead prediction using stationary signals on MLP network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	65.14634	-1.4304	4.526048	14.50009	20.35	67.97	0.75775	0.005956	0.059229
JPY/USD	64.72067	-1.5806	5.717462	11.36989	21.75	62.59	0.781611	0.003491	0.042648
USD/EUR	69.91986	-1.7923	4.666936	15.00254	22.29	62.95	0.610187	0.003835	0.046202
NASDAQO	60.5218	-4.8447	13.27316	4.575426	23.7	65.87	0.784347	0.002685	0.035463
NASDAQC	62.32483	-4.8762	12.58836	4.955069	22.83	64.53	0.663549	0.003322	0.041591
DJIAO	59.49555	-4.4906	11.18889	5.322315	24.45	63.36	0.646463	0.002166	0.023119
DJIAC	58.85693	-2.9165	11.27457	5.22938	23.26	64.88	0.844715	0.002789	0.037976
DJUAO	52.97216	-6.1459	12.48506	4.267582	23.84	63.34	0.831784	0.002309	0.033269
DJUAC	52.01663	-5.6157	12.56166	4.169173	23.9	64.61	0.833803	0.002318	0.033309
OIL	51.19201	-19.024	67.3681	0.766643	21.03	61.36	1.016665	0.00283	0.042386

Table 7.4: The result of one step ahead prediction using stationary signals on SONIA

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	73.15769	-1.51535	4.346859	16.83866	21.93	66.09	0.522666	0.004108	0.048544
JPY/USD	76.19371	-1.24388	5.424529	14.04827	23.41	62.35	0.530438	0.002369	0.034869
USD/EUR	71.00268	-2.69518	4.63746	15.34295	23.04	63.07	0.506081	0.00318	0.039536
NASDAQO	62.66176	-8.84982	13.20117	4.742449	24.14	66.82	0.70604	0.002417	0.030737
NASDAQC	63.35985	-8.32449	12.53349	5.057593	22.75	62.39	0.673079	0.00337	0.038363
DJIAO	63.71905	-6.60512	11.00119	5.800506	23.61	61.92	0.785339	0.002631	0.032257
DJIAC	62.29099	-7.99077	11.12426	5.610039	23.53	62.98	0.792066	0.002616	0.032189
DJUAO	67.0127	-3.55542	11.93661	5.614729	25.02	60.48	0.631187	0.001752	0.028335
DJUAC	68.02663	-4.17998	11.94297	5.678424	25.03	59.95	0.640598	0.001718	0.02866
OIL	72.16195	-8.32751	60.15546	1.20381	22.2	62.76	0.78583	0.002187	0.039051

Table 7.5: The result of one step ahead prediction using stationary signals on DSMIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	**76.401946**	-1.29011	4.256646	17.951165	22.43	65.88	0.465058	0.003655	0.046061
JPY/USD	**76.287889**	-1.50278	5.420273	14.083690	23.36	59.77	0.537780	0.002402	0.034772
USD/EUR	72.941440	-1.91315	4.590001	15.897520	23.08	62.80	0.501631	0.003153	0.039662
NASDAQO	66.466713	-6.89071	12.999914	5.117712	24.49	67.62	0.650886	0.002228	0.029922
NASDAQC	64.77886	-6.40323	12.450211	5.207525	22.86	62.34	0.6562777	0.0032860	0.0382681
DJIAO	64.859029	-7.57360	10.951396	5.928563	23.72	60.38	0.765907	0.002566	0.031713
DJIAC	66.071119	-6.60576	10.952866	6.038112	23.67	61.65	0.766192	0.002530	0.031569
DJUAO	69.986696	-3.66422	11.787191	5.937937	25.31	60.56	0.591318	0.001641	0.027465
DJUAC	**70.593948**	-3.41091	11.802643	5.981827	25.35	61.11	0.596172	0.001658	0.027371
OIL	73.352611	-7.08040	59.745353	1.228245	22.69	61.67	0.692816	0.001928	0.035963

Table 7.6: The result of one step ahead prediction using stationary signals on R-MLP network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	69.87597	-1.14692	4.434009	15.76153	21.64	68.42	0.56236	0.00442	0.05143
JPY/USD	69.11974	-1.38279	5.621507	12.30291	22.5	63.74	0.656447	0.002932	0.03997
USD/EUR	72.43294	-1.3342	4.604267	15.73393	23.38	62.78	0.469033	0.002943	0.04065
NASDAQO	61.31236	-4.58009	13.23235	4.649596	23.68	67.07	0.788035	0.002697	0.03559
NASDAQC	63.32054	-5.12025	12.53594	5.053141	23.22	65.07	0.605016	0.003029	0.0391
DJIAO	60.396	-2.82338	11.1544	5.416199	22.98	56.24	0.909105	0.003046	0.03947
DJIAC	57.27957	-2.98009	11.34621	5.051273	23	66.01	0.897256	0.00263	0.03919
DJUAO	53.52269	-3.85035	12.51297	4.28039	24.27	64.33	0.751373	0.002086	0.03232
DJUAC	56.41693	-3.54951	12.45137	4.531404	24.5	64.67	0.724093	0.002013	0.03166
OIL	54.96625	-17.3305	65.8069	0.857579	21.46	61.33	0.935372	0.001703	0.032536

Table 7.7: The result of one step ahead prediction using stationary signals on R-SONIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	73.63894	-1.14585	4.334805	16.98832	22.28	66.74	0.481713	0.003786	0.047653
JPY/USD	75.08265	-1.37047	5.457777	13.75739	23.4	62.46	0.531308	0.002373	0.035949
USD/EUR	**74.79281**	-1.24481	4.539692	16.47533	23.42	61.54	0.463063	0.00291	0.039707
NASDAQO	63.4887	-4.19713	13.15954	4.82642	24.59	68.73	0.635346	0.00217	0.030901
NASDAQC	**66.11718**	-4.14573	12.37758	5.341885	23.18	62.7	0.609063	0.00305	0.037929
DJIAO	62.14083	-3.10446	11.07906	5.610178	23.97	63.6	0.721802	0.002418	0.03321
DJIAC	63.69784	-5.62573	11.06588	5.760461	23.95	63.84	0.719279	0.002375	0.032265
DJUAO	69.75802	-3.29048	11.79905	5.912536	25.13	60.7	0.615357	0.001708	0.028379
DJUAC	69.74614	-3.3862	11.84642	5.888317	25.12	61.1	0.627945	0.001746	0.028874
OIL	73.578	-6.76682	59.63936	1.234447	23.23	61.73	0.611774	0.001703	0.032536

Table 7.8: The result of one step ahead prediction using stationary signals on R-DSMIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	75.552307	-1.145854	4.281168	17.649052	22.45	65.83	0.462755	0.003637	0.046711
JPY/USD	76.199110	-2.006950	5.424258	14.050502	23.14	59.61	0.565199	0.002525	0.034462
USD/EUR	72.223068	-1.870829	4.607536	15.693625	23.02	62.58	0.508964	0.003199	0.039765
NASDAQO	**67.736511**	-6.679803	12.933600	5.240688	24.56	67.42	0.641108	0.002194	0.029937
NASDAQC	61.817786	-8.434717	12.618953	4.900625	22.70	62.15	0.681348	0.003412	0.038382
DJIAO	**69.501792**	-4.122863	10.730026	6.478727	24.10	62.51	0.700827	0.002348	0.031286
DJIAC	**67.27255**	-5.997100	10.891160	6.185754	23.97	63.22	0.716507	0.002366	0.031486
DJUAO	**70.250021**	-3.789960	11.772569	5.968616	25.10	61.11	0.619493	0.001720	0.028111
DJUAC	69.652083	-3.479243	11.850855	5.878571	25.24	59.92	0.610930	0.001699	0.028103
OIL	**74.342906**	-6.750138	59.281248	1.255279	23.17	59.09	0.619850	0.001725	0.032856

Table 7.9: Number of hidden nodes in the MLP, SMIA and DSMIA for one step ahead stationary signals

Time Series	one step for stationary signal		
	MLP	SONIA	DSMIA
USD/UKP	8	12	12
JPY/USD	8	24	21
USD/EUR	8	19	17
NASDAQO	8	18	16
NASDAQC	8	21	19
DJIAO	8	19	16
DJIAC	7	19	16
DJUAO	8	25	22
DJUAC	8	23	20
OIL	5	13	13

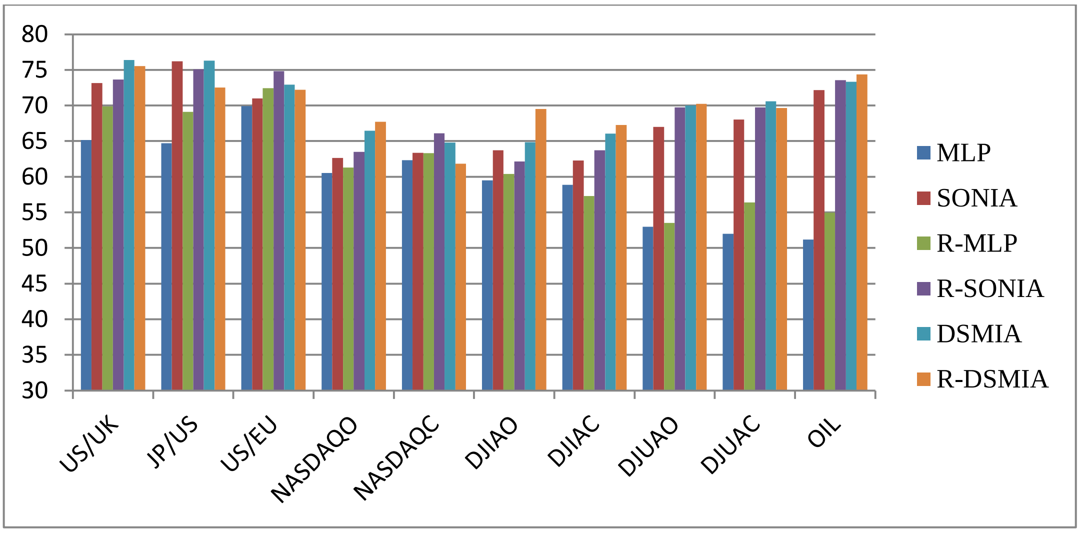


Figure 7.2: Annualised return on stationary signals for the prediction of one step ahead

Five Step ahead Prediction using Stationary Signals

In this section the result of the simulation for five step ahead forecasting using stationary signals are presented. The main objectives of these experiments are to show the predictive ability of the proposed networks and the benchmarks networks to forecast five steps ahead. Tables 7.10 to 7.15 represent the average result of simulation obtained from the test data set using 10 financial signals for six different neural network architectures.

Simulation Result

In terms of evaluating the performance of the network based on Annualised return measures, the proposed DSMIA and R-DSMIA have obtained the highest percentage compared to the benchmark networks except NASDAQC time series. The R-SONIA has achieved the best values of AR on this time series. From Tables 7.10 to 7.12, it can be shown that the DSMIA network outperformed the SONIA and MLP networks. These results demonstrate that the DSMIA achieved the best profits on average for all 10 time series when compared to MLP and SONIA networks. On other hand, from Tables 7.12 and 7.15, the comparison between the performance of DSMIA and R-DSMIA networks showed that using regularisation techniques on the R-DSMIA network has significantly improved the performance of the DSMIA. The R-DSMIA successfully obtained the best profits in comparison to the DSMIA network when forecasting USD/UKP, USD/EUR, NASDAQO, DJIAO, DJIAC, DJUAO, DJUAC and OIL time series.

Table 7.10: The result of five step ahead prediction using stationary signals on MLP network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	84.72936	-3.15533	16.24499	5.246373	23.14	63.7	0.0660456	0.003232	0.0411
JPY/USD	81.19235	-3.54864	17.56791	4.62544	22.56	62.18	0.377454	0.002881	0.0407
USD/EUR	91.46321	-1.78913	15.69251	5.829595	26.71	64.65	0.175615	0.001367	0.02773
NASDAQO	77.57114	-10.1308	37.68809	2.063196	24.46	61.47	0.481259	0.001557	0.02869
NASDAQC	83.09106	-7.34318	36.53018	2.278035	24.93	59.92	0.376718	0.001462	0.02844
DJIAO	74.77388	-8.24405	33.66308	2.225729	23.73	61.92	0.668271	0.002005	0.03216
DJIAC	65.96346	-13.0605	35.11867	1.888711	23.09	61.45	0.76142	0.002287	0.03444
DJUAO	74.07559	-11.1968	38.27525	1.941499	24.53	60.86	0.543456	0.00145	0.02664
DJUAC	68.93831	-20.4777	39.02029	1.786946	24.21	59.66	0.58564	0.001572	0.02748
OIL	75.89575	-8.03638	194.2871	0.390889	20.24	56.88	1.171494	0.003058	0.04623

Table 7.11: The result of five step ahead prediction using stationary signals on SONIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	88.15049	-3.2779	15.87492	5.561351	24.98	65.49	0.420061	0.002056	0.28749
JPY/USD	87.17208	-2.72278	16.76482	5.199892	23.02	63.16	0.429469	0.002515	0.03654
USD/EUR	91.84363	-1.39054	15.6362	5.873841	23.41	65.09	0.375141	0.00292	0.033599
NASDAQO	85.16167	-6.19597	35.93796	2.369793	25.22	62.49	0.38383	0.001242	0.023329
NASDAQC	85.02744	-7.36317	36.06489	2.357881	24.33	59.48	0.416113	0.001615	0.002624
DJIAO	85.3598	-8.41446	31.63171	2.699558	24.81	62.91	0.509205	0.001527	0.023769
DJIAC	84.89345	-8.37277	31.78481	2.672691	24.68	62.99	0.52398	0.001574	0.023998
DJUAO	81.28562	-7.78359	37.03390	2.564821	26.28	61.72	0.354277	0.000945	0.02222
DJUAC	86.66031	-3.70212	36.27336	2.389255	27.27	62.97	0.280906	0.000754	0.019388
OIL	91.82156	-12.60818	157.372	0.584465	25.37	60.06	0.357378	0.000933	0.025845

Table 7.12: The result of five step ahead prediction using stationary signals on DSMIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	89.047661	-2.963665	15.757916	5.661860	24.85	65.93	0.429909	0.002104	0.028496
JPYUSD	**87.201969**	-2.722772	16.760712	5.202832	22.30	63.48	0.388892	0.002968	0.039623
USDEUR	92.028790	-1.432039	15.607786	5.896499	23.38	65.23	0.377785	0.002941	0.033627
NASDAQO	86.323692	-6.109268	35.637459	2.422628	24.93	61.85	0.411139	0.001331	0.023618
NASDAQC	85.105993	-7.658861	36.041823	2.361925	24.00	59.39	0.449935	0.001747	0.026903
DJIAO	86.629872	-9.305728	31.347378	2.766082	24.72	63.27	0.520153	0.001560	0.023078
DJIAC	86.106254	-9.268669	31.426138	2.754751	25.62	63.54	0.432033	0.001298	0.022549
DJUAO	87.135184	-3.357676	35.864341	2.429673	27.10	62.35	0.292279	0.000780	0.019636
DJUAC	87.517079	-3.410785	36.089621	2.425167	27.36	62.41	0.274922	0.000738	0.019372
OIL	92.039333	-11.644154	156.745293	0.588284	25.02	61.21	0.387805	0.001012	0.026859

Table 7.13: The result of five step ahead prediction using stationary signals on R-MLP network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	87.1223	-2.45425	16.01374	5.441671	26.59	66.02	0.28828	0.001411	0.02807
JPYUSD	83.74337	-2.72277	17.240726	4.857615	23.46	63.58	0.297567	0.002271	0.03626
USDEUR	90.66374	-2.555499	15.812136	5.735123	25.87	64.57	0.214152	0.001667	0.03039
NASDAQO	77.94086	-9.173755	37.578259	2.079828	25.27	61.52	0.392295	0.00127	0.02618
NASDAQC	84.49206	-7.024514	36.195788	2.335423	25.5	60.69	0.322956	0.001254	0.02656
DJIAO	74.35885	-8.166544	33.752804	2.20523	24.04	62.12	0.620153	0.00186	0.03118
DJIAC	69.99318	-10.711946	34.536964	2.032267	23.77	62.3	0.65989	0.001982	0.03222
DJUAO	76.14721	-12.706323	37.978064	2.005844	25.07	62.05	0.471418	0.001258	0.02555
DJUAC	75.9389	-13.174437	38.347615	1.98104	24.94	61.1	0.48512	0.001302	0.02587
OIL	81.74823	-57.254185	180.79057	0.46131	22.02	54.7	0.851406	0.00222	0.03846

Table 7.14: The result of five step ahead prediction using stationary signals on R-SONIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	90.07944	-1.77722	15.64255	5.758673	25.68	66.16	0.354363	0.001734	0.027244
JPY/USD	86.87133	-2.72277	16.80812	5.168465	23.5	64.07	0.29506	0.002252	0.035276
USD/EUR	91.62076	-1.35824	15.67015	5.846852	23.61	65.02	0.358039	0.002787	0.033194
NASDAQO	85.4554	-6.3989	35.86336	2.382872	23.82	61.59	0.37723	0.001221	0.023468
NASDAQC	**86.29109**	-7.08089	35.73558	2.414784	24.67	59.75	0.386041	0.001499	0.026094
DJIAO	83.8411	-7.00081	31.95809	2.623515	24.99	62.31	0.487654	0.001463	0.023946
DJIAC	85.13061	-6.72123	31.74464	2.681791	24.99	62.86	0.486933	0.001463	0.023834
DJUAO	86.84489	-3.4824	35.92529	2.417534	27.42	62.59	0.270406	0.000721	0.019334
DJUAC	86.55567	-4.75658	36.2965	2.384731	27.03	61.98	0.296947	0.000797	0.020296
OIL	91.28559	-13.3041	158.8773	0.575535	25.36	60.03	0.358022	0.0000934	0.025879

Table 7.15: The result of five step ahead prediction using stationary signals on R-DSMIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	**90.313820**	-1.4801296	15.612071	5.78492965	25.48	66.56	0.37023	0.00181	0.02693
JPY/USD	86.89212	-2.722772	16.804883	5.17086372	22.78	65.47	0.3480072	0.0026563	0.037818
USD/EUR	**92.77733**	-1.452800	15.492416	5.988683	23.87	65.51	0.337786	0.002630	0.032606
NASDAQO	**87.03317**	-5.90720	35.453549	2.454971	24.88	60.41	0.415961	0.001346	0.023746
NASDAQC	84.74664	-7.054415	36.136873	2.345472	24.51	60.10	0.400550	0.001555	0.026497
DJIAO	**87.71156**	-5.915513	31.118235	2.819477	25.47	62.94	0.436948	0.001311	0.022956
DJIAC	**88.80179**	-6.29735	30.920888	2.872796	25.26	64.49	0.458317	0.001377	0.022437
DJUAO	**87.57334**	-3.088121	35.770111	2.448449	27.36	63.52	0.275195	0.000734	0.019251
DJUAC	**87.72027**	-2.834078	36.0466	2.433627	27.37	62.01	0.2737	0.000734	0.0193
OIL	**93.43971**	-10.42434	152.72823	0.612524	25.51	59.85	0.345309	0.000901	0.025388

In terms of the maximum drawdown measure, it can be shown that R-DSMIA and DSMIA achieved the highest value of maximum drawdown when predicting the USD/UKP, NASDAQC, NASDAQO, DJIAO, DJIAC, DJUAO and DJUAC time series. While on JPY/USD exchange rate signals, the four neural networks – which are R-MLP, R-SONIA, DSMIA, and R-DSMIA networks – produced the same value on MDD measure. The R-SONIA produced better results when it was used to predict USD/EUR signals. On OIL signal, the MLP has produced the highest value compared to the other networks.

In addition, looking at the comparison between the performance of the DSMIA network and regularisation techniques with the DSMIA network based on the maximum drawdown value, from Tables 7.12 and 7.15 it can be observed that R-DSMIA has lower maximum loss in compression with DSMIA networks when it was used to forecast all signals except USD/EUR exchange rate signal.

For the Annualised Volatility values, the results are shown in Tables 7.10 to 7.15. It can be observed that the proposed DSMIA networks have lower volatility compared to other networks except USD/UKP and NASDAQC time series. The R-SONIA has achieved the best value on volatility when predicting these two time series. However, looking at the comparison between the performance of the DSMIA network and regularisation techniques with the DSMIA network based on AV value, from Tables 5.12 and 5.15, it can be observed that R-DSMIA has lower volatility in comparison with DSMIA networks when forecasting the USD/UKP, USD/EUR exchange rate and when it was used to forecast three time series of stock opening price and closing price namely, NASDAQO, DJIAO DJIAC, DJUAC and DJUAO, in addition to OIL price signal. In the rest of the signals, DSMIA has achieved better value than the R-DSMIA network.

In term of evaluating the Sharp Ratio measure, the result that has been presented in Tables 7.10 to 7.15 shows that the best value of SR was produced by the proposed DSMIA and R-DSMIA networks except for the NASDAQC and DJUAO time series. The R-SONIA network has produced the highest value of SR on NASDAQC time series. For DJUAO stock opening price time series, the SONIA has obtained the best result.

For the Signal to Noise Ratio, Tables 7.10 to 7.15 show that the proposed DSMIA and R-DSMIA obtained the highest value of SNR when they were used to predict DJIAO, DJIAC and DJUAC and OIL time series. The R-SONIA has produced the best SNR value on DJUAO time series. The MLP and R-MLP networks achieved the highest value of SNR on the rest of the time series.

In the correct directional change measurement, the best value was achieved by the R-SONIA network for predicting USD/UKP time series. The DSMIA and R-DSMIA achieved the highest value of CDC when predicting USD/UKP, USD/EU, DJIAO, DJIAC, DJUAO and OIL time series, while SONIA obtained the best value on NASDAQO and DJUAC signals compared to the other networks.

As can be observed from Table 7.16, the network structures for MLP that obtain the best result are realised with networks of five to eight hidden units. Meanwhile, the DSMIA network needs more hidden units, which are selected between 12 to 22 hidden units. For the SONIA network, the best average profit is achieved using 12 to 25 hidden units.

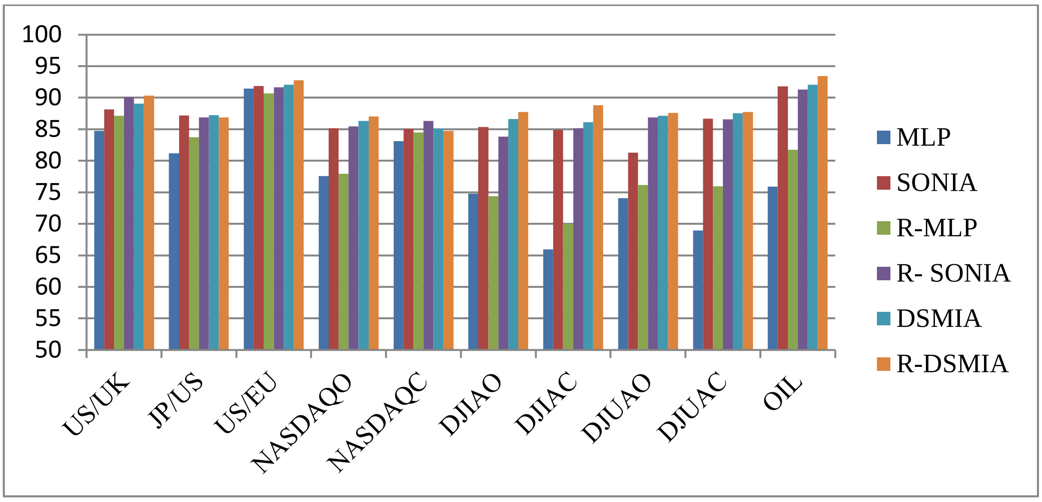


Figure 7.3: Annualised return on stationary signals for the prediction of five steps ahead

Table 7.16: Number of hidden nodes in MLP, SONIA and DSMIA for five step ahead stationary signals

Time Series	Five step for stationary signal		
	MLP	SONIA	DSMIA
USD/UKP	5	12	12
JPYUSD	6	24	21
USD/EUR	8	19	17
NASDAQO	8	18	16
NASDAQC	8	21	19
DJIAO	8	19	16
DJIAC	8	19	16
DJUAO	7	25	22
DJUAC	8	23	20
OIL	8	13	13

7.2.6.2 Prediction of Non-Stationary Signals

The simulation results of the non-stationary prediction of 10 financial time series are presented in this section. For non-stationary signals, all data signals are presented to the networks directly, without any transformation.

One step ahead prediction using non-stationary signals

Figure 7.3 summarises the results achieved from all neural networks. The results were taken from the 30 simulations of testing the data set. The result of the Annualised Returns from

Tables 7.17 to 7.22 shows that the proposed DSMIA and R-DSMIA obtained the best profit return compared to all other network models in seven time series data, namely JPY/USD, NASDAQO, DJIAO, DJIAC, DJUAO, DJUAC and OIL signals. The R-DSMIA achieved the highest profit on JPY/USD, DJIAO, DJUAO and OIL time series. Meanwhile, DSMIA obtained the best average of profit on DJIAC, and DJUAC signals. R-SONIA achieved the best profit on USD/UKP and NASDAQC signals, while R-MLP produced the best result on USD/EUR signals compared to the other networks.

In terms of other financial measures, when measuring the maximum drawdown, it can be detected that the proposed DSMIA and R-DSMIA networks achieved the lowest maximum loss on four time series, which are JPY/USD, NASDAQC, DJUAC and DJIAO signals.

In the cases of evaluating the CDC obtained by all networks, DSMIA achieved the highest values in five out of 10 signals, which are the NASDAQC, DJIAO, DJIAC, DJUAC and DJUAO, while the R-DSMIA produced the highest value on JPY/USD signals. Meanwhile, R-SONIA obtained the best average of CDC on USD/UKP NASDAQO and OIL time series. For demonstration purposes, the Annualised Return achieved by all neural networks is illustrated in Figure 7.4.

Table 7.17: The result of one step ahead prediction using non-stationary signals on MLP network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	6.669613	-14.957188	11.403352	0.585399	16.3	52.78	1.5066392	0.015878	0.117558
JPY/USD	-0.473902	-22.500975	12.802219	-0.037	17.68	47.7	0.592037	0.010555	0.086176
USD/EUR	7.703251	-11.293929	10.106966	-0.762452	13.41	52.34	4.343417	0.029857	0.15849
NASDAQO	-6.27845	-15.470936	30.732116	-0.20439	17.74	47.6	1.368523	0.011319	0.095453
NASDAQC	-10.20173	-70.907413	30.476655	-0.3351	19.38	48.06	0.935094	0.007854	0.078606
DJIAO	-9.977635	-63.928973	27.480595	-0.36342	12.7	48.41	2.909061	0.034668	0.172562
DJIAC	-12.48747	-73.355125	27.88965	-0.44844	13.07	47.66	2.62701	0.032062	0.165596
DJUAO	-7.6654061	-67.988187	29.921184	-0.25648	14.19	49.04	5.944246	0.025324	0.150641
DJUAC	-6.666097	-67.160675	30.590123	-0.21824	14.94	48.58	4.871031	0.01188	0.13792
OIL	4.151457	-13.563781	33.256952	-0.000123	19.08	51.34	0.996917	0.004017	0.05088

Table 7.18: The result of one step ahead prediction using non-stationary signals on SONIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	2.073759	-19.83157	11.39914	0.182488	20.5	50.31	0.626626	0.006592	0.06683
JPY/USD	-8.271959	-32.49779	12.77795	-0.64743	27.04	45.45	0.078149	0.001395	0.02496
USD/EUR	2.414358	-13.22568	10.10582	0.239181	13.97	52.79	4.852491	0.033364	0.1375
NASDAQO	-3.619703	-64.55476	30.73380	-0.1177	19.83	47.77	1.512596	0.012511	0.08605
NASDAQC	3.531254	-49.69712	30.47249	0.115987	16.05	50.55	2.380055	0.019991	0.11615
DJIAO	-15.34971	-88.91595	27.36166	-0.56231	16.97	46.1	1.312162	0.015638	0.10414
DJIAC	-15.34804	-92.87567	27.77387	-0.55418	18.51	45.56	1.212927	0.014767	0.09252
DJUAC	-5.201504	-45.16187	30.64218	-0.16979	16.88	47.63	3.722834	0.016193	0.11578
DJUAO	-5.068463	-40.01279	29.9737	-0.16912	16.51	48.16	3.819412	0.016272	0.11806
OIL	24.68917	-50.27416	128.1977	0.193607	11.44	57.1	2.890821	0.046991	0.18384

Table 7.19: The result of one step ahead prediction using non-stationary signals on DSMIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	8.6514328	-12.67	11.37	0.7617414	17.0	51.82	1.2859	0.0134828	0.101301
JPY/USD	-4.069113	-25.57624	12.794142	-0.318106	24.24	47.46	0.173247	0.003092	0.032686
USD/EUR	5.498935	-10.78767	10.101788	0.544569	11.97	51.40	6.279409	0.043175	0.168811
NASDAQO	-1.201741	-51.47693	30.738439	-0.039112	15.45	49.20	2.568989	0.021249	0.119179
NASDAQC	**7.886264**	-35.41700	30.495996	0.259041	17.87	52.25	1.429705	0.012009	0.083217
DJIAO	-3.891887	-64.33177	27.500631	-0.141980	14.87	51.67	2.043707	0.024356	0.127939
DJIAC	**4.078142**	-61.63275	27.869717	0.146780	13.07	51.10	2.655057	0.032324	0.160880
DJUAC	**-1.449810**	-48.2815	30.64368	-0.047426	12.72	50.92	8.587276	0.037352	0.179645
DJUAO	0.355431	-34.4738	29.98155	0.011822	12.92	51.48	8.51384	0.036272	0.174691
OIL	26.76248	-43.9429	128.00130	0.209142	11.26	57.32	2.951938	0.047984	0.183241

Table 7.20: The result of one step ahead prediction using non-stationary signals on R-MLP network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	0.470592	-22.72943	11.41861	0.041302	15.04	50.48	0.78955	0.008322	0.08495
JPY/USD	-1.08558	-20.85247	12.80468	-0.08487	19.38	46.06	0.3966	0.007071	0.0700
USD/EUR	**13.39764**	-8.21238	10.07715	1.329773	16.23	53.37	2.29262	0.01576	0.1141
NASDAQO	-5.15092	-49.74936	30.77493	-0.16743	20.86	47.42	0.65789	0.005455	0.0648
NASDAQC	-2.75681	46.53370	30.59627	-0.09014	21.62	49.8	0.5355	0.004508	0.05872
DJIAO	-2.12597	-36.2518	27.56866	-0.07724	13.56	50.6	2.39523	0.028616	0.15632
DJIAC	-5.46303	-48.1665	27.96083	-0.19588	13.77	49.58	2.23022	0.027219	0.15244
DJUAC	-7.75481	-52.47248	29.98587	-0.25877	17.44	47.64	2.78005	0.011869	0.10290
DJUAO	-8.15845	-55.99526	30.625883	-0.26628	17.81	46.55	2.51131	0.010947	0.09856
OIL	9.53221	-70.0853	130.31572	0.074606	10.88	52.29	3.24083	0.052305	0.191319

Table 7.21: The result of one step ahead prediction using non-stationary signals on R-SONIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	**13.222424**	-11.377406	11.370933	1.163817	16.4	53.43	1.479572	0.015595	0.111146
JPY/USD	0.109788	-15.434371	12.807689	0.00857	20.34	47.92	0.3144	0.005605	0.044706
USD/EUR	10.548035	-9.23998	10.084079	1.046945	11.51	52.33	6.581025	0.045239	0.196167
NASDAQO	**5.048035**	-30.243216	30.777562	0.164053	15.17	51.45	2.348151	0.019469	0.125064
NASDAQC	-1.929973	-40.470801	30.597727	-0.6314	17.07	50.55	1.59496	0.013426	0.101682
DJIAO	-1.448578	-31.948202	27.580758	-0.05255	16.88	50.25	1.202832	0.01437	0.109019
DJIAC	-3.431984	-14.144426	27.984762	-0.12285	15.65	49.62	1.611291	0.019665	0.125119
DJUAC	-7.377525	-51.271605	29.992723	-0.24607	14.3	47.28	5.613782	0.023967	0.147707
DJUAO	-0.69999	-44.666601	30.664225	-0.02314	13.91	51.48	6.051142	0.026377	0.154649
OIL	24.824701	-51.651684	126.72161	0.196018	11.09	65	3.123595	0.050429	0.190329

Table 7.22: The result of one step ahead prediction using non-stationary signals on R-DSMIA network

Time Series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	7.961360	-11.99492	11.386540	0.698326	19.04	50.75	0.866705	0.009135	0.077785
JPYUSD	**4.117505**	-12.73740	12.802085	0.321687	15.74	51.88	0.936123	0.016690	0.109100
USD/EUR	4.79326645	-9.637221	10.102186	0.474795	16.15	50.60	2.807111	0.019296	0.106179
NASDAQO	-4.559185	-49.91566	30.769507	-0.14822	14.63	51.15	3.715639	0.0255475	0.088114
NASDAQC	3.7920	-10.68990	10.104513	0.37547	14.39	49.92	4.0264272	0.0276844	0.1340
DJIAO	**0.363890**	-26.91238	27.582999	0.013194	20.66	51.48	0.547829	0.006545	0.067739
DJIAC	0.227811	-31.03056	27.998842	0.008123	16.05	50.43	1.409024	0.017197	0.116045
DJUAO	**0.049041**	-50.14233	30.014484	0.001550	16.42	48.20	3.441886	0.014694	0.115271
DJUAC	-0.760969	-40.29130	30.67966	-0.0248	13.42	50.85	6.848141	0.029851	0.162966
OIL	**38.573172**	-24.26994	125.75972	0.306841	14.39	60.86	1.594795	0.025739	0.132382

Table 7.23: Number of hidden nodes in SONIA and DSMIA for one step ahead non-stationary signals

Time Series	One step for Non-stationary signals		
	MLP	SONIA	DSMIA
USD/UKP	8	5	4
JPY/USD	7	6	4
USD/EUR	8	7	4
NASDAQO	6	7	3
NASDAQC	6	7	3
DJIAO	6	7	2
DJIAC	6	6	3
DJUAO	8	8	4
DJUAC	8	8	4
OIL	7	8	2

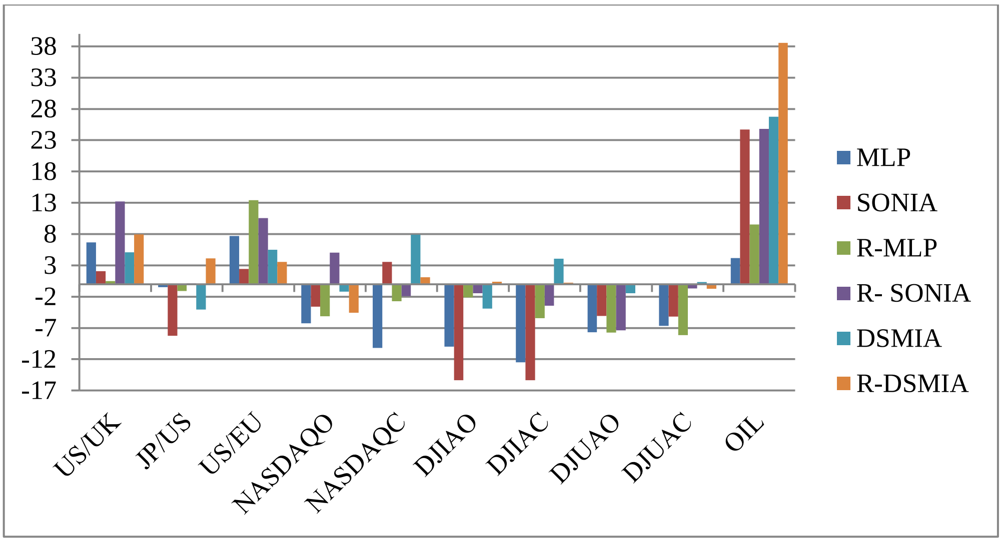


Figure 7.4: Annualised return on non-stationary signals for the prediction of one step ahead

Five step ahead prediction using non-stationary signals

The results were taken from the 30 simulations of testing the data set. Figure 7.5 summarises the result achieved from all neural networks. The result of the annualised returns from Table 7.24 to 7.29 show that the proposed DSMIA and R-DSMIA obtained the best profit return compared to all other network models in all-time series data. The R-DSMIA achieved the highest profit on USD/UKP, DJIAC, NASDAQC and OIL time series. Meanwhile, DSMIA obtained the best average of profit on the rest of the signals.

By looking at the other financial measures such as the maximum drawdown, volatility and sharp ratio the result showed that most of the best values were obtained by DSMIA and RDSMIA networks. The R-DSMIA achieved the highest values in five of 10 signals when evaluating the correct directional change; these are the USD/UKP, JPY/USD, DJIAC, NASDAQC and OIL signals. In the same measure, DSMIA made the highest CDC when forecasting five of the signals, namely the USD/EUR, NASDAQO, DJIAO, DJUAO and DJUAC signals. When evaluating the NMSE, MSE and MAE, it can be detected that DSMIA and R-DSMIA outperform other neural networks in some of the signals.

Table 7.24: The result of five step ahead prediction using non-stationary signals on MLP network

Time series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	-0.559585	-14.8441	11.42579	-0.049097	14.7	52.27	2.07635	0.02193	0.13984
JPY/USD	-3.59391	-20.2364	12.8158	-0.28073	16.34	47.95	0.7863	0.014011	0.09782
USD/EUR	1.117382	-11.8548	10.1309	0.11028	13.22	50.79	4.535523	0.031171	0.161433
NASDAQO	-2.40269	-37.1925	30.8166	-0.07803	16.54	48.44	1.7835	0.014823	0.109801
NASDAQC	1.65726	-34.1537	30.5985	0.054188	17	49.79	1.5557	0.013096	0.103413
DJIAO	-0.321544	-37.7277	27.5912	-0.011719	12.39	52.28	3.0879	0.036984	0.17838
DJIAC	-0.555403	-40.42709	27.5927	-0.020069	12.42	51.27	3.071542	0.036788	0.177886
DJUAO	-3.439273	-49.0705	29.9932	-0.11477	14.07	49.57	5.95375	0.025418	0.001689
DJUAC	-3.523605	-51.062871	30.70504	-0.114946	15.49	49.44	4.265581	0.018631	0.129215
OIL	0.01561	-88.71052	131.6793	-0.000611	10.6	49.42	3.471954	0.055847	0.199564

Table 7.25: The result of five step ahead prediction using non-stationary signals on SONIA network

Time series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	-1.48508	-17.776	11.41287	-0.13091	16.83	53.2	1.30044	0.013707	0.107113
JPY/USD	2.839703	-19.8414	12.7995	-0.22183	21.66	50.98	0.23189	0.004134	0.050603
USD/EUR	2.819784	-8.63958	10.11326	0.278768	13.32	51.89	5.38194	0.036996	0.149088
NASDAQO	-5.12982	-38.6083	30.77346	-0.16675	19.94	48.05	0.99797	0.008274	0.075397
NASDAQC	-1.00699	-32.3624	30.59728	-0.1253	17.1	48.41	1.63071	0.013727	0.099345
DJIAO	2.440176	-20.558	27.58149	0.088481	16.83	52.54	1.3179	0.015745	0.105434
DJIAC	1.880855	-22.1424	28.00402	0.067178	17.14	52.36	1.13388	0.013839	0.100822
DJUAO	2.857381	-35.7076	29.9849	0.095616	13.38	50.15	8.12858	0.034703	0.171216
DJUAC	1.735058	-34.5416	30.67759	0.056716	14.88	49.83	6.35314	0.027693	0.148254
OIL	-6.13704	-90.0799	130.3251	-0.04803	10.92	49.03	3.22148	0.051992	0.195043

Table 7.26: : The result of five step ahead prediction using non-stationary signals on DSMIA

Time series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	1.824697	-13.364766	11.417538	0.159847	15.30	51.52	1.813762	0.019118	0.128763
JPY/USD	**4.883264**	-14.536049	12.797590	0.381776	22.04	51.52	0.228185	0.004068	0.044344
USD/EUR	5.924942	-8.593920	10.109854	0.586446	11.57	52.27	6.985444	0.048019	0.182709
NASDAQO	**0.102613**	-37.366685	30.767597	0.003337	16.73	50.45	1.715399	0.014223	0.102406
NASDAQC	-0.645413	-32.550353	30.602293	-0.021085	18.99	49.45	1.104145	0.009295	0.076422
DJIAO	**2.444301**	-21.343170	27.579252	0.088642	19.07	52.73	0.937367	0.011199	0.086451
DJIAC	2.555257	-20.984023	28.001119	0.091279	12.83	51.60	2.789213	0.034041	0.167456
DJUAO	**4.642005**	-30.055884	29.992023	0.154933	13.48	51.13	7.056958	0.030128	0.162958
DJUAC	**4.005996**	-30.602317	30.662968	0.130994	13.76	50.33	6.438469	0.028065	0.157004
OIL	11.290324	-87.792361	130.448316	0.087339	10.78	53.00	3.357866	0.054193	0.194540

Table 7.27: The result of five ahead prediction using non-stationary signals on R-MLP network

Time series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	-3.09729	-16.877584	11.43124	-0.271036	15.85	54.27	1.58986	0.016787	0.122338
JPY/USD	-3.309134	-19.066487	12.807072	-0.258545	16.15	48.69	0.830921	0.014806	0.1023303
USD/EUR	3.733809	-8.871813	10.130352	0.368653	13.41	51.38	4.274036	0.029372	0.157885
NASDAQO	-4.589262	-39.692504	30.813032	-0.148991	20.5	48.22	0.699381	0.005812	0.066495
NASDAQC	-1.772365	-39.37541	30.635332	-0.057879	20.61	51.01	0.794322	0.006702	0.0681
DJIAO	-4.65767	-42.126048	27.587913	-0.168913	13.31	51.87	2.523355	0.030222	0.160633
DJIAC	-4.517743	-14.89301	28.014315	-0.161395	13.41	51.09	2.401935	0.029386	0.158422
DJUAO	-3.079075	-49.765621	30.050515	-0.102547	17.32	49.58	2.853534	0.012208	0.104095
DJUAC	-3.133686	-25.975362	30.717461	-0.10209	17.63	49.05	2.600441	0.011358	0.100258
OIL	0.113113	-82.687761	131.81674	0.000935	10.82	49.71	3.34083	0.053146	0.192832

Table 7.28: The result of five ahead prediction using non-stationary signals on R-SONIA network

Time series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	1.79951	-15.08815	11.42942	0.157756	14.11	51.4	2.392466	0.025269	0.14951
JPY/USD	2.034985	-14.67284	12.8153	0.159264	16.19	46.34	0.811097	0.014453	0.1007
USD/EUR	3.813749	-11.57839	10.13094	0.376467	11.37	51.93	6.722933	0.046658	0.19983
NASDAQO	-1.92625	-39.49456	30.82329	0.062527	16.13	47.64	1.877175	0.015601	0.11217
NASDAQC	-3.73281	-40.71476	30.62114	-0.1219	16.56	51.41	1.674038	0.014124	0.10597
DJIAO	-1.96245	-40.93965	27.59616	-0.07122	14.9	51.19	1.945163	0.023297	0.13512
DJIAC	-1.52512	-37.91129	28.01929	-0.05443	12.47	51.91	2.965914	0.036285	0.17679
DJUAO	0.054411	-48.78379	30.06198	-0.001792	17.15	49.74	3.008243	0.01287	0.10657
DJUAC	2.001813	-44.45037	30.72487	-0.065155	16.52	49.67	3.476785	0.015186	0.11555
OIL	-1.24209	-76.71338	132.1194	-0.00939	10.99	52.61	3.196299	0.051413	0.19085

Table 7.29: The result of five ahead prediction using non-stationary signals on R-DSMIA network

Time series	AR	MDD	AV	SR	SNR	CDC	NMSE	MSE	MAE
USD/UKP	**5.056818**	-13.46734	11.42431	0.443006	15.83	55.24	1.599081	0.016889	0.120819
JPY/USD	3.433556	-17.12925	12.81203	0.268642	14.08	52.03	1.311543	0.023371	0.130662
USD/EUR	**6.0058482**	-9.565491	10.09772	0.5950839	15.59	50.49	2.864027	0.0196921	0.115550
NASDAQO	-0.527541	-46.57787	30.82325	-0.017188	16.94	47.69	1.563239	0.012992	0.099038
NASDAQC	**3.584825**	-28.57183	30.62947	0.117098	17.21	53.32	1.445291	0.012194	0.094748
DJIAO	-0.104052	-36.79902	27.60053	-0.003778	10.62	50.10	9.546326	0.114335	0.26430
DJIAC	**9.405722**	-52.77260	27.778877	0.340198	12.32	53.83	3.266775	0.039771	0.182715
DJUAO	3.019954	-39.93085	30.04967	0.100646	14.59	49.90	5.905766	0.025267	0.14669
DJUAC	2.096306	-35.07649	30.72332	0.068255	14.99	49.47	4.817698	0.021042	0.136444
OIL	**14.81964**	-73.74407	127.788520	0.118494	11.43	55.42	2.902436	0.047179	0.17917

Table 7.30: Number of hidden nodes in SONIA and DSMIA for five step ahead non-stationary signals

Time Series	Five step for Non-stationary signals		
	MLP	SONIA	DSMIA
USD/UKP	8	4	4
JPY/USD	7	5	4-3
USD/EUR	7	7	4-2
NASDAQO	5	5	2
NASDAQC	7	8	2-3
DJIAO	5	5	2
DJIAC	7	5	3-2
DJUAO	7	6	4
DJUAC	8	7	4
OIL	5	6	2

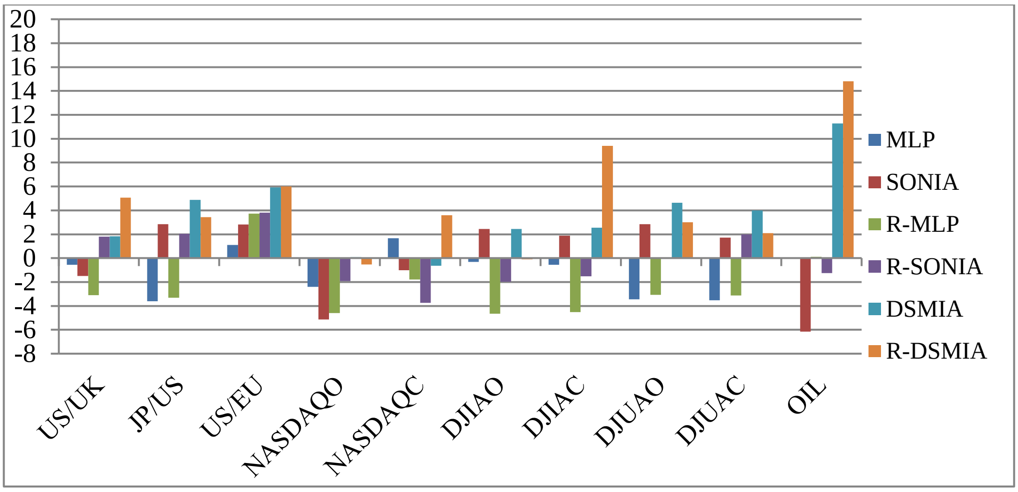


Figure 7.5: Annualised return on non-stationary signals for the prediction of five steps ahead

Table 7.31: The learning rate for DSMIA network

Time Series	Stationary		Non-stationary	
	One step	Five step	One step	Five step
USD/UKP	0.9/0.01	0.9/0.1	0.5/0.5	0.9/0.5
JPY/USD	0.9/0.1	0.6/0.09	0.1/0.5	0.1/0.5
USD/EUR	0.9/0.01	0.9/0.01	0.9/0.5	0.5/0.5
NASDAQO	0.9/0.1	0.7/0.1	0.1/0.01	0.1/0.1
NASDAQC	0.6/0.5	0.6/0.5	0.1/0.1	0.9/0.5
DJIAO	0.1/0.5	0.1/0.5	0.9/0.2	0.9/0.01
DJIAC	0.9/0.1	0.9/0.1	0.1/0.2	0.1/0.2
DJUAO	0.9/0.01	0.9/0.1	0.9/0.5	0.1/0.1
DJUAC	0.9/0.01	0.1/0.4	0.1/0.1	0.1/0.1
OIL	0.1/0.001	0.1/0.01	0.1/0.1	0.1/0.1

7.2.7 Financial Discussion

This section summarises the simulation result that been showed above. The motivation of this section is to discuss some issues, which have arisen due to the comparison of the results achieved by the networks discussed previously.

7.2.7.1 Comparison between the Proposed DSMIA network and the SONIA in Stationary Signals

The simulation results using stationary prediction indicated that the proposed network generates slight improvements using the various evaluation measures. The neural networks have learnt the signals very well; this is related to the fact that noise points of the non-stationary signals have been smoothed by using the Relative Difference in Percentage of Price. The DSMIA generates a good result in comparison to the SONIA neural network. For five step and one step ahead prediction, the result of the Annualised Return (AR) as shown in Figure 7.6 demonstrated that the proposed model attained high profit values in some data series compared with the SONIA neural network.

By looking at the Maximum Drawdown (MDD) and Volatility (AV) for one and five step ahead predictions for stationary signals, simulation results as indicated in Tables 7.3 to 7.15 clarify the advantages of the proposed DSMIA network.

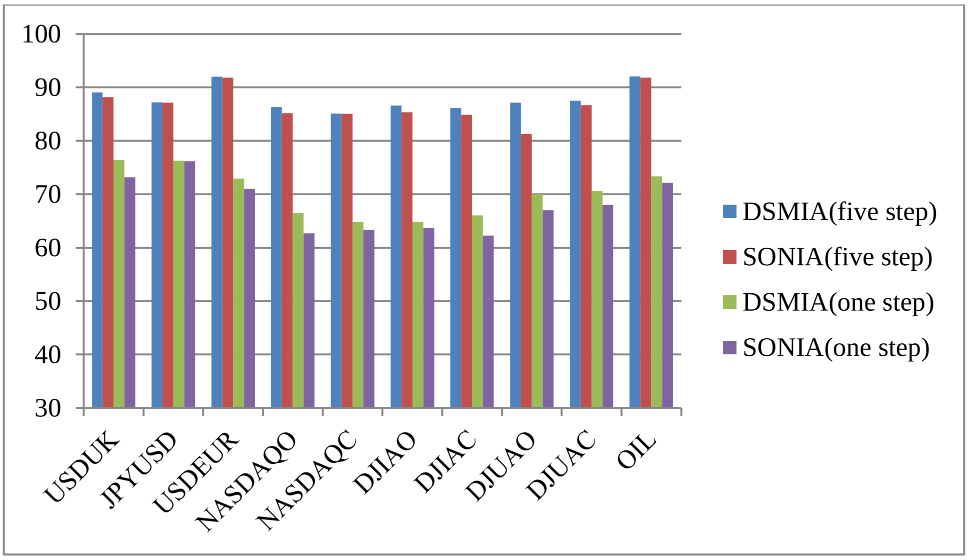


Figure 7.6: Annualised return result of the proposed DSMIA network and the SONIA in stationary signals

7.2.7.2 Comparison between the Proposed DSMIA network and the SONIA in Non-Stationary Signals

The analysis of the non-stationary signal, in terms of one step ahead prediction results as illustrated in Tables 7.3 to 7.9 and Figures 7.4 and 7.5, shows that the different evaluation measures vary for the various data series. In some cases, the proposed DSMIA managed to generate better profit values than the rest of the networks. On the other hand, for five step ahead prediction, the proposed DSMIA network has been shown to generate more profits than most of the benchmarked neural networks. The results of the Annualised Return are illustrated in Figure 7.7 and demonstrate that the proposed model attained high profit values in all 10 series data when compared with the SONIA network. This really confirmed the success of the proposed network to predict the financial time series in order to achieve high profit value.

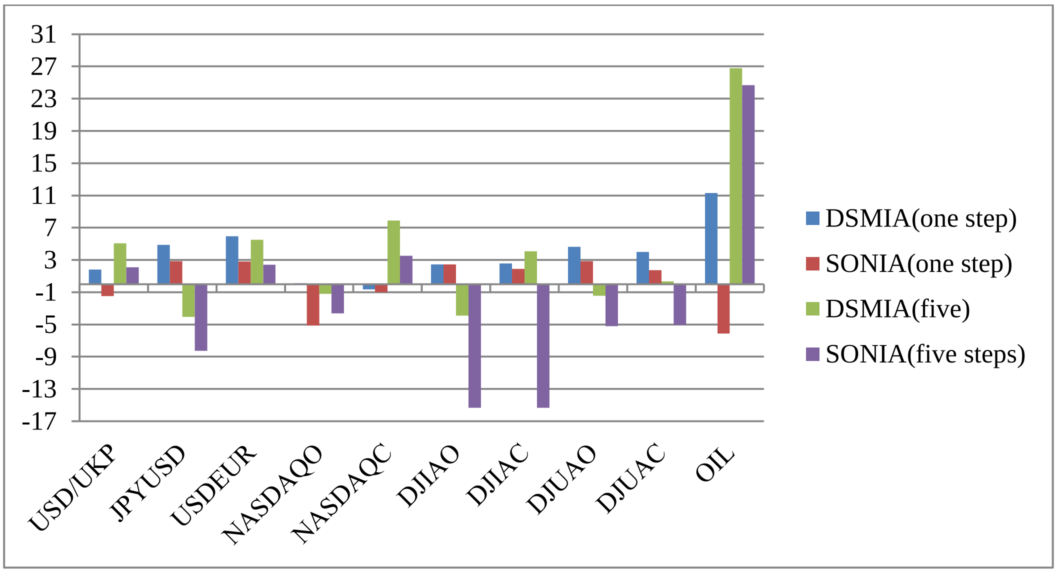


Figure 7.7: Annualised return result of the proposed DSMIA network and the SONIA in non-stationary signals

7.2.7.3 The Conclusion for Forecasting Results of the Non-stationary Signals for all Networks utilised in this Research

From the research experiments, the simulation results showed that the prediction of non-stationary financial signals is very difficult since the signals are highly noisy and volatile. These instable behaviours lead to signals changing and falling sharply at some point during the network training. During the training, the neural networks are attempting to learn the price values of the non-stationary signals; however, their responses are not sufficient. This is related to the fact that the behaviour of price values is not stable. Although the prediction for the non-stationary signals usually presents inconsistent results, the extensive experiments of this research confirmed that the proposed DSMIA and R-DSMIA achieved the best profit values compared to other networks. This has been clearly represented in Tables 7.17 to 7.29.

7.2.7.4 The Benefits of using DSMIA and R-DSMIA for Forecasting Stationary and Non-Stationary Signals

As can be observed from Tables 7.3 to 7.29, in most cases the proposed DSMIA neural network generates profit when used for the prediction of one step and five step ahead predictions for both stationary and non-stationary signals. From the various experiments, the simulation results indicated that the proposed DSMIA has shown improvements in the prediction of financial time series when benchmarked with feed-forward neural networks. The proposed network shows highly nonlinear dynamical behaviour provided by the recurrent feedback, consequently allowing a better input-output mapping and a better prediction. These recurrent connections make the neural network based on the external inputs as well as inter history of the system inputs. This can confirm that these connections stored the information about the previous values of the signal and hence better prediction was attained in comparison to feed-forward neural network architectures. Furthermore, the proposed network significantly helps to improve the result of profit return. In addition, the learning process in the DSMIA network is centred on the local properties of the signal; the self-organising hidden layer is specifically programmed to adapt to these properties. As such, DSMIA architecture networks have a more detailed mapping of the underlying data structure, enabling them to respond better to the data changes or structural shifts common in non-stationary signals. It can be concluded that the experiments' results indicate that the proposed network generates slight improvements in some of the financial time series data. Furthermore, using the weight decay method in the DSMIA network has improved the generalisation ability of the network. The prediction results for one step and five step ahead prove that the application of R-DSMIA models is considered as a promising tool for nonlinear financial time series prediction, as illustrated in Figures 7.8 and 7.9.

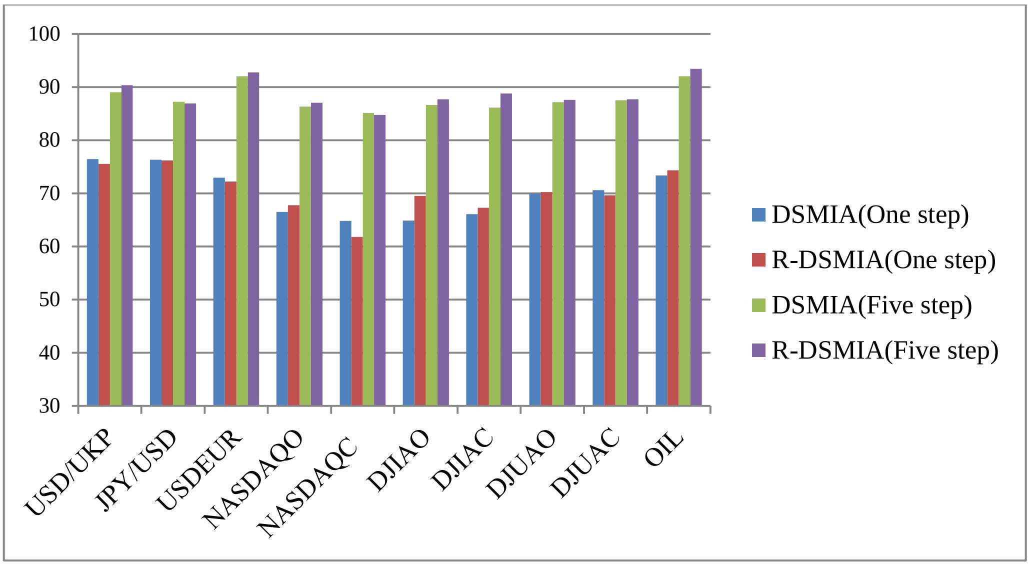


Figure 7.8: Annualised return result of the proposed DSMIA network and the R-DSMIA in stationary signals

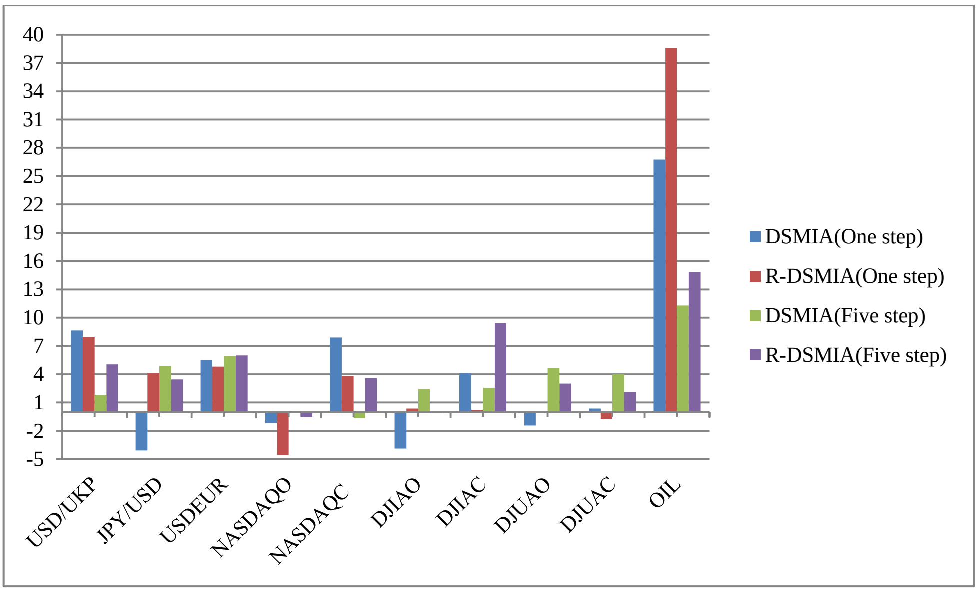


Figure 7.9: Annualised return result of the proposed DSMIA network and the R-DSMIA in non-stationary signals

In accordance with the number of hidden neurons that have been used in the neural networks, it is vital to judge a network's parsimony and simplicity. From Tables 7.9, 7.16, 7.23 and 7.30, it can be observed that the proposed network requires fewer hidden units when compared with the SONIA network for the prediction of one and five step ahead predictions. This will help to reduce the computational complexity, and will improve the network generalisation ability, as can be observed from Table 7.32, which means that the recurrent link in the network has provided and enhanced the ability of the network to learn the pattern of signals with fewer hidden units compared with other networks that have been used in these simulations. However, this enhancement was reached at the expenses of longer processing and training time, as shown in Table 7.33.

Table 7.32: The average result for MSE

	One ahead prediction using Stationary signals			Five ahead prediction using Stationary signals			One ahead prediction using Non-stationary signals			Five ahead prediction using Non-stationary signals		
	MSE training	MSE Testing	MSE validation	MSE Training	MSE Testing	MSE Validation	MSE Training	MSE Testing	MSE Validation	MSE Training	MSE Testing	MSE Validation
USD/UKP	0.0037	0.003655	0.0029	0.0010	0.002104	7.7454e-04	1.7153e-04	0.013482	0.0041	4.0859e-04	0.019118	0.0049
JPY/USD	0.0016	0.002402	0.0014	0.0014	0.002968	0.0016	4.7066e-04	0.003092	0.0044	9.1795e-04	0.004068	0.0016
USD/EUR	0.0031	0.003153	0.0019	0.0017	0.002941	8.6123e-04	1.2812e-04	0.043175	4.7616e-04	0.00049	0.048019	7.5966e-04
NASDAQO	0.0011	0.002228	4.9229e-04	7.9912e-04	0.001331	3.1220e-04	9.4954e-04	0.021249	0.0126	3.2749e-04	0.014223	0.0011
NASDAQC	0.0022	0.0032860	8.3629e-04	0.0011	0.001747	8.1516e-04	4.1464e-04	0.012009	0.0053	3.0324e-04	0.009295	0.0018
DJIAO	0.00101	0.002566	4.0168e-04	6.0977e-04	0.001560	2.1980e-04	0.0011	0.024356	0.0116	4.6189e-04	0.011199	0.0106
DJIAC	0.0010	0.002530	3.9837e-04	7.8794e-04	0.001298	2.6511e-04	9.6443e-05	0.032324	0.0066	5.6379e-04	0.034041	0.0057
DJUAO	0.0011	0.001641	5.9834e-04	8.5756e-04	0.000780	3.2458e-04	0.0018	0.037352	0.0040	0.0011	0.030128	0.0079
DJUAC	0.0011	0.001658	6.3052e-04	6.1468e-04	0.000738	4.4068e-04	4.8806e-04	0.036272	0.0169	4.7149e-04	0.028065	0.0113
OIL	0.00264	0.001928	0.0023	0.0027	0.001012	0.0032	0.0069	0.047984	0.0037	4.9769e-04	0.054193	7.5966e-04

However, the proposed network has not shown any improved results on some financial data. Nevertheless, it must be taken into account that this might be related to the raw data, since the data are affected by several factors such as the threat of war, good or bad economic climates, announcements of company earnings, and the advertisement of economic statistics.

Table 7.33: A comparison between the times required to complete the simulation using the SONIA and the proposed DSMIA network

Time series	SONIA	DSMIA
USD/UKP	25.2207	64.0190
JPY/USD	33.2589	79.6417
USD/EUR	30.1757	72.6679
NASDAQO	28.9414	42.1859
NASDAQC	30.7793	71.9675
DJIAO	29.6505	36.7067
DJIAC	29.3825	70.2133
DJUAO	34.0757	81.5248
DJUAC	32.6150	70.8630
OIL	4.9141	10.7170

7.3 Medical Time Series

In this section, the proposed network will be evaluated and compared with several machine learning classifiers by using an open data set, involving 300 records (38 preterm and 262 term) (Fele-Zorz et al. 2008). EHG signals have been pre-processed, and features have been extracted from signals in order to classify them into term and preterm subjects.

7.3.1 Uterine EMF Signal Sources

The data used in this research were recorded at the Department of Obstetrics and Gynaecology, Medical Centre, Ljubljana between 1997 and 2006 (Physionet.org 2010). In the TPEHG database, there are 300 records of patients, as shown in Table 7.34. These records are openly available, via the TPEHG data set, on the Physionet website. The signals in this study had already been collected by Fele- Zorz et al. (2008). Each record was collected by regular examinations at the 22nd week of gestation or around the 32nd week of gestation. The signal in each record is 30 minutes long, has a sampling frequency (f_s) of 20 Hz, and has a 16-bit resolution over a range of ±2.5 millivolts.

Prior to sampling, the signals were sent through an analogue three-pole Butterworth filter, in the range of 1 to 5 Hz. Each record is obtained from three channels, Channel 1, Channel 2 and Channel 3. The Channel 1 signal was measured between E2 and E1, Channel 2 was recorded between E2 and E3 and the Channel 3 signal was recorded between E4 and E3.

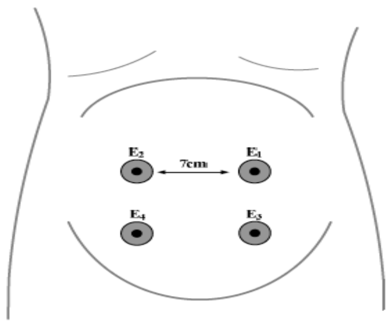


Figure 7.10: Placement of electrodes on the mother's abdomen. (Baghamoradi et al. 2011)

The recording time shows the gestational age. Each recording was classified as a full-term or preterm delivery, after birth. Figures 7.11 and 7.12 show two examples of EHG signals from different records. The preterm birth used in this signal is referred to as birth following a fully completed 37 weeks. The recordings were categorised into four types as follows:

1. Early – Term: Recordings made early, signed as a term delivery
2. Early – Preterm: Recordings made early, signed as a preterm delivery
3. Late – Term: Recordings made late, signed as a term delivery
4. Late – Preterm: Recordings made late, signed as a preterm delivery

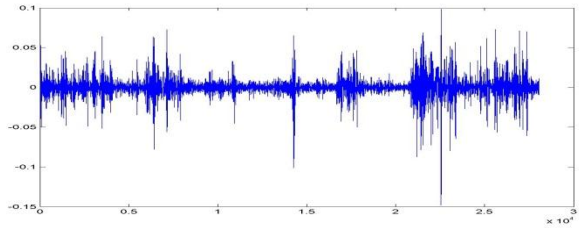


Figure 7.11: Row Data Plot for the uterine EHG signals – Preterm subject

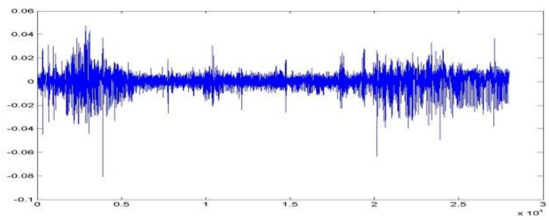


Figure 7.12: Row Data Plot for the uterine EHG signals – Term subject

Table 7.34: The various recording for the Term-Preterm Electrohysterogram data set

	Number of recordings	Time of recordings	Deliveries
Term	143	22.7	39.7
262 recodes	119	30.8	39.6
Prematurely	19	23.0	34.2
38 records	19	30.2	34.7

7.3.2 The Data Pre-processing

Since the EHG signals are recording from the muscle, their quality is affected by different factors, such as the movement of the body, or movement of the foetus – breathing, and heartbeat. However, in order to reduce these effects, a step should be taken before analysing or classifying the EHG signals. This step is pre-processing and it consists of a number of methods which are: filtering, de-noising, and automatic recognition of bursts (Fergus et al. 2013). Recently, a number of studies have focused on filtering the EHG raw signals to allow frequencies between 0.05 Hz to 16 Hz (Leman et al. 1999; Verdenik , Pajntar 2001; Maner et al. 2003; Marque et al. 2007; Maner 2007). Some researchers have filtered EHG signals as high as 50 Hz (Buhimschi et al. 1997). However, it is not recommended to use uterine EHG with such a wide range of frequencies; this is because more interference can affect the signal (Cheung 2012). It was documented that the uterine EMG signals' content ranges from 0 to <5 Hz (Devedeuxet al. 1993).

In the TPEHG data set, the signals were filtered using three different 4-pole digital Butterworth filters. Signals have been filtered twice in both backwards and forwards directions in order to overcome the phase-shifting that can occur when utilising these filters. The band-pass filters were as follows:

- Filter 1: 0.08-4Hz
- Filter 2: 0.3-4Hz
- Filter 3: 0.3-3Hz

Fele-Zorz et al. (2008) used 0.08-4 Hz in order to compare their outcomes to earlier studies (Verdenik, Pajntar 2001). In addition, they increased the lower limit on frequency cut-off and selected 0.3-4 Hz. They used this filter in order to test the elimination of noise in lower frequencies, due to skin stretching and breathing. The third filter was selected to remove higher frequencies.

7.3.3 The Data Features

The next step for analysing EHG signals is extracting features. A number of techniques have been applied to capture the most important information of biological signals and to extract the features of interest for the classification task. These techniques are based on spectral and temporal analysis measures (Barrios 2010). Characterising biological signals such as EHG can be very challenging. This is because it requires a suitable method to consider when choosing a feature representation for EHG classification. First of all, EHG signals naturally involve information that is both spatial and temporal. The way that biological signals such as EHG are recorded is through the process of placing electrodes on the body surface. Therefore, any feature testing that neglects the patterns either across the electrode or through time may abandon some important pattern in the signal.

Furthermore, EHG has relatively high temporal resolution, normally between 128 and 1002 samples per second, producing a numerous amount of data to process. The importance of exploring the biological time series has contributed to the proposed number of featured methods that can be used to discover important properties of the biological time series signals. They are categorised into two types of methods: linear and nonlinear techniques. A number of researchers have explored the characterisation ability of linear and nonlinear features on biological time series (Lehnertz et al. 2001; G. Fele-Zorz et al. 2008; Balli & Palaniappan 2010)

7.3.3.1 Linear Features

This section will discuss linear signal processing techniques provided with the TPEHG data set (Physionet.org 2010). This feature has been applied to uterine EHG signals in order to distinguish between preterm and term patients. Three linear signal processing techniques were used: Root Mean Square (RMS) (Rosa et al. 2007; Pajntar et al. 1998), Median Frequency (MF), and Peak Frequency (PF) (Fele-Zorz et al. 2008). The last two features were determined from power spectra densities. The RMS value of the EHG signal will change before the labour starts (Verdenik I, Pajntar M 2001). Hence, it can be used to detect true labour. Leskošek et al. (2002) applied RMS of EHG signals to find the similarity between human and mammal contraction. Fele-Zorz et al. (2008) found that Median Frequency (MF) parameters are efficient to identify preterm labour. In addition, Verdenik and Pajntar (2001) found that an increase in gestational age leads to a decrease in median frequency. However, it has been shown that the values of RMS and MF of uterine EHG signal are affected by the placement of electrodes on the abdominal exterior (Kavsek G, Pajntar M 1999). Moslem et al. (2011) indicated that the median frequency feature can identify contractions better than the other frequency parameters such as mean frequency and peak frequency. It has been shown that MF has a sensitivity of 0.83 and a specificity of 0.69, which are higher than the other frequency-related parameters. Moreover, MF has shown the highest classification performance. Studies report that Mean frequency has the most potential to distinguish between preterm and term subjects. Nevertheless, root mean squares and peak frequencies have had conflicting results. However, some research has indicated that these linear features are suitable for discriminating between preterm and term subjects (Fergus et al. 2013). Despite the fact that linear analysis is very simple to apply and interpret, it is unable to discover information on nonlinear relation of biological signals (Diab et al. 2012). Therefore, using nonlinear analysis techniques would be reasonable in obtaining an improved characterisation of EHG signals.

7.3.3.2 Nonlinear features

Recently, there have been great efforts to define nonlinear parameters to characterise the dynamic behaviour of biological signals. These studies (Hassan et al. 2009; Mars and Lopes da Silva 1983; Akay 2000), have indicated that the relation between biomedical signals is nonlinear. Consequently, the application of nonlinear processing signals is highly recommended (Fele-Zorz et al. 2008). In the last decade, a number of nonlinear methods have been studied for analysis of EEG biological signals collected from physiologic and pathologic conditions (Lehnertz et al. 2001) or EMG (Diab et al. 2012; Cunha & de Oliveira 2000). Many investigators concluded that nonlinear features are useful for classification of biological signals. Fele-Zorz et al. (2008) have proved that nonlinear features had more satisfactory results than linear features. One nonlinear measure, sample entropy, was shown to offer good information about uterine EHG signals (Hassan et al. 2010). According to experiments (Fele-Zorz et al. 2008; Baghamoradi et al. 2011), the sample entropy features are well suited to distinguishing between term and preterm records. Mars, Lopes and Silva (1983) confirmed that, when labour is in progress, the sample entropy values decrease. Hence, they can be used to evaluate the progress of the labour (Vrhovec 2009). In addition, it has been shown that the values of sample entropy for full-term labour records are higher than those for preterm labour (Fele-Zorz et al. 2008). In this study, four types of features are used, three linear and one nonlinear. The selection of these features is based on the previous study, which was conducted by Fele-Zorz et al. (2008).

7.3.4 Experiments

This section will involve two experiments that are proposed to classify EHG signals. This study is only interested in whether the records are preterm or term. The aim of this section is to evaluate the effectiveness, and efficiency, of using the SONIA network and the proposed DSIA network for medical classification. Novel recurrent neural network architecture based on the immune algorithm and the self-organised neural network is proposed for the classification of Electrohysterography signals into term and preterm. Figure 7.13 shows the proposed schematic for classifying the EHG signals.

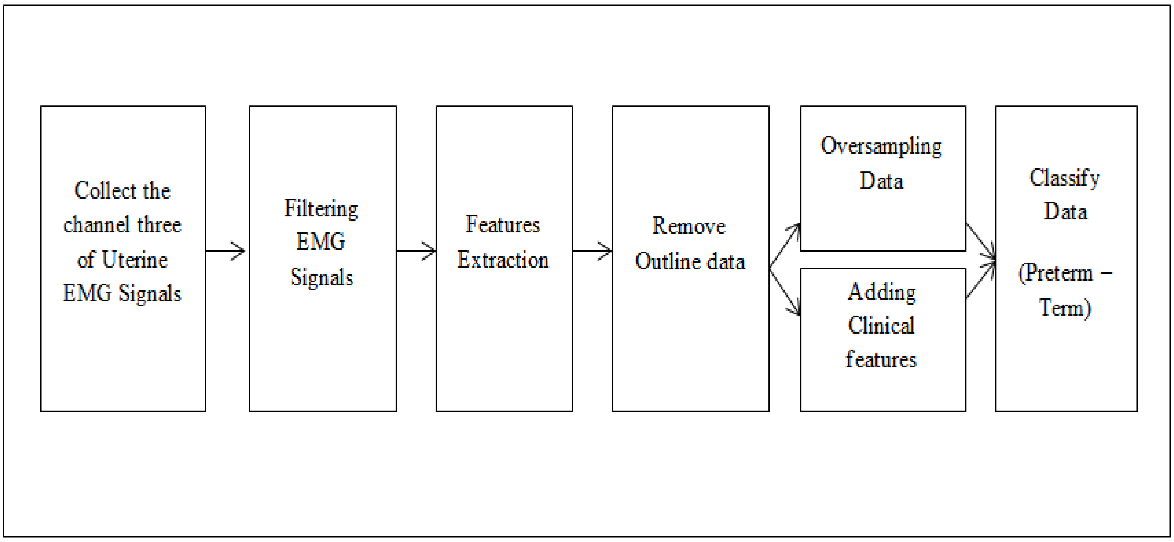


Figure 7.13: Proposed method for classifying Preterm and Term classes using Uterine EHG signals

The proposed network (DSIA) was benchmarked with three feed-forward neural networking models (MLP, SONIA and Fuzzy SONIA) and Elman recurrent neural network. Following an analysis of the literature, simple yet powerful algorithms, which give good results, will also be considered in our experiments. These are K-Nearest Neighbour Classifier (KNN), Decision tree classifier (treec) and Support Vector Classifier (svc).

Initially, for simplicity, the 0.3-3 Hz filter on the third channel was selected in this experiment, since it had been considered by Fele-Zorz et al. (2008) as one of the filters that provided more probability of discrimination between the classes of premature or term deliveries.

Experimental observations have pointed out that the classification of biological signals can be improved by gathering linear and nonlinear features. Balli and Palaniappan (2010) showed that such a combination of linear and nonlinear features has slightly increased the classification accuracy with an average improvement of up to 20% to detect the ECG signal for heart diseases. In this thesis, the combined features of linear and nonlinear methods are considered. Therefore, the data set sets were generated by extracting three linear features – root mean squares, peak frequency and median frequency – and one nonlinear feature – sample entropy.

7.3.4.1 Removing Outliers

By initially reviewing the features in the TPEHG data set, using quantile-quantile plots (Q-Q Plots), for each of the features, normal distributions were not evident, as illustrated in Figure 7.14. From the plots, it can be shown that there are likely outliers in the data. This is particularly happening with root mean squares, median frequency and peak frequency, as there are significant departures from the reference line for several observations. These outliers could be shown as a result of the movement of the mother or the baby, or the interference of other equipment around the hospital room where the signals were captured.

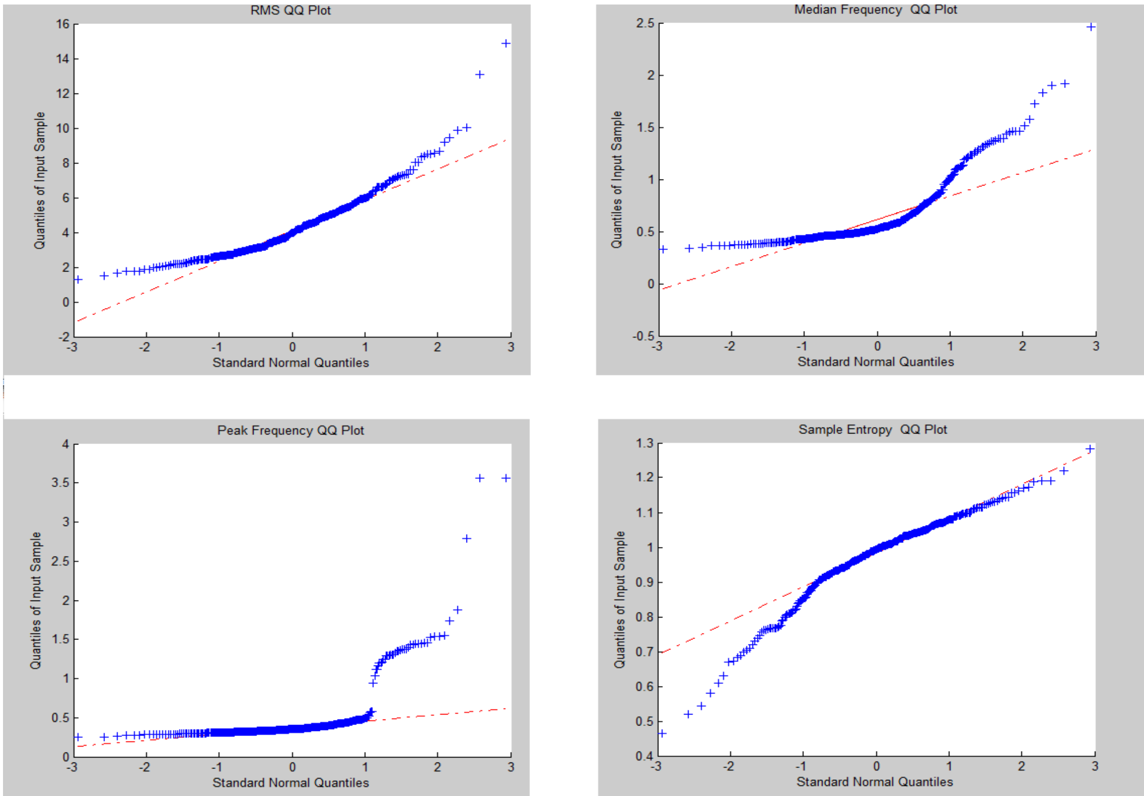


Figure 7.14: Outliers in Uterine EHG data

The outlined data have been removed based on the upper and lower limits for each feature. This transpired across all records that reside outside the body of records in the data set. For example, in the root mean squares feature, most of the records reside within 1.5 and 7. All records with root mean square values higher than 7 and lower than 1.5 have been removed. This process was carried out for median frequency (values bigger than 0.7 and less than 0.3 have been removed), peak frequency (values bigger than 0.5 and less than 0.25 have been removed) and sample entropy (values bigger than 1.0 and less than 0.5 have been removed). This makes sure the removal of values is furthest from the sample mean, as can be seen from

Figure 7.15. So, the data have been reduced to 215 samples (32 preterm data, and 183 term data).

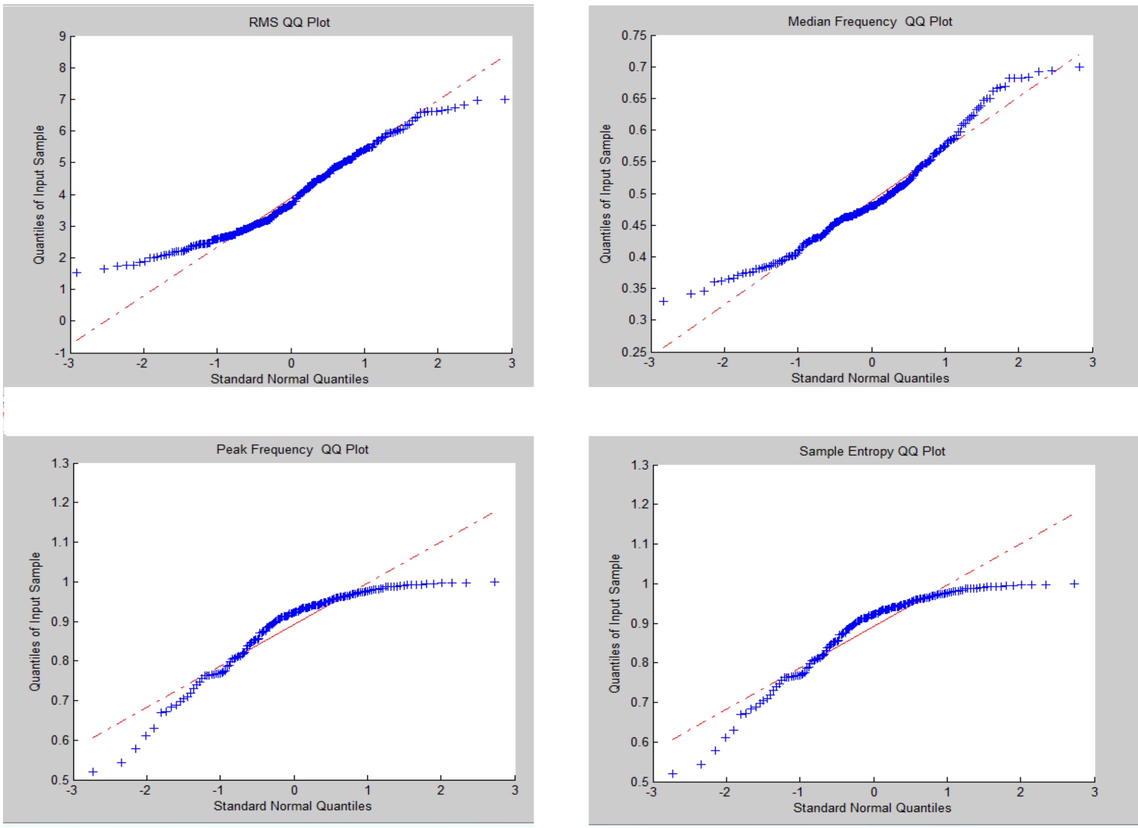


Figure 7.15: Uterine EHG data with Outliers removed

7.3.4.2 Performance Measures

The performances of each classifier have been evaluated by computing the sensitivity, specificity, positive, negative predicted values and area under the ROC that each neural network produced to separate the groups of uterine EHG signals. The formulas used to measure sensitivity, specificity and accuracy are defined as follows:

$$Accuracy = ((TP + TN)/(TP + TN + FP + FN)) \times 100 \quad (7.24)$$

$$Sensitivity = TP/(TP + FN) \quad (7.25)$$

$$Specificity = TN/(FP + TN) \quad (7.26)$$

$$Negative = TN/(FN + TN) \quad (7.27)$$

$$Positive = TP/(TP + FP) \quad (7.28)$$

Where TP is true positive and it refers to correctly detect preterm subjects. TN is true negative, which refers to correctly detected term subjects. FP (false positive) and FN (false negatives) refer to the number of uncorrected detections of term and preterm subjects respectively. These have been chosen since they are appropriate evaluation measures for classifiers, which produce binary output (Lasko et al. 2005).

ROC analysis

The Receiver Operating Characteristic (ROC) curve shows the cut-off values for the true negatives and false positives. ROC analysis was used in a significant number of studies. They have been criticised for the selection of a specific threshold value, which leads to subjectivity, as well as their inability to demonstrate whether the model has a good fit to the data (Lobo et al. 2008). The ROC curve is a two-dimensional measure of classification performance. ROC is mostly used in biomedical research to perform a diagnostic test through graphical analysis. The diagnostic tests evaluate the curve in the plot of sensitivity against specificity from the application of thresholds to the system output (Witten & Frank, 1999; Zhou & Harezlak, 2002). In other words, the ROC curve can be calculated through positive fraction vs. the false positive fraction, and the x-axis represents false positive (1-specificity) and true positive (sensitivity) is represented in the y-axis. If the testing of the classification using the ROC performs well, the upper right-hand corner of the graph rapidly rises towards 1.0 area in the graph, but if it is lower than 0.5 it means that the result of the test is incorrect.

The Area Under the Curve (AUC) has also been used to evaluate the performance of all classifiers. The AUC shows the probability of correctly identifying the positive subject which is preterm higher than a randomly chosen negative subject which is term subjects (Hanley & Mcneil 1982).

7.3.4.3 Fuzzy-similarity-based self-organized network inspired by immune algorithm (F-SONIA)

This methods introduced by Widyanto et al. (2006). It is an extension of SONIA network (Widyanto et al. 2005). They apply fuzzy similarity to SONIA in order to improve the classification ability of SONIA network. They have been proposed this method as an artificial odor discrimination system to classify three mixture-fragrances. Form their experiments, they conclude that using the fuzzy similarity as input to the SONIA classifier can improve the accuracy of the classification and achieved (100 %, 98.33 %, 94.17 %) classification accuracy which is higher than the basic SONIA network which achieved on the three mixture fragrances (90.83 %,98.17% , 90.83 %) .

The fuzzy similarity measure applied in fuzzy SONIA is applied to find the relationship between the inputs and weights on the self-organised map units. This is computed by finding the maximum value of the intersection region between the fuzzy set of the input vectors and the weights of the hidden units. The average value of the μ_{ji} values of the intersection are the output of the fuzzy- hidden units.

7.3.4.4 Simulation Result and Analysis

In this section, the simulation results using DSIA with other benchmarks networks are presented. It will involve two experiments.

Simulation result of oversampling data

Since there is a lack of preterm data compared to term data in the TPEHG data set, the neural network will not have enough cases of preterm data from which to learn (32 preterm samples), in contrast to term data (183 term samples). Therefore, the oversampling method has been used to generate another 87 items of preterm data. The generation has been done randomly between minimum and maximum value of each feature in the 32 preterm data, resulting in 119 preterm samples. The newly generated data are mixed with the original data set.

Results for 0.3-3 Hz TPEHG Filter on Channel 3 with RMS, MF, PF, and Sample Entropy with Oversampling

This evaluation uses the 0.3-3 Hz filtered signals on Channel 3 with seven classifiers. The performance for each classifier is evaluated using Sensitivity, Specificity, Negative and Positive predicted values with 30 simulations. The data have been split up as follows: 40% of the data has been selected randomly as training data, with 20% for validation and 40% as testing data. The experiments have been run 30 times to generate the average results. The learning rate of the DSIA network parameter, initially set to 0.1, decreased over time, but did not decrease below 0.01. Table 6.39 showed all the parameters that were selected to build four neural networks.

Classifier Performance

Evaluation of the neural networks' classification of oversampling data is summarised in Tables 7.35 and 7.36. Table 7.35 shows the average performance for all the classifiers used in this experiment.

First of all, the performance of the proposed DSIA will be evaluated. The results demonstrated that the proposed DSIA network showed the best accuracy with 73% compared to all classifiers. Furthermore, the results in Table 7.36 show the best values obtained by the DSIA network in terms of mean error, standard deviation compared to the other networks. The simulation results indicated that the sensitivity for preterm records obtained with the DSIA network is slightly better than other classifiers; the sensitivity is also higher than the SONIA. This means that the DSIA network has the ability to predict the true positive value of the preterm class; it can also predict the true negative value, which is the term class. However, the DSIA shows the highest values for Sensitivity and True Positives with slightly lower values for Specificity and True Negatives.

Secondly, the novel application of SONIA will be evaluated. The sensitivity (preterm) obtained by the SONIA network is slightly better than that obtained by the MLP; the specificity is also higher than the MLP. This means that the SONIA network has the ability to predict the true positive value of the preterm class; it can also predict the true negative value, which is the term class. The results in Table 7.36 show the best values obtained by the SONIA network in terms of mean error, standard deviation and classification accuracy compared to the MLP network.

Table 7.35: Classifier Performance Results for the 0.3-3Hz Filter

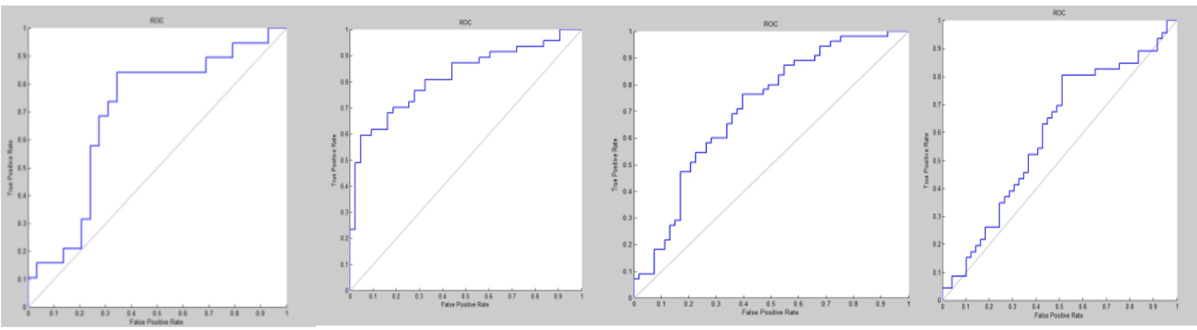
	Sensitivity	Specificity	True Negative	True Positive	AUC
MLP	0.6481	0.5691	0.6261	0.6205	0.5089
SONIA	0.6316	0.6920	0.7959	0.6073	0.78
KNNC	0.6944	0.6388	0.6764	0.6578	0.45
TREEC	0.5833	0.6111	0.5945	0.6000	0.58
SVC	0.666	0.6388	0.6571	0.6486	0.54
DSIA	0.7660	0.6809	0.7269	0.7241	0.8168
Fuzzy_SONIA	0.8642	0.4566	0.7647	0.6227	0.70
Elman	0.4928	0.5816	0.564	0.543	0.53

Table 7.36: Mean Error, Standard Deviation and Classifier Accuracy

	Mean error	Std Error	Accuracy
MLP	0.3073	0.0369	61.60%
SONIA	0.2244	0.0031	70.3472 %
KNNC	0.4000	0.0424	66%
TREEC	0.3907	0.0565	59%
SVC	0.4292	0.0476	65%
DSIA	0.1807	9.3409e-05	73.06%
Fuzzy_SONIA	0.2287	0.0015	66.41 %
Elman	0.2584	0.0187	53.8%

ROC Analysis

The receiver operator characteristic (ROC) curve in Figures 7.16 illustrates the trade-off between a classifier's true positives rate (sensitivity) versus its false positives rate (1-specificity). Figure 7.16 shows that the DSIA performance is better than the SONIA, which indicates that the DSIA curve is close to the upper left-hand corner and its area is greater than the SONIA curve; this confirms that the DSIA has greater power for classification than other neural networks. Figure 7.16 shows that the Elman network performance is lower than other networks. The simulation results indicated that the proposed DSIA network showed the best AUC with 81%; this is slightly better than the SONIA, which achieved 71%. From Table 7.35, it can be seen that the AUC of the DSIA network is greater than the other classifiers.



a) ROC for SONIA **b) ROC for DSIA** **c) ROC for fuzzy_SONIA** **d) ROC for Elman**

Figure 7.16: ROC curve for two best performing classifiers

Simulation result for the extended features

Each EHG signal record contains clinical information relating to the patients; this consists of: the pregnancy duration at the time of recording, maternal age, number of previous deliveries (parity), previous abortions, weight at the time of recording, hypertension, diabetes, bleeding first trimester, bleeding second trimester, funnelling, smoker. These 11 items of clinical information are added to the four features of TPEHG. However, some information was missing for some of the women, which led to unknown features on some recorders. Hence, the records with unknown information have been removed, thereby reducing the number of samples in the data set. These new data with extra features contain 19 preterm data samples and 108 term data samples. As before, the 19 preterm records are oversampled using a min/max technique to generate the 104 preterm data items. This technique allows a new data set to be constructed that provides an even balance between term and preterm records.

Results for 0.3-3 Hz TPEHG Filter on Channel 3 with RMS, MF, PF, and Sample Entropy with Clinical Data and Over-Sampling

These 11 items of clinical information are added to the original TPEHG feature set (RMS, MF, PF and Sample Entropy). The data set consists of 104 preterm data samples and 108 term data samples. The performance for each classifier is evaluated using Sensitivity, Specificity, Negative and Positive predicted values with an average of 30 simulations. The data have been split up as follows: 40% of the data has been selected randomly as training data, with 20% for validation and 40% as testing data.

Classifier Performance

The evaluated results for the proposed DSIA network are illustrated in Tables 7.37 and 7.38. The experiment results confirm that extending the number of features to 15 has significantly improved the classifiers' performance. These features have provided classification methods with enough information for each record to allow them to obtain better values in all of the evaluation functions.

The simulation results from Table 7.37 demonstrated that the SONIA model scored the highest in all evaluation parameters, followed closely by the proposed DSIA model. However, DSIA achieved the higher value of the AUC compared to other classifiers. In term of other neural networks, Elman network has achieved good result compared to MLP and Fuzzy_SONIA networks. The results in Table 7.38 shows that the best values obtained in terms of mean error, standard deviation and classification accuracy are by the SONIA network followed by the DSIA network.

Table 7.37: Classifier Performance Results for the 0.3-3Hz Filter

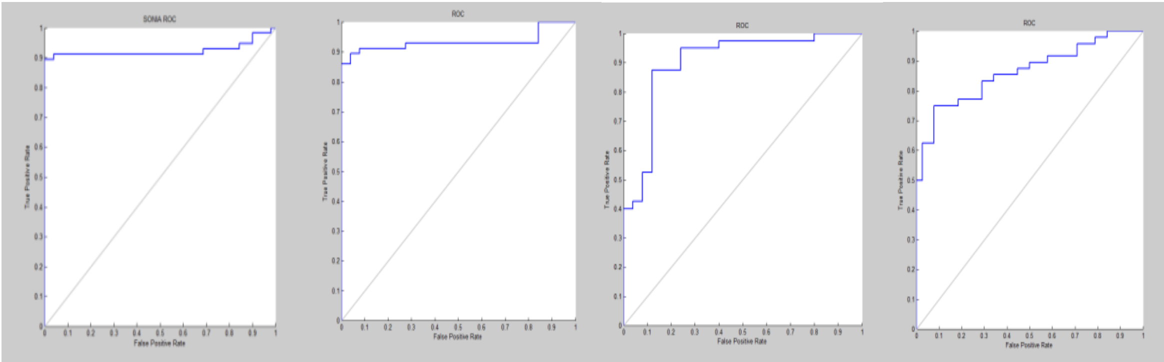
	Sensitivity	Specificity	True Negative	True Positive	AUC
MLP	0.8070	0.8627	0.8000	0.8803	0.88
SONIA	0.9123	0.9451	0.9060	0.9490	0.92
KNNC	0.6830	0.6408	0.6471	0.6853	0.6619
TREEC	0.8846	0.7619	0.8421	0.8214	0.8673
SVC	0.8076	0.8571	0.7826	0.8750	0.9029
DSIA	0.9123	0.8954	0.9013	0.9070	0.93
Fuzzy-SONIA	0.8401	0.7881	0.8561	0.7673	0.92
Elman	0.9092	0.8159	0.8837	0.8247	0.9009

Table 7.38: Mean Error, Standard Deviation and Classifier Accuracy

	Mean Error	Std Error	Accuracy
MLP	0.1681	0.0491	84.87%
SONIA	0.0741	0.0011	92.7778%
KNNC	0.2260	0.0505	66.3137%
TREEC	0.2433	0.569	82%
SVC	0.1761	0.0549	83%
DSIA	0.0870	0.00054	90.1852 %
Fussy-SONIA	0.1535	0.0025	81.1728
Elman	0.1295	0.0675	85.9302 %

ROC Analysis

The ROC curves in Figure 7.17 show an improvement in the performance of the classifiers compared to the ROC curve illustrated in Figure 7.16. The area under the curve for DSIA is 0.93 which is higher than SONIA, which was 0.92. Extending the number of features to 15 has significantly improved the classifiers’ performance. These features have provided classification methods with enough information from each record to allow them to obtain better values in all of the evaluation functions.



a) ROC for SONIA **b) ROC for DSIA** **c) ROC for Fuzzy_SONIA** **d) ROC for Elman**

Figure 7.17: ROC curve for two best performing classifiers

Table 7.39: The parameters selected to build four neural networks

15 Featuers	Time /Second	Hidden	Epoch
MLP	0.785411	40	1000
SONIA	18.6423	55	700
DSIA	25.4202	20	1500
Fuzzy_SONIA	26.8386	47	500
Elman	0.6875	5	1000
4 Featuers	**Time /Second**	**Hidden**	**Epoch**
MLP	0.9521	20	1000
SONIA	19.4	36	700
DSIA	21.3324	7	1500
Fuzzy_SONIA	27.8386	18	500
Elman	0.3658740	5	1000

In order to ensure stability of DSIA, as suggested by Voegtlin (2002), the value of β must be small and hence it has been set to 0.1 and α =0.9. The number of epochs is selected to be 1000-1500. It has been shown from Tables 7.37 and 7.38 that the result from DSIA outperforms some models. Table 7.39 shows that the number of hidden units in the DSIA network is very low compared to the SONIA network. This means that DSIA discovered the information and mapped the data by using a limited number of hidden units compared with other feed-forward neural networks. However, in the second experiment for additional features SONIA performed better than DSIA and this might be related to the extra features being completely different between each individual woman in the same classes.

7.3.5 Discussion Section

In this section, some of the issues faced by the comparison of different neural networks are addressed. This section attempts to elaborate on the observations resultant from all the experiment results.

Most of the uterine EHG signal studies concentrate on predicting true labour, which is based on the last stage of the pregnancy duration. This thesis has studied the uterine EHG signals of women in order to classify the preterm and term deliveries from the early stages of the pregnancy. It has been suggested that ANN is a better solution for nonlinear medical decision support systems than traditional statistical techniques (Li et al. 2000). Therefore, this experiment is based on applying five different types of neural networks, and three of them are applied on medical data for the first time. These neural networks include SONIA, Fuzzy_SONIA, and the newly proposed DSIA neural network.

The evaluation of the classifiers' performance in these experiments has been measured using sensitivity and specificity as the performance evaluation parameters, because these are suitable evaluation measurement for medical classification and specifically for binary output (term, preterm). This study uses sensitivity, which was selected as the true positive detection rate of preterm classes. In terms of specificity, this relates to the detection rate of true negatives or of term classes. In addition, the classifiers have been distinguished using the ROC curve, which is commonly used in medical decision-making. It is a useful method for visualising classifier performance.

7.3.5.1 Neural Network Models compared with the Traditional Statistical Methods for Medical Data Classification

From the two experiments – oversampling and extending the features – the proposed neural network models performed consistently better than the traditional statistical methods in terms of the area under the curves and the accuracy, as presented in Tables 7.35 to 7.38. For the sensitivity and specificity values, we can see that a sensitivity of 76-91% and specificity of 68-89% can be found in the ROC curve of the proposed DSIA model in oversampling and extended features' experiments, respectively. In the SONIA model, a sensitivity of 72-91% and specificity of 69-94% can be found in the ROC curve. On the other hand, the highest result achieved by statistical models rendered a sensitivity of 58-88.46% and a specificity of 61-76.19% in its ROC curve.

7.3.5.2 The Self-Organised Hidden Layer Immune Systems' Networks Compared with the MLP

The results obtained from the oversampling data show that the performance of self-Organised hidden layer immune systems and dynamic links improved the predictive capabilities of the classifiers. The simulation results indicated that the sensitivity and specificity for preterm records obtained with the SONIA network is slightly better than when using the MLP network. This means that the SONIA network has the ability to predict the true positive value of the preterm class; it can also predict the true negative value, which is the term class. More importantly, the proposed DSIA model shows promising results and outperformed the other classifiers. DSIA is a good method of classification, providing better values on almost all the cost functions. This improvement can be associated with the novel combination of supervised and unsupervised learning techniques used in the DSIA model and neural networks in general (Weijters et al. 1997). This has helped to overcome the limitations often found in back-propagation learning. This collaboration of self-organised hidden layer immune systems and recurrent links has overcome the limitation in generalising knowledge of BP learning. The DSIA network has performed well in the classification of uterine EHG signals because it has used SOM unsupervised methods in the hidden layer and the recurrent links. The hidden layer can cluster the input nodes to the centroids of the hidden units, which gives the local network pattern of the input data. The Euclidean distance was utilised to compute the distance between the input units and the centroids of hidden units (Widyanto et al. 2005). Thus, DSIA is able to exploit locality characteristics of the data

In order to evaluate the generalisation ability of SONIA and the proposed DSIA networks, a validation set has been used to compose vectors that are not in the training set and not in the test set. Furthermore, to avoid overfitting of the model to the training data the cost function must minimise the error of the validation data set instead of the training data set. Table 7.40 demonstrates that the DSIA network helps to improve generalisation capabilities of the SONIA network.

Experimental results confirmed that the proposed recurrent neural network improves discrimination and generalisation powers. This is related to using memory in the DSIA network, which saves the output of the hidden units. In general, in cases of binary classification using multilayer feed-forward neural networks, the hidden units learn to explore the useful information from the input pattern and the units in the output layer learn to separate the information given from the hidden layers. Therefore, it is reasonable to provide more information to the hidden units in order to improve the classification performance of the neural network.

7.3.5.3 The Issues Relating to the Neural Networks

There are some issues that have been faced and managed with neural networks, which are:

1. Network Size: generally, all neural networks that have been applied in this experiment contain one hidden layer. The problem we faced was selection of the number of neurons in the hidden layer. Table 7.39 shows the optimum number of neurons in the hidden layer that has been used for each neural network that provides the average result.
2. In terms of comparing the number of hidden units that have been used in the four neural network classifiers, DSIA has the least number of hidden units. This might be related to the existence of the feedback links that allows the hidden layer to explore the hidden pattern with fewer hidden units. The associative memory in the DSIA is trained to store information efficiently.
3. Time: More time was required to build the Fuzzy _SONIA network compared to the DSIA, SONIA and MLP networks.

Table 7.40: The parameters for neural networks

15 Features	MSE Training	MSE Validation	Std training	Std validation	Lr	mom
DSMIA	0.1308	0.1372	0.0657	0.0637	0.001	0.9
SONIA	0.0706	0.0768	0.0281	0.0275	0.01	0.9
Fuzzy_SONIA	0.1353	0.1428	0.0422	0.0342	0.0195	0.9
Four Features	MSE Training	MSE Validation	Std training	Std validation	Lr	mom
DSIA	0.2208	0.2336	0.0087	0.0062	0.005	0.9
Fuzzy_SONIA	0.2424	0.2412	0.0088	0.0096	0.002	0.9
SONIA	0.2227	0.2336	0.0015	0.0062	0.005	0.9

The neural network exhibits the most powerful discriminate capability when combining a self-organised hidden layer with an immune algorithm. In other words, the increase in performance has been achieved by these networks compared with the MLP network. The results report that the three networks that were extended from self-organised hidden layer combined with immune algorithm are the best compared to other classifiers that were used in this experiment. The outcome of the comparison yields that, by applying memory in the SONIA network, it has enhanced the explorative aspect of the hidden layer. Determining the optimal number of hidden units in the hidden layer is very crucial for the learning. These networks involve the immune system which has the ability to automatically decide the best number of hidden units; furthermore, they select the best values of the weights associated with each hidden unit. However, it must be mentioned that the proposed neural network needs more time for learning.

7.3.5.4 The Oversampling and the Additional Extended Feature Experiments

The initial classification with the data set in its original form achieved very low sensitivity, below 20%, while the specificity is higher. This means that the classifiers were classifying most of the cases into the majority class, which are term subjects. The main reason for the ineffective classification was the unequal amount of term records to preterm records. Therefore, in these experiments, the oversample method has significantly improved the sensitivity and specificity rates for all classifiers.

It was noticed that the TPEHG database record used in this experiment is an unbalanced data set of 262 terms and 38 preterms. This means that, if the unbalanced data set was run on the eight classifiers, the results are more likely to be biased. However, to resolve this problem, the oversampling method has been used. This technique generates an equal amount of data and even split between term and preterm records, enhancing the classifier to perform effectively. Furthermore, the first publication of the TPEHG data set was in 2010. However, in 2012, clinical data became freely available. During this experiment, additional features from the TPEHG database which clinical data was considered when analysing the data set. The final experiment results demonstrate that the general performance of classifiers is significantly improved further by comprising the information from the clinical data set, as illustrated in Figure 7.18. The results suggest that these new features were incorporated into the original data set to help enhance the classifier capability in differentiating term and preterm records.

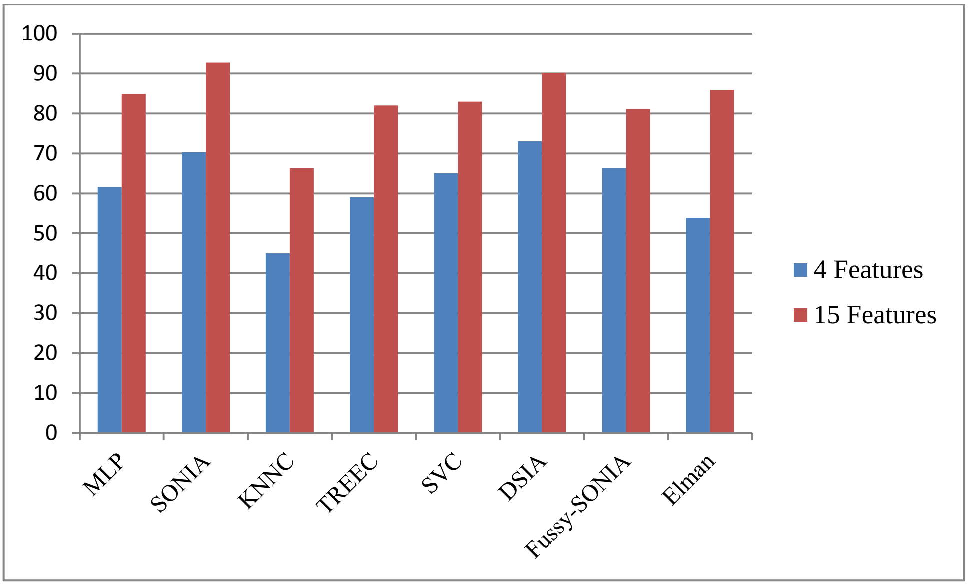


Figure 7.18: The accuracy results for classifiers

Generally, these experiments achieved better results than those in Verdenik and Pajntar (2001), where their results were as follows: a sensitivity of 47%, specificity of 90%, and an error rate of 25%. In addition, the experiments' results are better than those of certain other authors (Baghamoradi, Naji, & Aryadoost, 2011; a. Diab, Hassan, Marque, & Karlsson, 2012; M. O. Diab, Moselm, Khalil, & Marque, 2012; Hassan et al., 2012; B Moslem, Diab, Marque, & Khalil, 2011; B Moslem, Karlsson, Diab, Khalil, & Marque, 2011; B. Moslem, Diab, Khalil, & Marque, 2011, 2012; Bassam Moslem, Khalil, Diab, Chkeir, & Marque, 2011; Bassam Moslem, Khalil, Diab, & Marque, 2012; Bassam Moslem & Khalil, 2011).

However, the findings in Diab, et al. (2010) produced a very high result on the sensitivity rate of 100% and the specificity rate of 94%. Diab et al. (2010) have used several alternative techniques, including artificial neural networks and autoregressive models. However, the data size is much smaller than the data size utilised in these experiments, with 15 preterm and 15 term.

The neural network models developed in this study provided quite acceptable results. The DSIA neural network achieved the higher value of AUC (81-93%) compared to other classifiers; thus, this network can potentially be applied to medical decision support systems.

The experiments' results demonstrate that using Dynamic neural network techniques with a context unit is effective in classifying and analysing real-life data with high confidence:

whether an expectant mother is likely to have a premature birth or have a term birth. Dynamic SONIA has proven that it can be used as an explorative tool to find statistical dependencies between EMG signals. The unsupervised learning in DSIA enables the neural network to determine the statistical properties of input data. Furthermore, the effectiveness of the proposed DSIA in supervised pattern recognition is encouraging for its use on other data sets with a high proportion of accuracy. This network has the ability to produce a more robust classification with better generalisation ability. Learning behaviour of the neural network model enhances the classification properties.

7.4 Chapter Summary

This chapter has presented extensive simulation results of a number of neural network architectures. The first sets of experiments are based on stationary and non-stationary prediction of ten financial time series to forecast the one step and five step ahead predictions. The next sets of experiments are based on using the proposed network to classify and model Uterine EHG signals. The neural network models developed in this thesis provided acceptable results in the two experiments. In the next chapter, the overall outcome of this research work will be presented with suggestions for future works.

CHAPTER 8 CONCLUSION AND FUTURE WORK

8.1 Introduction

This work presents a novel neural network architecture based on the recurrent links and the immune algorithm and its application for financial and medical data analysis.

For financial time series prediction, the research work underlines the important contribution of a new recurrent self-organising multilayer neural network inspired by artificial immune systems based on the Jordan network. In this study historical data of financial time series was utilised with ten data sets. The proposed network performs well in both stationary and non-stationary predictions. This relates to the fact that the proposed network is able to look at the temporal locality of the signal and extract the required information while other networks such as the feed-forward neural networks are more capable of predicting the overall trends in the signals. In terms of applying a regularisation scheme to the DSMIA network, the network provides encouraging results using a weight-decay between the hidden and output nodes.

For medical data analysis, the EHG signals are used, which present uterine contraction during pregnancy duration. This research utilises these signals using the proposed dynamic self-organised network inspired by the immune algorithm based on the Elman recurrent network to classify the EHG signals. These signals were pre-processed and features were extracted before using neural networks classifiers. The main aim in this study was to classify between the potentially preterm and term subjects. This study has proved the ability of the proposed network to perform binary classification. About 76-91% of the preterm subjects were correctly classified using DSIA.

From the number of experiments that have been proposed in this thesis, it can be concluded that the different application of dynamic neural networks for analysis of time series signals was highly successful and promising.

8.2 The Contribution

The construction of the proposed dynamic self-organising multilayer neural network inspired by artificial immune systems is the main contribution and novelty in this research work. The unique characteristics of the dynamic self-organising multilayer neural network inspired by artificial immune systems which combine the properties of supervised and unsupervised networks as well as the recurrent neural network make it suitable and useful for forecasting and classification. The purpose of the proposed network was to look at the improvement achieved when using recurrent links to the structure of the SONIA network. From the number of experiments that have been addressed in this thesis, it can be concluded that the recurrent links have enhanced the performance of the network.

The network has shown its advantages in forecasting both stationary and non-stationary signals, particularly regarding temporally local training behaviour. The proposed recurrent connections give the network a memory, enabling it to recall past behaviours and hence the network produced better results for time series forecasting in comparison to the benchmarked networks. These connections can detect useful predictions for stationary and non-stationary time series data. Although the financial time series is volatile, so it will make more mistakes, the proposed network is able to catch up with the trend in time series. Furthermore, the network generated profits (using the annualised return as a financial measure) for the non-stationary data prediction while most benchmarked networks fail to do so. Therefore, the proposed neural network is a promising tool. From the experiment results, it can be seen that the current study can be a great tool for many forecasters of stock prices or oil prices, as it suggests DSMIA can be effectively used as a financial forecasting tool. The use of DSMIA in financial time series forecasting has demonstrated that the proposed network is possibly beneficial for technical trading to predict daily financial time series.

From the EHG classification result, the suitability of the proposed network for medical data classification has been shown. It indicated that the recurrent neural network is a promising tool for medical classification. The results of this work can encourage more extensive use of recurrent neural networks in different types of medical applications such as classification, and that this use of recurrent neural networks can produce models that are more accurate than the currently used feed-forward neural networks or traditional statistical models. This study has shown the benefits to support and advance some aspects of the healthcare system. The proposed network can be used as a diagnostic tool to support medical experts to make the correct decisions about their patients.

8.3 Future Direction of This Research Work

With the success of the proposed network, there are some further research directions, involving improvements to the proposed network and extending its application approach. These are discussed below.

Different techniques

Future work will be based on applying different techniques to the DSMIA network in order to improve its performance. One of the major limitations of the proposed network is computational performance. Hence, another direction of research must be taken which investigates the best choice of network architecture and this includes the number of inputs and the use of higher order terms in the input units. The utilising of high order terms in the neural network, as suggested by Knowles et al. (2005), can provide reduced computational time and reduced number of input units in the ANN. This may improve the performance of the proposed network. In addition, improving the efficiency of the prediction methods and procedures can be done by combining Elman and Jordan architectures in the proposed network. This combination can enhance the network performance. Future direction will include the use of fuzzy logic in the structure of the proposed dynamic self-organised neural network to improve the classifier performance. Another problem that has been faced is that selection of the best values for the learning rate and momentum parameters that are used in the neural networks is challenging, as there is a need to carefully test for many variables manually by trial and error. One direction for future improvement in this problem is to use some type of genetic algorithm to automatically find suitable neural network parameters.

Utilising the proposed network for unsupervised learning

Since clustering methods have been widely used in different applications of data mining, the adaption of unsupervised learning in the proposed network might serve these different applications, such as medical diagnostics and pattern recognition for large databases, with many attributes. The structure of the proposed network can be adapted for clustering tasks by changing the back-propagation algorithm in the output layer which is supervised learning algorithm to unsupervised learning algorithm. This can extend the application of the proposed network to analyse very complex and large data sets.

DSMIA for pre-processing methods

Since the literature review showed that the RNN can be used to reduce noise from biomedical signals and improve the quality of signals, this can broaden the scope of the medical applications that can benefit from recurrent neural network models to filter and model the biomedical signals. The further work will show the ability of the proposed network to be used as signals pre-processing methods. It will use the network to filter the EHG signals in order to remove noise and improve signal noise ration of the signals.

REFFRENCES

Aamodt, R. (2010). *Using Artificial Neural Networks To Forecast Financial Time Series*. Unpublished Master's thesis,Norwegian University of Science and Technology.

Adhikari, R. & Agrawal, R.K. (2013). *An Introductory Study on Time Series Modeling and Forecasting,* CoRR , http://arxiv.org/abs/1302.6613, [Accessed March 2014].

Agrawal, G., Srivastav, A.K. & Srivastava, A. (2010). A Study of Exchange Rates Movement and Stock Market Volatility. *International Journal of Business and Management*, 5(12), pp.62–73.

Ahangar, R.G., Yahyazadehfar, M. & Pournaghshband, H. (2010). The Comparison of Methods Artificial Neural Network with Linear Regression Using Specific Variables for Prediction Stock Price in Tehran Stock Exchange. *(IJCSIS) International Journal of Computer Science and Information Security*, 7(2), pp.38–46.

Ahmadifard, M., Sadenejad, F., Mohammadi, I. & Aramesh, K. (2013). Forecasting stock market return using ANFIS: the case of Tehran Stock Exchange. *International Journal of Advanced Studies in Humanities and Social Science*, 1(5), pp.452–459.

Akay, M.F. (2000). *Nonlinear Biomedical Signal Processing: Dynamic Analysis and Modeling*, Vol. 2. New York: Wiley-IEEE Press

Akay, M.F. (2009). Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2), pp.3240–3247.

Akrami, A., Akrami, A., Solhjoo, S. & Nasrabad, A. (2005). EEG-Based Mental Task Classification: Linear and Nonlinear Classification of Movement Imagery. In *Engineering in Medicine and Biology 27th Annual Conference*. Shanghai, China, pp. 4626–4629.

Al-Aweel, I.C., Krishnamurthy, K.B., Hausdorff, J.M., Mietus, J.E., Ives, J.R., Blum, A.S., Schomer, D.L. & Goldberger A.L., (1999). Postictal heart rate oscillations in partial epilepsy. *Neurology*, 53(7).

Alexandridis, A. & Livanis, E. (2008). Forecasting Crude Oil Prices Using Wavelet Neural Networks. In *5th FSDET (ΦΣΔΕΤ)*. Athens, Greece, pp. 1–8.

Alnaa, S.E. & Ahiakpor, F., 2011. ARIMA (autoregressive integrated moving average) approach to predicting inflation in Ghana. *Journal of Economics and International Finance*, 3(May), pp.328–336.

Al-shayea, Q.K. (2011). Artificial Neural Networks in Medical Diagnosis. *IJCSI International Journal of Computer Science*, 8(2), pp.150–154.

Arafat, S. (2003). *Uncertanity Modeling For Classification and Analysis of Medical Signals*. PhD thesis, University of Missouri-Columbia.

Araújo, A.F.R. & Darbo Jr., H. (1997). The Role Played by Intralayer and Interlayer Feedback Connections in Recurrent Neural Networks used for Planning. *Journal of the Brazilian Computer Society*, 4(1) pp. 38-49

Arvind, R. Karthik, B., Sriraam, N. & Kannan. J.K. (2010). Automated Detection of PD Resting Tremor Using PSD with Recurrent Neural Network Classifier. *International Conference on Advances in Recent Technologies in Communication and Computing*, pp.414–417. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5655586 [Accessed November 20, 2013].

Aryal, D.R. & Yao-wu, W. (2003). Neural Network Forecasting of the Production Level of Chinese Construction Industry. *Journal of Comparative International Management*, 6(2), pp.45–64.

Atiya, A. (1988). Learning on a General Network. In Anderson, D.A. ed, *Neural Information Processing Systems, NIPS*. New York: American Institute of Physics

Baghamoradi, S., Naji, M. & Aryadoost, H. (2011). Evaluation of cepstral analysis of EHG signals to prediction of preterm labor. In *18th Iranian Conference on Biomedical Engineering*. Tehran, Iran, pp. 1–3.

Baghamoradi, S.M., Naji, M. & Aryadoost, H. (2011). Evaluation of Cepstral Analysis of EHG Signals to Prediction of Preterm Labor. In *18th Iranian Conference on Biomedical Engineering*. Tehran, Iran, pp. 14–16.

Bagherifard, K., Nilashi, M., Ibrahim, O., Janahmadi, N. & Ebrahimi, L. (2012). Comparative Study of Artificial Neural Network and ARIMA Models in Predicting Exchange Rate. *Journal of Applied Sciences, Engineering and Technology*, 4(21), pp.4397–4403.

Baker, P.N. & Kenny, L. (2011). *Obstetrics by Ten Teachers* 19th ed., London: Hodder Arnold

Balli, T. & Palaniappan, R. (2010). Classification of biological signals using linear and nonlinear features. *Physiological Measurement*, 31(7), pp.903–20. Available at: http://www.ncbi.nlm.nih.gov/pubmed/20505216 [Accessed November 20, 2013].

Bansal, A., Kauffman, R.J. & Weitz, R.R. (1993). Comparing The Performance Of Regression and Neural Networks as Data Quality Varies: A Business Value Approach. *Journal of Management Information Systems*, 10, pp.11–32.

Barabanov, N.E. & Prokhorov, D. V (2002). Stability analysis of discrete-time recurrent neural networks. *IEEE Transactions on Neural Networks / a publication of the IEEE Neural Networks Council*, 13(2), pp.292–303. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18244432 [Accessed November 20, 2012].

Barrios, L.A. (2010). *Real-Time Control Of A Multi-Fingered Robot Hand Using EMG Signals*. PhD thesis, San Diego State University.

Bazregar, S. & Mahdinejad, K. (2013). Preterm Birth Detection Using EMG Signal Processing. *Life Science Journal*, 10(3), pp.25–30.

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition.* Oxford University Press, Inc. New York, NY, USA

Box, G.E.P., Jenkins, G.W. & Reinsel, G.C. (1994). *Time Series Analysis - Forecasting and Control*, (3rd ed.), Englewood Cliffs, NJ:Prentice-Hall.

Bozler, E. (1938). Electrical Stimulation and Conduction in Smooth Muscle. *American Journal of Physiology*, 122(3), pp.614–623.

Buhimschi, C., Boyle, M.B. & Garfield, R.E. (1997). Electrical activity of the human uterus during pregnancy as recorded from the abdominal surface. *Obstetrics & Gynecology*, 90(1), pp.102–111.

Buhimschi, C. & Garfield, R.E. (1996). Uterine contractility as assessed by abdominal surface recording of electromyographic activity in rats during pregnancy. *American journal of obstetrics and gynecology*, 174(2), pp.744–53.

Burrows, T.L. & Niranjan, M. (1994). The use of recurrent neural networks for classification. *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pp.117–125. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=366057.

Cajal, S.R.Y. (1990). *New Ideas on the Structure of the Nervous System in Man and Vertebrates,* English translation by N. Swanson and L.W. Swanson., Cambridge, MA: MIT Press.

Campolucci, P. (1998). *A Circuit Theory Approach to Recurrent Neural Network Architectures and Learning Methods*. University of Bologna, Italy. Available at: http://www.dca.fee.unicamp.br/~marcio/ia004/artigos/Campolucci_RNNarchitecturesAndLearn_PhD98.pdf [Accessed March 2012]

Cao, L. & Tay, F.E. (2001). Financial Forecasting Using Support Vector Machines. *Neural Computing & Applications*, 10(2), pp.184–192.

Cao, L.J. & Tay, F.E.H. (2003a). Financial Time Series Forecasting. *IEEE Transactions on Neural Networks*, 14(6), pp.1506–1518.

Cao, L.J. & Tay, F.E.H. (2003b). Support Vector Machine with Adaptive Parameters in Financial time Series Forecasting. *IEEE Transactions on Neural Networks*, 14(6), pp.1506–1518.

Castiglione, F. (2001). Forecasting price increments using an artificial Neural Network, in Complex Dynamics in Economics, a Special Issue of Advances in Complex Systems, 4(1): 45-56, Hermes-Oxford.

Cass, R. & Radl, B. (1996). Adaptive process optimization using Functional-Link Networks and Evolutionary Algorithm, Control Eng. Practice, Vol. 4, No. 11, pp. 1579-1584.

Chappell, G. & Taylor, J.G. (1993). The Temporal Kohonen Map. *Neural Network*, 6, pp.441–445.

Charniak, E. (1991). Bayesian Networks without Tears. *AI Magazine*, 12(4), pp.50–63.

Chen, A.-S. & Leung, M.T. (2005). Performance evaluation of neural network architectures: the case of predicting foreign exchange correlations. *Journal of Forecasting*, 24(6), pp.403–420. Available at: http://doi.wiley.com/10.1002/for.967 [Accessed May 23, 2013].

Chen, H.-Y., Chuang, C.-H. , Yang, Y.-J., Wu, T.-P. (2011). Exploring the risk factors of preterm birth using data mining. *Expert Systems with Applications*, 38(5), pp.5384–5387. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0957417410011619 [Accessed December 23, 2012].

Chen, W.-H., Shih, J.-Y. & Wu, S. (2006). Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *International Journal of Electronic Finance*, 1(1), pp.49–67.

Chen, X., Zhu, X. & Zhang, D. (2010). A discriminant bispectrum feature for surface electromyogram signal classification. *Medical engineering & physics*, 32(2), pp.126–35. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19955011 [Accessed November 4, 2013].

Chendeb, M. Khalil, M., Hewson, D. & Duchêne, J. (2010). Classification of non stationary signals using multiscale decomposition. *Journal of Biomedical Science and Engineering*, 03(02), pp.193–199. Available at: http://www.scirp.org/Journal/PaperDownload.aspx?paperID=1336&fileName=JBiSE20100200010_89721058.pdf [Accessed March 5, 2014].

Cheron, G., Draye, J.-P., Bourgeios, M. & Libert, G. (1996). A Dynamic Neural Network Identification of Electromyograph and Arm Trajectory Relationship during Complex Movements. *IEEE Trans Biomed Eng*, 43(5), pp.552–228.

Cheung, P. (2012). *Evaluating the Effect of Data Preprocessing on the Prediction of Preterm Deliveries from EHG using Machine Learning Algorithms*. MSc Thesis, John Moores University.

Chiang, A. (2010). *Automated Quantification of Human Alpha Rhythms*. PhD Thesis ,The University of Sydney.

Chowdhury, R.H. Reaz, M. B. I. , Ali, M., Bakar, A., Chellappan, K. & Chang, T.G. (2013). Surface electromyography signal processing and classification techniques. *Sensors (Basel, Switzerland)*, 13(9), pp.12431–66. Available at: http://www.ncbi.nlm.nih.gov/pubmed/24048337 [Accessed October 31, 2013].

Christian L Dunis, J.L. and U.S., 2012. Currency trading in volatile markets: Did neural networks outperform for the EUR/USD during the financial crisis 2007–2009? *Journal of Derivatives & Hedge Funds*, 18, pp.2–41.

Chung, J.R., Kwon, J. & Choe, Y. (2009). Evolution of recollection and prediction in neural networks. *2009 International Joint Conference on Neural Networks*, pp.571–577.

Clements, M.P., Franses, P.H. & Swanson, N.R. (2004). Forecasting economic and financial time-series with non-linear models. *International Journal of Forecasting*, 20(2), pp.169–183. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0169207003001432 [Accessed November 17, 2013].

Cunha, J.P. & de Oliveira, P.G. (2000). A new and fast nonlinear method for association analysis of biosignals. *IEEE transactions on bio-medical engineering*, 47(6), pp.757–63. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10833850 [accessed April 2013]

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals and Systems*, 2, pp.303–314.

Dablemont, S., Bellegem, S.V.A.N. & Verleysen, M. (2007). Modelling and Forecasting financial time series of tick data by functional analysis and neural networks. FORECASTING FINANCIAL MARKETS.

Dase, R.K. & Pawar, D.D. (2010). Application of Artificial Neural Network for stock market predictions: A review of literature. *International Journal of Machine Intelligence,*, 2(2), pp.14–17.

Deistler, M. (1996). Time Series Econometrics. *CWI Quarterly*, 9(3), pp.165–179.

Übeyli, D.E. (2010). Recurrent neural networks employing Lyapunov exponents for analysis of ECG signals. *Expert Systems with Applications*, 37(2), pp.1192–1199. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0957417409005806 [Accessed October 29, 2013].

Detienne, K.B., Detienne, D.H. & Joshi, S. A. (2003). Neural Networks as Statistical Tools for Business Researchers. *Organizational Research Methods*, 6(2), pp.236–265. Available at: http://orm.sagepub.com/cgi/doi/10.1177/1094428103251907 [Accessed November 29, 2013].

Deniz, Y. & Karım, S.A. (2013). A Comparison of Linear and Nonlinear Models in Forecasting Market Risk: The Evidence from Turkish Derivative Exchange. *Journal of Economics and Behavioral Studies*, 5(3), pp.164–172.

Devedeux D., Marque, C., Mansour, S., Germain, G & Duchene J. (1993). Uterine electromyography: a critical reviewe. *American Journal of Obstetrics and Gynecology*, 169(6), pp.1636–1653.

Diab, A., Hassan, M. , Marque, C. & Karlsson, B. (2012). Quantitative performance analysis of four methods of evaluating signal nonlinearity: Application to uterine EMG signals. *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp.1045–1048. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6346113.

Diab, M. O., Moslem, B., Khalil, M. & Marque, C. (2012). Classification of uterine EMG signals by using Normalized Wavelet Packet Energy. *2012 16th IEEE Mediterranean Electrotechnical Conference*, pp.335–338. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6196443.

Diab, M.O., Moslem, B., Khalil, M. & Marque, C. (2012) Classification of Uterine EMG Signals by using Normalized Wavelet Packet Energy. 16thIEEE Mediterranean Electrotechnical Conference: 335–338.Diab, M.O. et al., 2010. Classification of Uterine EMG signals using Supervised Classification method. *Biomedical Science and Engineering*, 3(9), pp.837–842.

Diab, M.O., Marque, C. & Khalil, M.A. (2007). Classification for Uterine EMG Signals: Comparison Between AR Model and Statistical Classification Method. *International Journal of Computational Cognition*, 5(1), pp.8–14.

Dijk, E.O. (1999). Analysis of Recurrent Neural Networks with Application to Speaker Independent Phoneme Recognition. MSc thesis ,*Electrical Engineering*. University of Twente.

Dingwell, J.B. (2006). Lyapunov exponents. In M. Akay, ed. *The Wiley Encyclopedia of Biomedical Engineering*. New York: John Wily & Sons Inc.

Doret, M. (2005). Uterine Electromyograpy Characteristics for early Diagnosis of Mifepristone-induced Preterm Labour. *Obstetrics and Gynecology*, 105(4), pp.822–30.

Doya, K. (1995). Supervised Learning in Recurrent Networks. In M. A. Arbib, ed. *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA: The MIT Press, pp. 1–9.

Dunis, C.L. & Williams, M. (2003). *Applications of Advanced Regression Analysis for Trading and Investment, In applied for quntative method fro trading and invenstment* , John Wiley & Sons, New York, pp 1-40.

Dunis, C.L. & Williams, M. (2002). Modelling and Trading the EUR / USD Exchange Rate: Do Neural Network Models Perform Better? *Trading and Regulation*, 8(3), pp.1–24.

Elman, L.J. (1990). Finding Structure in Time. *Cognitive Science*, 14, pp. 179-211'.

Erfanian, A. & Mahmoudi, B. (2005). Real-time ocular artifact suppression using recurrent neural network for electro-encephalogram based brain-computer interface. *Medical & Biological Engineering & Computing*, 43(2), pp.296–305. Available at: http://www.ncbi.nlm.nih.gov/pubmed/15865142.

Espinoza, R., Lombardi, M.J. & Fornari, F. (2009). The role of financial variables in Predicting economic activity,Available at: http://ssrn.com/abstract=1498989 [Accessed August 2012]

Eswaran, H. Wilson, J. D., Murphy, P., Preissl, H. & Lowery, C. L. (2002). Application of wavelet transform to uterine electromyographic signals recorded using abdominal surface electrodes. *The Journal of Maternal-Fetal & Neonatal Medicine: the official journal of the European Association of Perinatal Medicine, the Federation of Asia and Oceania Perinatal Societies, the International Society of Perinatal Obstetricians*, 11(3), pp.158–66. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12380670 [Accessed 5th August 2013]

Evans, D.H. McDicken, W.N., Skidmore, R. & Woodcock, J. P. (1989). *Doppler Ultrasound: Physics Instrumentation and Clinical Applications* 1st ed., London: Wiley.

Faraway, J. & Chatfield, C. (1998). Time series forecasting with neural networks: a comparative study using the airline data. *Journal of the Royal Statistical Society*, 47(2), pp.231–250.

Farkas, I. & Crocker, M.W. (2006). Recurrent networks and natural language: exploiting self-organization. In *in procceding of 28th annual conference of the congitive scienca*.

Fausett, L. (1994). *Fundamentals of neural networks architectures, algorithms, and applications,* Englewood Cliffs, NJ: N. Prentice Hall.

Fei, G. & Yu, Y.L. (1994). A modified Sigma-Pi BP Network with Self-feedback and its Application in Time Series Analysis, *Proceedings of the 5th International Conference,* vol. 2243-508F, pp. 508-515.

Fele-Zorz, G., Kavsek, G., Novak-Antolic, Z. & Jager, F. (2008). A comparison of various linear and non-linear signal processing techniques to separate uterine EMG records of term and pre-term delivery groups. *Medical & Biological Engineering & Computing*, 46(9), pp.911–22. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18437439 [Accessed January 15, 2013].

Fergus, P., Cheung, P., Hussain, A., Al-Jumeily, D., Dobbins, C. & Iram S. (2013). Prediction of Preterm Deliveries from EHG Signals Using Machine Learning. *PloS one*, 8(10), p.p77154. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3810473&tool=pmcentrez&rendertype=abstract [Accessed December 19, 2013].

Forney, E.M. (2011). *Electroencephalogram Classification By Forecasting with Recurrent Neural Network*. PhD thesis, Colorado State University.

Forney, E. M. & Anderson, C.W. (2011). Classification of EEG During Imagined Mental Tasks by Forecasting with Elman Recurrent Neural Networks. In *Proceedings of International Joint Conference on Neural Networks*. San Jose, California: IEEE, pp. 2749–2755 Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6033579 [Accessed January 9, 2013].

Freschi, F., & Repetto, (2006). M. VIS: An Artificial Immune network for Multi-objective optimization. Engineering Optimization, 38(8). p.p 975-996.

Garfield, R.E., Maner, W.L., MacKay, L.B., Schlembach, D. & Saade, G.R. (2005). Comparing uterine electromyography activity of antepartum patients versus term labor patients. *American Journal of Obstetrics and Gynecology*, 193(1), pp.23–9. Available at: http://www.ncbi.nlm.nih.gov/pubmed/16021054 [Accessed January 9, 2013].

Garfield, Robert E & Maner, W.L. (2007). Physiology and electrical activity of uterine contractions. *Seminars in cell & developmental biology*, 18(3), pp.289–95. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2048588&tool=pmcentrez&rendertype=abstract. [Accessed Jun 9, 2013].

Garfield, R E & Maner, W.L. (2007). Uterine Electromyography in Humans – Contractions, Labor, and Delivery. In T. Jarm, P. Kramar, & A. Zupanic, eds. *11th Mediterranean Conference on Medical and Biomedical Engineering and Computing*. Heidelberg, Berlin: Springer, pp. 128–130.

Ghazali, R., Nawi, N.M., Najib, M. & Salleh, M. (2010). Dynamic Ridge Polynomial Neural Network with a Real Time Recurrent Learning Algorithm : Forecasting the S & P 500. *Biomedical Soft Computing and Human Sciences*, 16(2), pp.97–103. Available at: http://www.brain.kyutech.ac.jp/~bmfsa-eng/Published/16_2/16_2_S12.pdf [Accessed January 17, 2012].

Ghazali, R. (2007). *Higher Order Neural Networks For Financial Time Series Prediction*. PhD thesis, Liverpool John Moores.

Ghazali, R.,Hussain, A., Nawi, N. & Mohamad, B. (2009). Non-stationary and stationary prediction of financial time series using dynamic ridge polynomial neural network. *Neurocomputing*, 72(10-12), pp.2359–2367.

Giles, C.L., Lawrence, S. & Tsoi, A.C. (2001). Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference. *Machine Learning*, 44(1/2), pp.161–183.

Giles, C. L., & Maxwell, T. (1987). Learning, invariance and generalization in high-order neural networks. *In Applied Optics*, vol. 26, no. 23, pp.4972-4978.

Giovanis, E. (2010). Application of Feed-Forward Neural Networks Autoregressive Models in Gross Domestic Product Prediction. *World Academy of Science, Engineering and Technology*, (1), pp.646–650.

Goldenberg, R. L., Culhane, J.F., Iams, J.D. & Romero, R. (2008). Epidemiology and causes of preterm birth. *The Lancet*, 371(9606), pp.75–84.

Gondry, J., Marque, C. & Duchene. J. (1993). Electrohysterography during pregnancy: preliminary report. *Biomed Instrum Technol*, 27(4), pp.318–324.

Gooijer, J.G. De & Hyndman, R.J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, 22(44), pp.443–473.

Graupe, D. (2010). Recognition and prediction of individual and combined muscular activation modes via surface EMG analysis. *European Journal of Translational Myology*, 1(3), pp.131–138.

Greenough, A. (2012). Long term respiratory outcomes of very premature birth (<32 weeks). *Seminars in fetal & neonatal medicine*, 17(2), pp.73–6.

Guimarães, G., Lobo, V.S. & Moura-Pires, F. (2002). A Taxonomy of Self-organizing Maps for Temporal Sequence Processing. *Intell Data Anal*, 7(4), pp.1–52.

Guler, N., Ubeyli, E. & Guler, I. (2005). Recurrent neural networks employing Lyapunov exponents for EEG signals classification. *Expert Systems with Applications*, 29(3), pp.506–514. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0957417405000679 [Accessed March 6, 2013].

Gupta, K.C. (2000). Neural Network Structures. In Q. J. Zhang and K. C. Gupta, ed. *Neural Networks for RF and Microwave Design*. Boston, MA: Artech House, pp. 61–103.

Hagan, M.T. & Menhaj, M.B. (1994). Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transactions on Neural Networks*, 5(6), pp.2–6. Available at: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00329697.

Hammer, B., Micheli, A., Sperduti, A. & Strickert, M. (2004). Recursive self-organizing network models. *Neural networks: the official journal of the International Neural Network Society*, 17(8-9), pp.1061–85.

Hammerstrom, D. (1993). Neural networks at work. *IEEE Spectrum*, 30(6), pp.26–32.

Hanley, A. & Mcneil, J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, pp.29–36.

Hassan, M., Alexanderson, A., Terrien, J., Muszynski, C. & Marque, C. (2012). Better Pregnancy Monitoring using Nonlinear Correlation Analysis of External Uterine Electromyography. *IEEE transactions on bio-medical engineering*. 60(4) pp.1160–1166.

Hassan, M., Terrien, J., Alexandersson, A., Marque, C., KarlssonB., 2010. Nonlinearity of EHG signals used to distinguish active labor from normal pregnancy contractions. In *32nd Annual International Conference of the IEEE EMBS*. pp. 2387–2390.

Hassan, M. Terrien J, Karlsson B. & Marque C. (2009). Spatial analysis of uterine EMG signals: evidence of increased in synchronization with term. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*. pp. 6296–9. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19963919 [Accessed April 2013]

Hassoun, M. (1996). *Fundamentals of Artificial Neural Networks*, Cambridge, MA: MIT Press.

Haykin, S. (1998). *Neural Networks: A comprehensive Foundation*, New Jersey: Prentice Hall PTR.

Hebb, D.O. (1949). *The Organization of Behavior*, New York: John Wiley.

Keijzer-Veen, M.G. & van der Heijden, A.J. (2012). The Effect of Preterm Birth on Kidney Development and Kidney Function over Time. In J. C. Morrison, ed. *Preterm Birth - Mother and Child*. InTech. pp.341-356

Hellbach, S., Eggert, J.P. & Edgar, K. (2008). Echo State Networks for Online Prediction of Movement Data – Comparing Investigations. In *Artificial Neural Networks (ICANN)*. Springer, pp. 710–719.

Hellstrom, T. & Holmstrom, K. (1997). *Predicting the stock market*, Technical Report Series IMa-TOM-1997-07,. Malardalen University, Vasteras, Sweden.

Herrera, J.L. (1999). *Time Series Prediction Using Inductive Reasoning Techniques*. Instituto de Organizacion y Control de Sistemas Industriales.

Hertz, J., Palmer, R. & Krogh, A. (1991). *Introduction to the Theory of Neural Computation*, Redwood City: Addison-Wesley.

Hill, T. & Lewicki, P. (2007). *STATISTICS: Methods and Applications*, Wiley, New York.

Ho, S.., Xie, M. & Goh, T.N. (2002). A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction. *Computers & Industrial Engineering*, 42(2-4), pp.371–375.

Holst, A. (1997). *The Use of a Bayesian Neural Network Model for Classification Tasks*. PhD thesis, Department of Numerical Analysis and Computing, Royal Institute of Technology, Stockholm, Sweden, (September).

Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*, 79(8), pp.2554–2558.

Hornik, M., Stinchcombe, M. & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 2, pp.359–366.

Huang, W. & Wu, J. (2009). Human action recognition using Recursive Self Organizing map and longest common subsequence matching. *2009 Workshop on Applications of Computer Vision (WACV)*, pp.1–6. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5403130.

Huang, W., Lai, K.K., and Nakamori, Y. (2004). Forecasting foreign exchange rates with artificial neural networks: A review. *International Journal of Information Technology & Decision Making*, 3(1), pp.145–165.

Huet, B. (1993). *Recurrent Neural Networks for Temporal Sequences Recognition*. MSc Thesis, University of Westminste.

Husain, H., Khalid, M. & Yusof, R. (2008). Direct Model Reference Adaptive Controller Based-On Neural-Fuzzy Techniques for Nonlinear Dynamical Systems. *American Journal of Applied Sciences*, 5(7), pp.769–776. Available at: http://www.thescipub.com/abstract/?doi=ajassp.2008.769.776.

Husken, M. & Stagge, P. (2003). Recurrent neural networks for time series classiÿcation. *Neurocomputing*, 50, pp.223–235.

Hussain, A.J., Ghazali, R., Al-jumeily, D., Merabti, M. & John, L. (2006). Dynamic Ridge Polynomial Neural Network for Financial Time Series Prediction. IEEE International conference on Innovation in Information Technology, IIT06, 2006, Dubai.

Hussain, A.J. Knowles, A., Lisboa, P. & El-Deredy, W. (2008). Financial time series prediction using polynomial pipelined neural networks. *Expert Systems with Applications*, 35(3), pp.1186–1199. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0957417407003442 [Accessed December 16, 2013].

Hussain, A.J. & Liatsis, P. (2002). Recurrent Pi-Sigma networks for DPCM image coding. *Neurocomputing*, 55, pp.363–382.

Iams, J.D. (2003). Prediction and early detection of preterm labor. *Obstetrics and Gynecology*, 101(2), pp.402–12. Available at: http://www.ncbi.nlm.nih.gov/pubmed/12576267[Accessed May 2013]

Iams, J.D. (2007). Preterm birth. In S. J. Gabbe SG, Niebyl JR, ed. *Obstetrics: Normal and Problem Pregnancies*. Philadelphia, Pa, USA: Churchill Livingstone Elsevier, pp. 668–712.

Ilbay, K., Übeyli, E.D., Ilbay, G. & Budak, F. (2011). A New Application of Recurrent Neural Networks for EMG-Based Diagnosis of Carpal Tunnel Syndrome. In H. Cardot, ed. *Recurrent Neural Network for Temporal Data Processing*. Croatia: InTech.

Hussain, A.J. & Al-jumeily, D. (2007). How Good Is the Backpropogation Neural Network Using a Self-Organised Network Inspired by Immune Algorithm (SONIA) When Used for Multi-step Financial Time Series Prediction? *ISNN (2)*, 4492(2), pp.921–930.

Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1(4), pp.295–307. Available at: http://linkinghub.elsevier.com/retrieve/pii/0893608088900032 [Accessed March 2013]

Jaeger, H. (2001). *The echo state approach to analysing and training recurrent neural networks - with an Erratum note*, Available at: http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf [Accessed May 2013]

Jaeger, H. & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667), pp.78–80.

Jayarekha, P. & Nair, T.R.G. (2009). Prefetching of VoD Programs Based On ART1 Requesting Clustering. *International Journal of Computer Science and Information Security (IJCSIS),*, 5(1), pp.128–134.

Jiao, L., Gong, M., Ma, W., & Shang, R. (2008). Multi-Objective Optimization Using Artificial Immune Systems. *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. IGI Global, p.p106-147.

Jordan, M.I. (1990). Attractor dynamics and parallelism in a connectionist sequential machine. In *Artificial Neural Networks*. NJ, USA: IEEE Press Piscataway, pp. 112 – 127.

Joshi, S., Shenoy, D., Rrashmi, V., Venugopal, K.R. & Patnaik, L. M. (2010). Classification of Alzheimer's Disease and Parkinson's Disease by Using Machine Learning and Neural Network Methods. *Second International Conference on Machine Learning and Computing*, pp.218–222.

Jurik, M. (1999). The notion of an optimal forecast horizon. Turbulence Lecture Notes in Mathematics , 898, p.1999.

Kamgaing, T. (2005). *Competing Neural Networks as Models for Non Stationary Financial Time Series*. Ph.D. Thesis, University of Kaiserslautern.

Kamruzzaman, J. (2004). ANN-Based Forecasting of Foreign Currency Exchange Rates. *Neural Information Processing - Letters and Reviews*, 3(2), pp.49–58.

Kamruzzaman, J. & Sarker, R. A. (2003). Forecasting of currency exchange rates using ANN: a case study. In *International Conference on Neural Networks and Signal Processing.* Nanjing, China: IEEE, pp. 793–797 Vol.1.

Kavsek G, Pajntar M, Leskosek B (1999). Electromyographic activity of the uterus above the placental implantation site. *Gynecol Obstet Invest*, 48(2).

Khemphila, A. & Boonjing, V. (2011). Heart Disease Classification Using Neural Network and Feature Selection. *2011 21st International Conference on Systems Engineering*, pp.406–409. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6041870 [Accessed April 2, 2013].

Khemphila, A. & Boonjing, V. (2012). Parkinsons Disease Classification using Neural Network and Feature selection. *Engineering and Technology*, pp.15–18.

Khoa, N.L.D., Sakakibara, K. & Nishikawa, I. (2006). Stock Price Forecasting using Back Propagation Neural Networks with Time and Profit Based Adjusted Weight Factors. In *ICASE*. pp. 5484–5488.

Kim, S.-S. (1998). Time-delay recurrent neural network for temporal correlations and prediction. *Neurocomputing*, 20(1-3), pp.253–263. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0925231298000186.

Kimoto, T., Asakawa, K., Yoda, M. & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In *1990 IJCNN International Joint Conference on Neural Networks*. San Diego, CA, USA: IEEE, pp. 1–6 vol.1. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5726498[Accessed Jun 2012].

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *International Joint Conference on Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers.

Konrad, P. (2005). The ABC of EMG: A Practical Introduction to Kinesiological Electromyography. *Noraxon INC. USA*, 1.0.

Koskela, T., Varsta, M., Heikkonen, J. & Kaski, K. (1998). Temporal Sequence Processing using Recurrent SOM. In *Proceedings of the Second International Conference Knowledge-Based Intelligent Electronic Systems*. Adelaide, SA: IEEE, pp. 21–23. Available at: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=725861&tag=1.

Kosmatopoulos, E.B. & Christodoulou, M. A. (1994). Stability, robustness and approximation properties of gradient recurrent high-order neural networks. *International Journal of Adaptive Control and Signal Processing*, 8(4), pp.393–406. Available at: http://doi.wiley.com/10.1002/acs.4480080408.

Kremer, S.C. (1995). On the computational power of Elman-style recurrent networks. *IEEE Transactions on Neural Networks / a publication of the IEEE Neural Networks Council*, 6(4), pp.1000–4. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18263388[accessed March 2013]

Krogh, A. & Hertz, J. (1995). A Simple Weight Decay Can Improve Generalization. In *Advances in Neural Information Processing Systems*, 4, pp.950–957.

Krollner, B. (2011). *Risk Management in the Australian Stock market using Artificial Neural Network*s. PhD Thesis, School of Information Technology.

Ku, C. & Lee, K.Y. (1995). Diagonal Recurrent Neural Networks for Dynamic Systems Control. *IEEE Transactions on Neural Networks*, 6(1).

Kurihara, Y. (2006). The Relationship between Exchange Rate and Stock Prices during the Quantitative Easing Policy in Japan. *International Journal of Business*, 11(4).

Lagerholm, M., Peterson, C., Braccini, G., Edenbrandt, L. & Sörnmo, L. (2000). Clustering ECG complexes using hermite functions and self-organizing maps. *IEEE Transactions on Bio-medical Engineering*, 47(7), pp.838–48. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10916254.

Larsen, J., Svarer, C., Andersen, L. N., & Hansen, L. K. (1998)Adaptive Regularization in Neural Network Modeling. *In Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg*. pp. 113–132.

Lasko, T.A., Bhagwat, J.G., Zou, K.H. & Ohno-Machado, L. (2005). The use of receiver operating characteristic curves in biomedical informatics. *Journal of Biomedical Informatics*, 38(5), pp.404–15.

Lehnertz, K., Andrzejak, R.G., Arnhold, J., Kreuz, T., Mormann, F., Rieke, C., Widman, G. & Elger, C. (2001). Nonlinear EEG Analysis in Epilepsy: Its Possible Use for Interictal Focus Localization, Seizure Anticipation, and Prevention. *Journal of Clinical Neurophysiology*, 18(3), pp.209–222.

Leman, H., Marque, C. & Gondry, J. (1999). Use of the Electrohysterogram signal for characterization of contractions during pregnancy. *IEEE Transactions on Biomedical Engineering*, 46(10), pp.1222–9.

Leondes, C.T. (2010). *Intelligent Knowledge-Based Systems: Business and Technology in the New Millennium.* Kluwer Academic Press. C. T. Leondes, ed., Springer.

Leskošek, B. & Pajntar, M. (2002). Electromyogaphy of The Pregnant Uterus in Humans and Sheep. In *IFMBE Proceedings*. pp. 544–545.

Li, Y.C., Liu, L., Chiu, W.T. & Jian, W. S. (2000). Neural network modeling for surgical decisions on traumatic brain injury patients. *International Journal of Medical Informatics*, 57(1), pp.1–9. Available at: http://www.ncbi.nlm.nih.gov/pubmed/10708251[Accessed May 2013]

Lin, T.-W. & Yu, C.-C. (2009). Forecasting Stock Market with Neural Networks. *SSRN Electronic Journal*. Available at: http://www.ssrn.com/abstract=1327544[Accessed Jun 2012]

Ling, S.H., Leung, F.H., Leung, K.F., Lam, H.K. & Iu, H.C. (2007). An Improved GA Based Modified Dynamic Neural Network for Cantonese-Digit Speech Recognition. In M. Grimm & K. Korschel, eds. *Robust Speech Recognition and Understanding*. Vienna, Austria: I-Tech, p. 460.

Liu, B., Wang, M., Yu, H., Yu, L. & Liu, Z. (2005). Study of Feature Classification Methods in BCI Based on Neural Networks. In *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*. China, pp. 2932–5. Available at: http://www.ncbi.nlm.nih.gov/pubmed/17282857 [Accessed May 2013]

Lobo, J.M., Jiménez-Valverde, A. & Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, 17(2), pp.145–151.

Lu, Nan, Wang, Jihong, Zhang, Chenghui, Vatish, Manu and Randeva, Harpal S. (2007) Uterine electromyography features extraction for classification of term and pre-term signals. *In: International Conference on Complex Systems and Applications*, Jinan, People's Republic of China, June 08-10, 2007. Published in: Dynamics of Continuous Discrete and Impulsive Systems. Series B. Applications & Algorithims, Vol.14 (No.4). pp. 1675-1679.

Lucovnik, M., Maner, W.L., Chambliss, L.R., Blumrick, R., Balducci, J., Novak-Antolic, Z. & Garfield, R.E. (2011). Noninvasive uterine electromyography for prediction of preterm delivery. *American Journal of Obstetrics and Gynecology*, 204(3), pp.228.e1–10.

Machado, L.O. (1996). *Medical Application of Artificial Network Connectionist Models of Survival*. Redwood, CA: Stanford University Press.

Mahdi, A. (2010). *The Application of Neural Network in Financial Time Series Analysis and Prediction Using Immune System*. MSc thesis, Liverpool John Moores University.

Mahdi, A., Hussain, A.J., Lisboa, P. & Al-Jumeily, D. (2009). The Application of the Neural Network Model Inspired by the Immune System in Financial Time Series Prediction. In *Developments in eSystems Engineering (DESE)*. pp. 370–376.

Mahdi, A.A., Hussain, A.J. & Al-Jumeily, D. (2010). The Prediction of Non-Stationary Physical Time Series Using the Application of Regularization Technique in Self-organised Multilayer Perceptrons Inspired by the Immune Algorithm. *2010 Developments in E-systems Engineering*, pp.213–218. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5633838 [Accessed February 11, 2014].

Makarov, V., Song, Y., Velarde, M.G., Hübner, D. & Cruse, H. (2008). Elements for a general memory structure: properties of recurrent neural networks used to form situation models. *Biological cybernetics*, 98(5), pp.371–95.

Makeshwar, M.S., Rajgure, N.K. & Pund, M. (2010). Object Identification using Neural Network. International Journal of Computer Science and Application , pp.29–32.

Maltarollo, V.G., Honório, K.M. & Borges, A. (2013). Applications of Artificial Neural Networks in Chemical Problems. In K. Suzuki, ed. *Artificial Neural Networks - Architectures and Applications*. Croatia: InTech.

Man-chung, C., Chi-cheong, W. & Chi-chung, L. (2000). Financial Time Series Forecasting by Neural Network Using Conjugate Gradient Learning Algorithm and Multiple Linear Regression Weight Initialization. *Computing in Economics and Finance*. Barcelona, Society for Computational Economics

Maner, W.L., Garfield, R.E., Maul, H., Olson, G. & Saade, G. (2003). Predicting term and preterm delivery with transabdominal uterine electromyography. *Obstetrics & Gynecology*, 101(6), pp.1254–1260.

Mangham, L.J., Petrou, S., Doyle, L.W., Draper, E.S. & Marlow. N. (2009). The Cost of Preterm Birth Throughout Childhood in England and Wales. *Pediatrics2*, 123(2), pp.312–327.

Mankar, V.R. & Ghatol, A.A. (2009). Design of Adaptive Filter Using Jordan/Elman Neural Network in a Typical EMG Signal Noise Removal. *Advances in Artificial Neural Systems*, 2009, pp.1–9. Available at: http://www.hindawi.com/journals/aans/2009/942697/ [Accessed November 12, 2013].

Marque, C.K., Terrien, J., Rihana, S. & Germai, G. (2007). Preterm labour detection by use of a biophysical marker: the uterine electrical activity. *BMC pregnancy and childbirth*, 7 Suppl 1, p.S5. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1892062&tool=pmcentrez&rendertype=abstract [Accessed January 15, 2013].

Marshall, J. (1962). *Regulation of activity in uterine smooth muscle*, Physiol Rev Suppl, 5, pp.213-217.

Al-Mashhadany, Y.I. (2012). Recurrent Neural Network with Human Simulator Based Virtual Reality. In M. E. and M. Mysara, ed. *Recurrent Neural Networks and Soft Computing Source*. InTech.

Masters, T. (1993). *Practical Neural Network Recipes in C++* 5th ed., San Francisco, CA: Morgan Kaufmann Publishers In; Pap/Dskt.

McCulloch, W. & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematic y al Biophysics*, 5, pp.115–133.

McQueen, T. A., Hopgood, A. A., Tepper, J. A., & Alle, T. J., (2003). A Recurrent Self-Organizing Map for Temporal Sequence Processing. Applications and Science in Soft Computing, A. Lotfi & J.Garibaldi (eds.), Advances in Soft Compuing Series, Springer , pp.3–8.

Meng, Q., Zhou, W., Chen, Y. & Zhou, J. (2010). Feature analysis of epileptic EEG using nonlinear prediction method. *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2010, pp.3998–4001. Available at: http://www.ncbi.nlm.nih.gov/pubmed/21097278[Accessed March 2013]

Mengistu, S.G., Quick, C.G. & Creed, I.F., 2013. Nutrient export from catchments on forested landscapes reveals complex nonstationary and stationary climate signals. *Water Resources Research*, 49(6), pp.3863–3880. Available at: http://doi.wiley.com/10.1002/wrcr.20302 [Accessed October 2013].

Mesnil, G., He, X., Deng, L., Bengio, Y., 2013. Investigation of Recurrent - Neural - Network Architectures and Learning Methods for Spoken Language Understanding. *Interspeech*,2.

Miniani, A. A. & Williams, R. D. (1990). Acceleration of back-propagation through learning rate and momentum adaptation. *Proceedings of International Joint Conference on Neural Networks*. San Diego: CA. 1, pp. 676-679.

Falk, M., Marohn, F., Michel, D.R., Mackey, M., Spachmann, C.S. & Englert, S. (2011). *A First Course on Time Series Analysis,* University of Wuerzburg.

Micheli, A. (2003). *Recursive Processing of Structured Domains in Machine Learning*. NY: Wiley.

Michie, E.D., Spiegelhalter, D.J. & Taylor, C.C. (1994). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, Hertfordshire, U.K.

Miller, A.S., Blott, B.H. & Hames, T.K. (1992). Review of neural network applications in medical imaging and signal processing. *Medical and Biological Engineering and Computing*, 30(5), pp.449–464.

Miller, C, J. (2008). *Real-Time Feature Extraction and Classification of Prehensile*. Master's thesis. San Diego State University.

Mirea, L. & Marcu, T. (2002). System identification using Functional-Link Neural Networks with dynamic structure. In *15th Triennial World Congress*. Barcelona, Spain.

Mishra, S. & Patra, S.K. (2008). Short term load forecasting using a novel recurrent neural network. In *TENCON 2008 - 2008 IEEE Region 10 Conference*. IEEE, pp. 1–6.

Diab, M.O., El-Merhie, A.& El-Halabi, L.K.N. (2010). Classification of uterine EMG signals using supervised classification method. *Journal of Biomedical Science and Engineering*, 03(09), pp.837–842. Available at: http://www.scirp.org/journal/PaperDownload.aspx?DOI=10.4236/jbise.2010.39113 [Accessed April 2, 2013].

Moslem, B., Khalil, M., Diab, M., Chkeir, A. & Marque, C. (2011). A Multisensor Data Fusion Approach for Improving the Classification Accuracy of Uterine EMG Signals. In *Electronics, Circuits and Systems (ICECS), 2011 18th IEEE International Conference on 11th-14th Dec*. Beirut, Lebanon: IEEE, pp. 93–96.

Moslem, B., Khalil, M., Diab, M., Chkeir, A., Marque, C. (2012). Classification of multichannel uterine EMG signals by using a weighted majority voting decision fusion rule. *2012 16th IEEE Mediterranean Electrotechnical Conference*, pp.331–334. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6196442 [Accessed March 2013]

Moslem, B., Diab, M. O., Khalil, M. & Marque, C. (2011). Classification of multichannel uterine EMG signals by using unsupervised competitive learning. In *IEEE Workshop on Signal Processing Systems*. pp. 267–272.

Moslem, B., Diab, M. O., Khalil, M. & Marque, C. (2012). Classification of multichannel uterine EMG signals using a reduced number of channels. *2012 8th International Symposium on Mechatronics and its Applications*, pp.1–4.

Moslem, B, Diab, M.O., Marque, C. & Khalil, M. (2011). Classification of multichannel uterine EMG signals. *Conference proceedings :Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2011, pp.2602–5.

Moslem, B, Karlsson, B., Diab, M.O., Khalil, M., Marque, C. (2011). Classification performance of the frequency-related parameters derived from uterine EMG signals. *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2011, pp.3371–4. Available at: http://www.ncbi.nlm.nih.gov/pubmed/22255062.

Moslem, B., Khalil, M., Marque, C., & Diab, M. O. (2010). Energy Distribution Analysis of Uterine Electromyography Signals. *Journal of Medical and Biological Engineering*, 30(6), pp.361–366.

Moslem, B. & Khalil, M. (2011a). A Multisensor Data Fusion Approach for Improving the Classification Accuracy of Uterine EMG Signals. , (Mv), pp.93–96.

Moslem, B. & Khalil, M. (2011b). Combining Multiple Support Vector Machines for Boosting the Classification Accuracy of Uterine EMG Signals. *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Lebanon, (Mv), pp.631–634.

Mourik, J. V. (2006). Dynamics and Topographic Organization of Recursive Self-Organizing Maps ˇ Peter Ti, *Neural Computation, Massachusetts Institute of Technology*. 18, pp.2529–2567.

Mozer, M.C. (1989). A Focused Backpropagation Algorithm for Temporal Pattern Recognition. *Complex System*, 3, pp.349–381.

Mozer, M.C. (1994). Neural net architectures for temporal sequence processing. In wigend, A. S., A, G.., eds: Time series predction: Forcasting the future and understanding the past, Reaading, MA, Addison-Wesley. pp.243–264.

Muglia, L.J. & Katz, M. (2010). The Enigma of Spontaneous Preterm Birth. *N Engl J Med*, 362(6), pp.529–35.

Mars, N.J. & Lopes da Silva, F.H. (1983). Propagation of Seizure Activity in Kindled Dogs. *Electroencephalogr Clin Neurophysiol*, 56, pp.194–209.

Office of National Statistics (ONS), (2013) Gestation-specific Infant Mortality in England and Wales, 2011, http://www.ons.gov.uk/ons/dcp171778_329137.pdf [accessed March 2013]

Pacifici, F. (2010). *Novel Neural Network-based Algorithms for Uban Classification and Change Detection from Satellite Imagery*. Tor Vergata University, Rome, Italy.

Pajntar, M., Verdenik, I., Pusenjak, S., Rudel, D. & Leskosek, B. (1998). Activity of smooth muscles in human cervix and uterus. *European journal of obstetrics, gynecology, and reproductive biology*, 79(2), pp.199–204.

Pedersen, M.W. (1997). *Optimization of Recurrent Neural Networks for Time Series Modeling*. PhD thesis. Technical University of Denmark.

Petridis, V., Kehagias, A., Petrou, L., Bakirtzis, A., Maslaris, N., Panagiotou, H., Kehagias, A., Kiartzis, S. (2001). "A Bayesian Multiple Models Combination Method for Time Series This paper has appeared in the journal : A Bayesian Multiple Models Combination Method for Time Series Prediction. *Journal of Intelligent and Robotic Systems*, 31, pp.69–89.

Petrosian, A. A., Prokhorov, D.V., Schiffer, R.B. & Petrosian, A. A. (2001). Recurrent Neural Network based Approach for Early Recognition of Alzheimer's Disease in EEG. Recurrent Neural Network based Approach for Early Recognition of Alzheimer's Disease in EEG. *Clinical Neurophysiology*, 112(8), pp.1378–1387.

Petrosian, A., Prokhorov, D.V., Homan, R., Dashei, R. & Ii, D. W. (2000). Recurrent neural network based prediction of epileptic seizures in intra- and extracranial EEG. *Neurocomputing*, 30, pp.201–218.

Phinyomark, A., Chujit, G., Phukpattaranont, P. & Limsakul, C. (2012). A preliminary study assessing time-domain EMG features of classifying exercises in preventing falls in the elderly. *9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp.1–4.

Physionet.org, (2010). *The Term-Preterm EHG Database (TPEHG DB)*. Available at: http://www.physionet.org/pn6/tpehgdb [Accessed March 2013].

Pissarenko, D. (2002). *Neural Networks For Financial Time Series Prediction: Overview Over Recent Research*, BSc thesis, Computer Studies Thesis, University of Derby in Austria

Pollock, D.S.G. (1987). The Methods of Time-Series Analysis. *Interdisciplinary Science Reviews*, 12(2), pp.1–19.

Pollock, D.S.G. (2001). Methodology for trend estimation. *Trends in Economic* , pp.1–18.

Poon, S.-H. (2005). *A Practical Guide to Forecasting Financial Market Volatility* J. Wiley, T. Atrium, & S. Gate, eds., West Sussex, England: John Wiley & Sons, LTD.

Porter-hudak, S. (1990). An Application of the Seasonal Fractionally Differenced Model to the Monetary Aggregates. *Journal of the American Statistical Association*, 85(410), pp.338–344.

Rabotti, C. (2010). *Characterization of uterine activity by electrohysterography*. PhD Thesis, Technische Universteit Eindhoven.

Rabotti, C., Mischi, M.,van Laar, J., Oei, G. & Bergmans, J. (2007). Electrohysterographic analysis of uterine contraction propagation with labor progression: a preliminary study. *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2007, pp.4135–8.

Rabotti, C., Mischi, M.,van Laar, J., Oei, G., Bergmans, J. (2008). Estimation of internal uterine pressure by joint amplitude and frequency analysis of Electrohysterographic signals. *Physiological measurement*, 29(7), pp.829–41. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18583724 [Accessed January 13, 2014].

Rabotti, C., Mischi, M., Oei, G., Bergmans, J. (2010). Noninvasive estimation of the electrohysterographic action-potential conduction velocity. *IEEE transactions on bio-medical engineering*, 57(9), pp.2178–87.

Ramadevi, R., Sheela, B. & Prakash, V. (2012). Role of Hidden Neurons in an Elman Recurrent Neural Network in Classification of Cavitation Signals. *International Journal of Computer Application*, 37(7), pp.9–13.

Rattihalli, R., Smith, L. & Field, D. (2012). Prevention of preterm births: are we looking in the wrong place? *Edition, Archives of disease in childhood. Fetal an dneonatal*, 97(3), pp.160–1.

Razmjooy, N., Mousavi, B.S., Sargolzaei, P. & Soleymani, F. (2011). Image thresholding based on evolutionary algorithms. *International Journal of the Physical Sciences* 6(31), pp.7203–7211.

Read, M., Andrews, P., & Timmis.J., (2011). An Introduction to Artificial Immune Systems. *The Handbook of Natural Computing*, edited by Rozenberg, G., Bäck, T., & Kok. J., Springer.

Rhaman, M.K. & Endo, T. (2008). Recurrent neural network classifier for Three Layer Conceptual Network and performance evaluation. *2008 11th International Conference on Computer and Information Technology*, pp.747–752. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4803087 [Accessed July 2013].

Ripley, B. (1996). *Pattern recognition and neural networks.*, Cambridge, UK: Cambridge University Press.

Ristanoski, G. & Bailey, J. (2011). Distribution Based Data Filtering for Financial Time Series Forecasting. In *24 Australasian Joint Conference on Artificial Intelligence*. Perth, Australia, pp. 122–132.

Rojas, R. (1996). *Neural Networks Introduction, A Systematic Introduction,* 1st ed., NY: Springer.

La Rosa, P.S., Nehorai, A., Eswaran, H., Lowery, C.L. & Preiss, H. (2007). Detection of uterine MMG contractions using a multiple change-point estimator and K-means cluster algorithm. *International Congress Series*, 1300, pp.745–748. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0531513107001094 [Accessed January 15, 2013].

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), pp.386–408.

Rout, M., Majhi, B., Mohapatra, U., Mahapatra, R. (2012). Stock Indices Prediction Using Radial Basis Function Neural Network. In B. K. Panigrahi et al., eds. *Swarm, Evolutionary, and Memetic Computing*. Heidelberg, Berlin: Springer, pp. 285–293.

Roverso, D. (2000). Multivariate temporal classification by windowed wavelet decomposition and recurrent neural networks. In*: Proceedings of the 3rd ANS International Topical Meeting on Nuclear Plant Instrumentation*. Washington, DC, USA, Control and Human-Machine Interface Technologies, pp. 527–538.

Roverso, D. (2002). Plant diagnostics by transient classification: The ALADDIN approach. *International Journal of Intelligent Systems*, 17(8), pp.767–790.

Rumelhart, D.E. & McClelland, J.L. (1986). *Parallel Distributed Processing*, Cambridge, MA: MIT Press.

Sá, J.A., Rocha, B., Almeida, A., Souza, J. R. (2011). Recurrent Self-Organizing Map for Severe Weather Patterns Recognition. In El Hefnawi, M. and Mysara, M., eds. *Recurrent Neural Networks and Soft Computing*. Croatia: InTech. Available at: http://www.intechopen.com/books/recurrent-neural-networks-and-soft-computing/-recurrent-self-organizingmap-.

Salhi, M.S., Arous, N. & Ellouze, N. (2009). Principal temporal extensions of SOM : Overview. *International Journal of Signal Processing, Image Processing and Pattern Recognition*. 2(4), pp.61-84.

Sarbaz, Y., Gharibzadeh, S., Towhidkhah,F., Banaie, M. & Jafari, A. (2011). A Gray-Box Neural Network Model of Parkinson's Disease Using Gait Signal. *Basic and Clinical Neuroscience* , Spring 2(3), pp.33–42.

Scarborough, D., & Somers, M.J. (2006). *Neural networks in organizational research: applying pattern recognition to the analysis of organizational behavior*, Washington, DC: American Psychological Association.

Schmidhuber, J., Wierstra, D., Gagliolo, M. & Gomez. F. (2007). Training Recurrent Networks by Evolino. *Neural Computation*, 19(3), pp.757–779.

Schwaerzel, R. & Bylander, T. (2006). Predicting currency exchange rates by genetic programming with trigonometric functions and high-order statistics. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. Seattle, Washington, USA: ACM Press.

Sfetsos, A. (2000). A comparison of various forecasting techniques applied to mean hourly wind speed time series. *Renewable Energy*, 21(1), pp.23–35. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0960148199001251 [Accessed May 2012].

Shachmurove, Y. & Witkowska, D. (2000). Utilizing Artificial Neural Network Model to Predict Stock Markets. *CARESS Working Paper*, pp.0–25.

Shaker, M.M. (2005). EEG Waves Classifier using Wavelet Transform and Fourier Transform. *International Journal of Biological and Life Sciences*, 1(2), pp.85–90.

Shen, X., Gao, X.Z. & Bie, R. (2008). Artificial Immune Networks:Model and Application. *International Journal of Computational Intelligence System*, 1, pp.168–176.

Shi, S.Q., Maner, W.L., Mackay, L.B. & Garfield, R.E. (2008). Identification of Term and Pre-Term Labor in Rats using Artificial Neural Networks on Uterine EMG Signals. *Am J Obstet Gynecol.*, 198(2), pp.1–10.

Shin, T. & Han, I. (2000). A Hybrid System Using Multiple Cyclic Decomposition Methods and Neural Network Techniques for Point Forecast Decision Making. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Hawaii: IEEE, pp. 1–10.

Shoeb, A.H. (2009). *Application of Machine Learning to Epileptic Seizure Onset Detection and Treatment*. PhD thesis, Masssachusetts Institute of Technology.

Shortliffe, E.H., Perreault, L.E., Wiederhold, G. & Fagan, L.M. (1990). *Medical Informatics: Computer Applications in Medicine*, Reading: Addison-Wesley Publishing.

Silva, L.E. V., Duque, J. J. , Guzo, M. G. , Soares, I. & Tin, R. (2010). Medical Multivariate Signal Reconstruction Using Recurrent Neural Network. In *Computing in Cardiology*. IEEE, pp. 445–447.

Singh, S. & Gupta, P.R. (2011). Breast Cancer detection and Classification using Neural Network. *International Journal of Advanced Engineering Sciences and Technologies*, 6(1), pp.4–9.

Siwek, K. & Osowski, S. (2001). Regularization of neural networks for improved load forecasting in power system. *ICECS 2001. 8th IEEE International Conference on Electronics, Circuits and Systems (Cat. No.01EX483)*, pp.1255–1258. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=957443 [Accessed May 2012].

Skowronski, M.D. & Harris, J.G. (2006). Minimum mean squared error time series classi fi cation using an echo state network prediction model. In *in the IEEE International Symposium on Circuits and Systems*. Island of Kos, Greece, pp. 3153–3156.

Somers, M.J. (2001). Thinking differently: assessing nonlinearities in the relationship between work attitudes and job performance using a Bayesian neural network. *Journal of Occupational and Organizational Psychology*, 74(1), pp.47–61.

Somers, M.J. & Casal, J.C. (2009). Using Artificial Neural Networks to Model Nonlinearity. *Organizational Research Methods*, 12(3), pp.403–417.

Steurer, E. (1993). Nonlinear modelling of the DEM/USD exchange rate. In Refenes, A.P. ed., *Neural Networks in the Capital Markets*. New York: Wiley, pp. 199–211.

Stornetta, S.W., Hogg, T. & Huberman, B.A. (1987). A Dynamical Approach to Temporal Pattern Processing. In *NIPS*. American Institute of Physics, pp. 750–759.

Strickert, M. & Hammer, B. (2005). Merge SOM for temporal data. *Neurocomputing*, 64, pp.39–71. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0925231204005107 [Accessed January 23, 2014].

Subasia, A., Kiymik, M.K., Alkana, A. & Koklukaya, E. (2005). Neural Network Classification of EEG Signals by Using AR. *Mathematical and Computational Applications*, 10(1), pp.57–70.

Sundareshan, M.K., Wong, Y.C. & Condarcure, T. (1999). Training Algorithms for Rrecurrent Neural Nets That Eliminate The Need for Compution of Error Gradients With Application to Trajectory Production Problem. In Medsker, L.R. & Jain, L.C., ed. *Recurrent Neural Networks: Design and Applications*. Florida: CRC Press.

Sutskever, I., Martens, J. & Hinton, G.E. (2011). Generating Text with Recurrent Neural Networks. In Getoor, L. and Scheffer, T., eds. *Proceedings of the 28 th International Conference on Machine Learning*. Bellevue, WA, USA: Omnipress, pp. 1017–1024.

Swarup, S., Lakkaraju, K., Smith, N., and Ray, S. (2005). A Recurrent Anticipatory Self-Organized Map for Robust Time Series Prediction. In The Workshop on Self-Organizing Maps (WSOM), Paris, France, Sep 5-8

Swingler, K. (1996). *Applying neural networks: a practical guide*, San Francisco, CA: Morgan Kaufmann Publishers In; Pap/Dsk edition.

Szkoła, J., Pancerz, K. & Warchoł, J. (2011). Recurrent neural networks in computer-based clinical decision support for laryngopathies: an experimental study. *Computational intelligence and neuroscience*, pp.289-398.

Tan, T.Z., Quek, C. & Ng, G.S. (2005). Brain-inspired Genetic Complementary Learning for Stock Market Prediction. *In IEEE congress on evolutionary computation*, 3, pp.2653–2660.

Tawfik, H., Huang, R., Samy, M. & Nagar, A.K. (2009). On the Use of Intelligent Systems for the Modelling of Financial Literacy Parameters. *Journal of Information Technology Research (JITR).* 2 (4), pp. 17-35.

Tellez, P. (2013). Number of Hidden Neurons in A Recurrent Neural Network Estimation. *International Journal of Scientific & Engineering Research, (IJSER)*, (January), pp.0–17.

Tenti, P. (1996). Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*, 10, pp.567-581.

Thomason, M. (1999). The practitioner method and tools. *Journal of Computational Intelligence in Finance*, 7(3), pp.36–45.

Timmis, J. (2001). *Artificial immune systems: a novel data analysis technique inspired by the immune network theory*. PhD thesis. University of Wales.

Tino, P., Farkas, I. & Mourik, J. van, 2005. Recursive Self-Organizing Map as a Contractive Iterative Function System. In *Intelligent Data Engineering and Automated Learning*. Springer, pp. 327–334.

Tsoi, A C. & Back, a D. (1994). Locally recurrent globally feedforward networks: a critical review of architectures. *IEEE Transactions on Neural Networks / a publication of the IEEE Neural Networks Council*, 5(2), pp.229–39. Available at: http://www.ncbi.nlm.nih.gov/pubmed/18267793 [Accessed May 2013].

Tsuji, T., Fukuda, O., Kaneko, M. & Ito, K. (2000). Pattern classification of time-series EMG signals using neural networks. *International Journal of Adaptive Control and Signal Processing*, 14 (February), pp.829–848.

Übeyli, E.D. (2009). Analysis of EEG signals by implementing eigenvector methods/recurrent neural networks. *Digital Signal Processing*, 19(1), pp.134–143. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1051200408001243 [Accessed April 16, 2013].

Übeyli, E.D. & Guler, I. (2007). Statistics over Lyapunov Exponents for Feature Extraction : Electroencephalographic Changes Detection Case. *World Academy of Science, Engineering and Technology*, pp.625–628.

Übeyli, E.D. & Übeyli, M. (2008). Case Studies for Applications of Elman Recurrent Neural Networks. In X. H. and P. Balasubramaniam, ed. *Recurrent Neural Networks*. Vienna, Austria: I-Tech, p. 440.

Varsta, M., Heikkonen, J. & Lampinen, J. (2001). Temporal Kohonen Map and the Recurrent Self-Organizing Map : Analytical and Experimental Comparison. *Neural Processing Letters*, pp.237–251.

Vecer, J. & York, N. (2006). Maximum Drawdown and Directional Trading. Risk , 9(12) , pp.1–8.

Verdenik, I., Pajntar, M. & Leskosek, B. (2001). Uterine electrical activity as predictor of preterm birth in women with preterm contractions. *Eur J Obstet Gynecol Reprod Biol, 95(2), pp.149–53*.

Verplancke, T., Looy,S., Steurbaut, K., Benoit, D., Turck, F.D., Moor, G.D. & Decruyenaere, J. (2010). A novel time series analysis approach for prediction of dialysis in critically ill patients using echo-state networks. *BMC medical informatics and decision making*, 10, p.4.

Voegtlin, T. (2002). Recursive self-organizing maps. *Neural networks: The Official Journal of the International Neural Network Society*, 15(8-9), pp.979–91.

Vojinovic, Z., Kecman, V. & Seidel, R. (2001). A Data Mining Approach to Financial Time Series Modelling and Forecasting. *International Journal of Intelligent Systems in Accounting, Fiance & Management*, 239, pp.225–239.

Vrhovec, J. (2009). Evaluating the progress of the labour with sample entropy calculated from the uterine EMG activity. *Elektrotehniški vestnik*, 76(4), pp.165–170.

Walczak, S. (2001). *An Empirical Analysis of Data Requirements for Financial Forecasting* with Neural Networks. *Journal of Management Information Systems*, 17(4), pp.203–222.

Wang, K., Lee, C.-H. & Juang, B. (1997). Selective Feature Extraction via Signal Decomposition. *IEEE Signal Processing Letters*, 4(1), pp.8–11.

Wang, X., Ma, L., Wang, B. & Wang, T. (2013). A hybrid optimization-based recurrent neural network for real-time data prediction. *Neurocomputing*, 120, pp.547–559. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0925231213004876 [Accessed January 23, 2014].

WebMd (2012). *Premature Labor*. Available at: http://www.webmd.com/baby/premature-labor.

Weijters, A., Weijters, A., Bosch, A.V. & Herik, H.J. (1997). Behavioural Aspects of Combining Backpropagation Learning and Self- organizing Maps. *Connection Science*, 9(3), pp.235–251.

WelcomeBabyHome, (2006). *The Birth Process.* Available at: http://www.welcomebabyhome.com/pregnancy/pregnancy_birth_process.htm [Accessed January 11, 2013].

Werbos, P.J. (1988). Generalization of Backpropagation with Application to a Recurrent Gas Market Model. *Neural Networks*, 1(4), pp.339–356. Available at: http://www.sciencedirect.com/science/article/pii/089360808890007X#?np=y [[Accessed Jun 2012].

Wesley-smith, I. (2006). A Parallel Artificial Neural Network Implementation. In *Proceedings of The National Conference On Undergraduate Research (NCUR)*. Asheville, North Carolina.

West, B.J., 1985. *Lecture notes in biomathematics: An essay on the importance of being nonlinear*. Heidelberg, Berlin: Springer-Verlag.

WHO (2012). *Born too soon: The Global Action Report on Preterm Birth*, Available at: http://www.who.int/pmnch/media/news/2012/preterm_birth_report/en/ [Accessed May 2013].

Widrow, B., Rumelhart, D.E. & Lehr, M.A. (1994). Networks :Neural Networks: Applications in Industry, Business and Science. *Communication of the ACM*, 37(3), pp.93–105.

Widyanto, M.R., Kusumoputro, B., Nobuhara, H. , Kawamoto, K. & Hirota, K. (2006). A Fuzzy-Similarity-Based Self-Organized Network Inspired by Immune Algorithm for Three-Mixture-Fragrance Recognition. *IEEE Transactions on Industrial Electronics*, 53(1), pp.313–321.

Widyanto, M.R., Nobuhara, H., Kawamoto, K., Hirota, K. & Kusumoputro, B. (2005). Improving recognition and generalization capability of back-propagation NN using a self-organized network inspired by immune algorithm (SONIA). *Applied Soft Computing*, 6(1), pp.72–84.

Maner, W. & Garfield, R. (2007). Identification of human term and preterm labor using artificial neural networks on uterine electromyography data. *Ann Biomed Eng*, 35(3), pp.465–473.

Williams, R.J. & Zipser, D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2), pp.270–280.

Wold, H. & Whittle, P. (1956). A Study in the Analysis of Stationary Time Series. *Economic Journal*, 66(262), pp.327–330.

Yao, J., Poh, H. & Jasic, T. (1996). Foreign Exchange Rates Forecasting with Neural Networks. National University of Singapore Working Paper, In *International Conference on Neural Information Processing*. Hong Kong.

Yao, J. & Tan, C.L. (2000). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, 34(1-4), pp.79–98. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0925231200003003 [Accessed March 2012].

Yao, K., Zweig, G , Hwang, M-Y , Shi, Y , Yu, D. (2013). Recurrent Neural Networks for Language Understanding. *Interspeech*, (1), pp.2–6.

Yümlü, S., Gürgen, F.S. & Okay, N. (2005). A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction. *Pattern Recognition Letters*, 26(13), pp.2093–2103. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167865505000875 [Accessed June 18, 2013].

Zekic, M. (1998). Neural network applications in stock market predictions: A methodology analysis. In B. A. and R. V. Logozar, ed. *Proceedings of the 9th International Conference on Information and Intelligent Systems*. pp. 255–263.

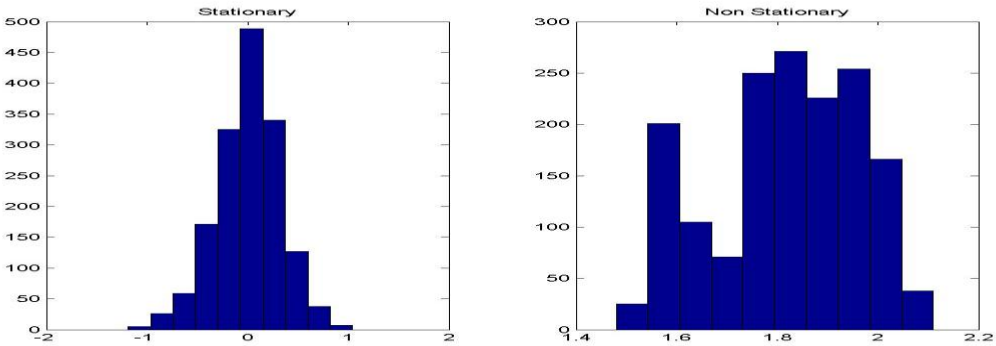
Zhang, G.P. (2000). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 30(4), pp.451–462. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=897072 [Accessed July 2011]

Zhang, G.P., Patuwo, B.E. & Hu, M.Y. (2001). A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computer and Operation research* , 28. p.381-396.

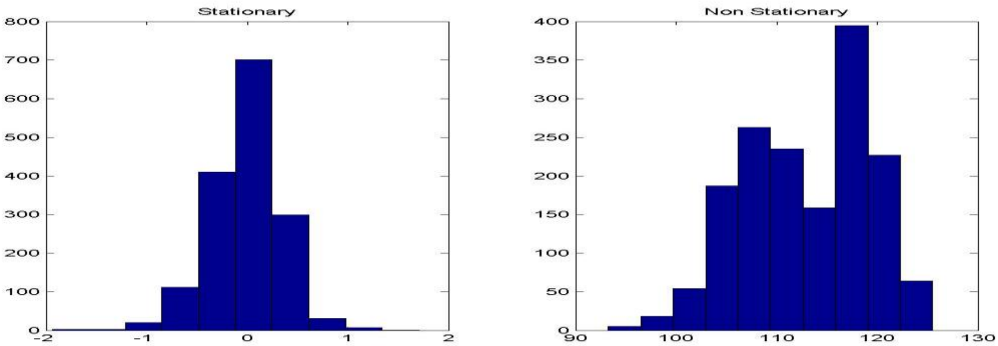
Zhang, M. (2008). *Artificial Higher Order Neural Networks for Economics and Business* M. Zhang, ed., IGI Global. Available at: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-897-0[Accessed March 2012].

Zhang, X., Liu, Y., Yang, M., Zhang, T. & Young, A.A. (2013). Comparative study of four time series methods in forecasting typhoid fever incidence in China. *PloS one*, 8(5), p.e63116. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3641111&tool=pmcentrez&rendertype=abstract [Accessed November 1, 2013].

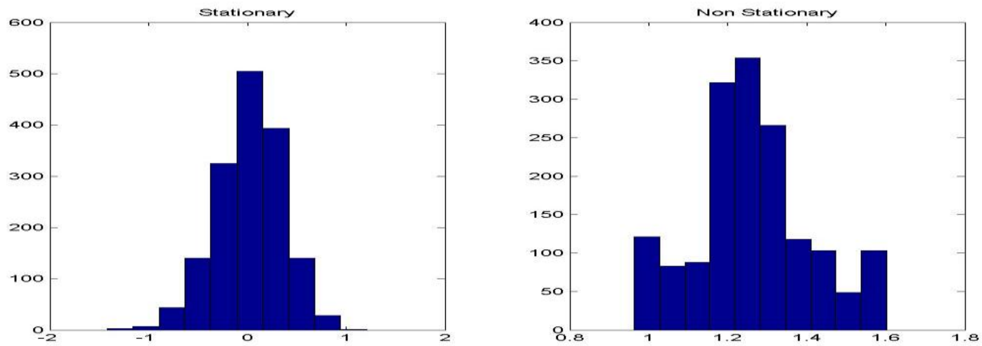
APPENDIX



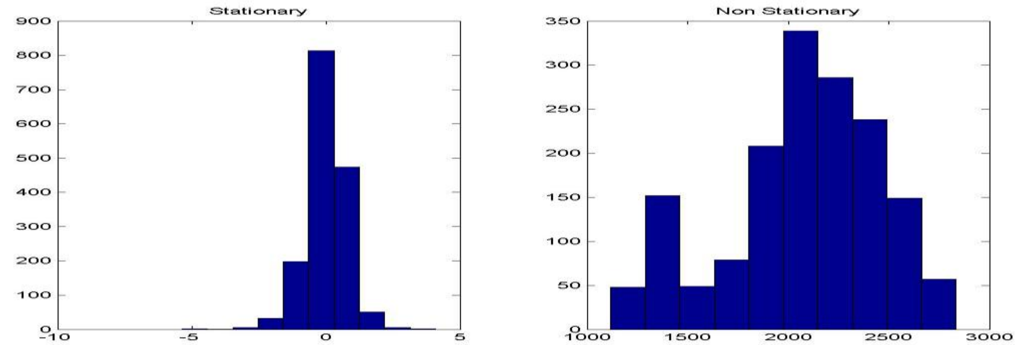
Appendix 1: Histogram for the USDUKP signals after and before pre-processing



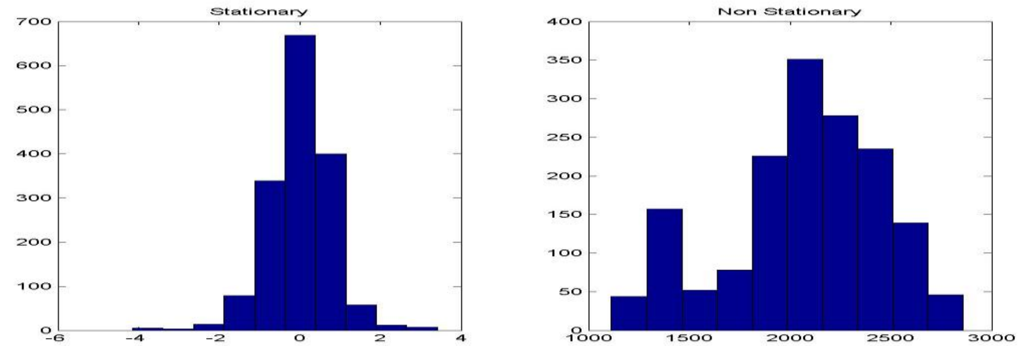
Appendix 2: Histogram for the JPYUSD signals before and after pre-processing



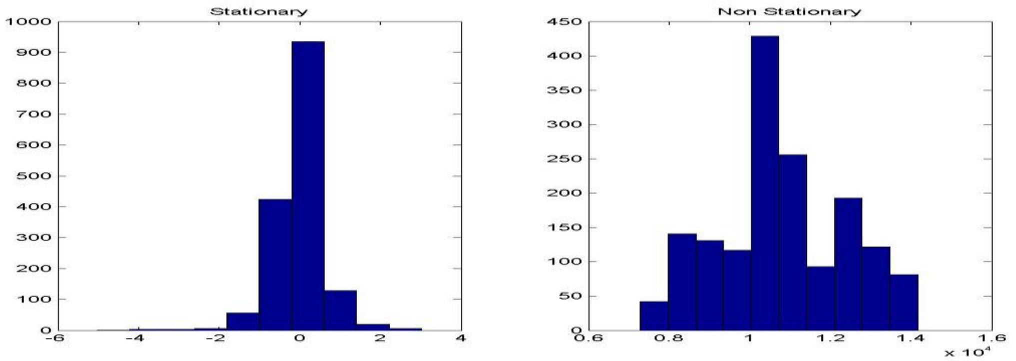
Appendix 3: Histogram for the USDEUR signals after and before pre-processing



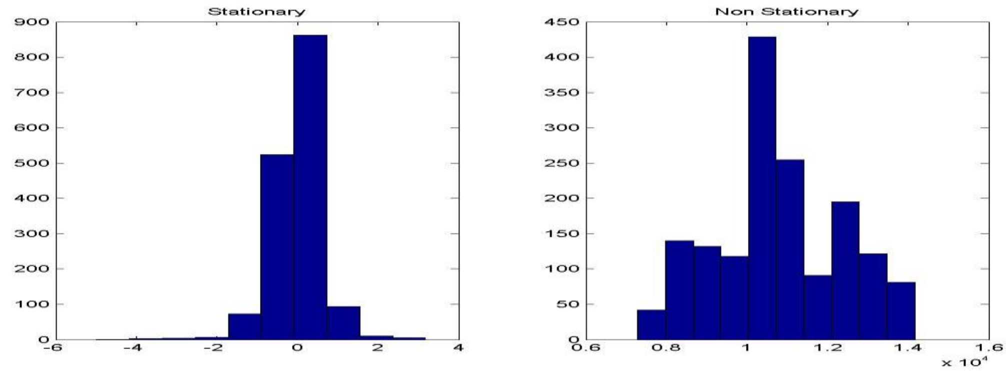
Appendix 4: Histogram for the NASDAQO signals after and before pre-processing



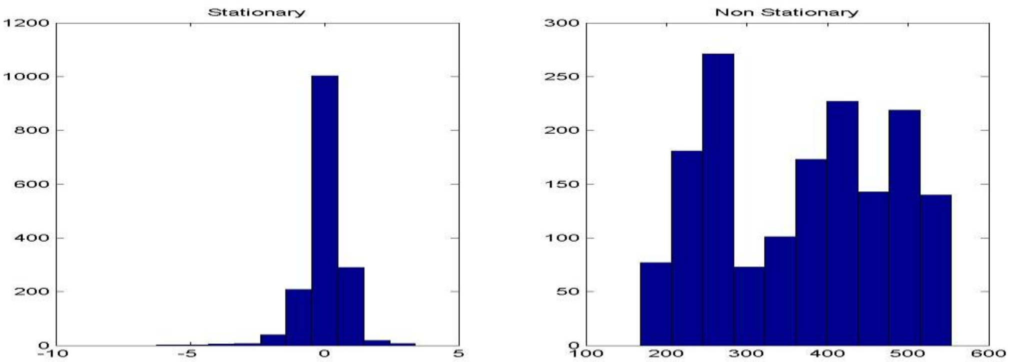
Appendix 5: Histogram for the NASDAQC signals after and before pre-processing



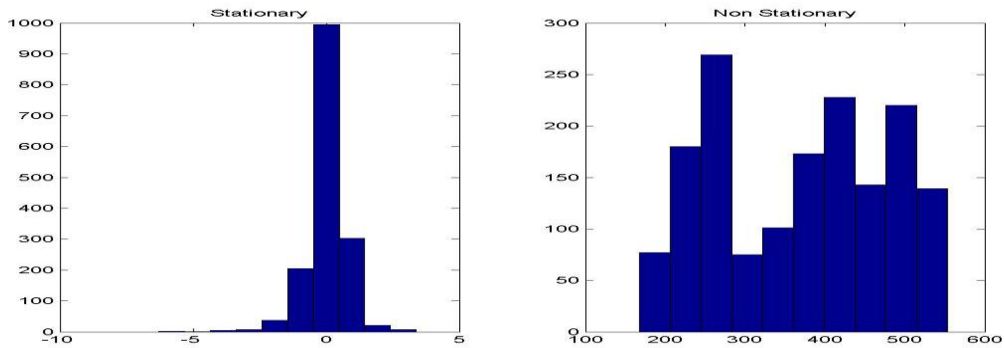
Appendix 6: Histogram for the DJIAO signals after and before pre-processing



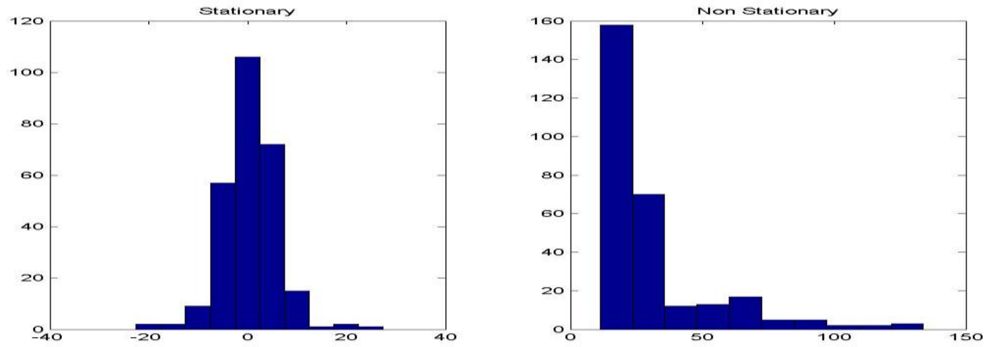
Appendix 7: Histogram for the DJIAC signals after and before pre-processing



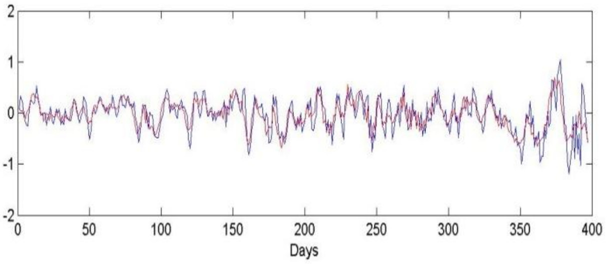
Appendix 8: Histogram for the DJUAO signals after and before pre-processing



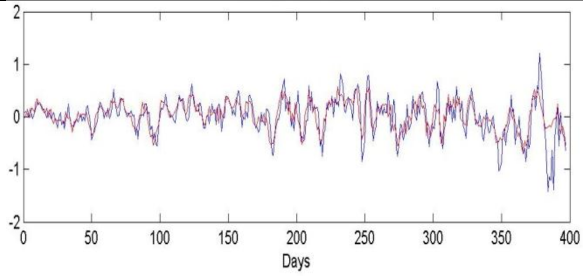
Appendix 9: Histogram for the DJUAC signals after and before pre-processing



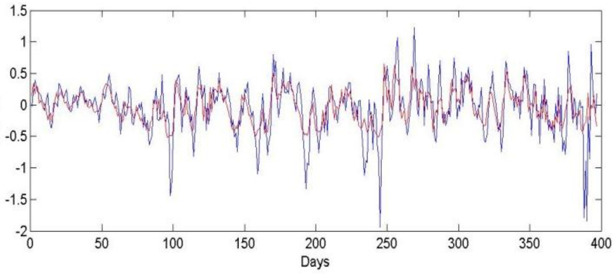
Appendix 10: Histogram for the OIL signals after and before pre-processing



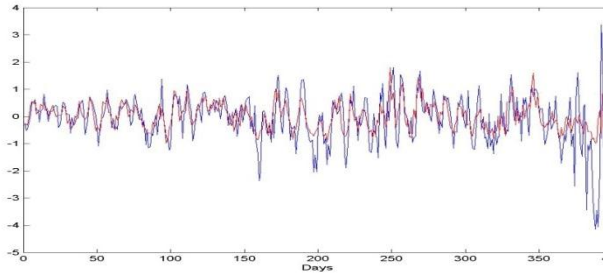
USDUKP



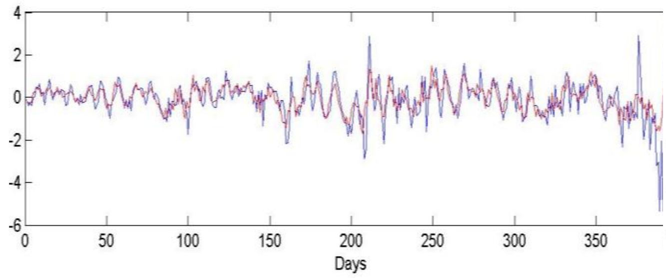
USDEUR



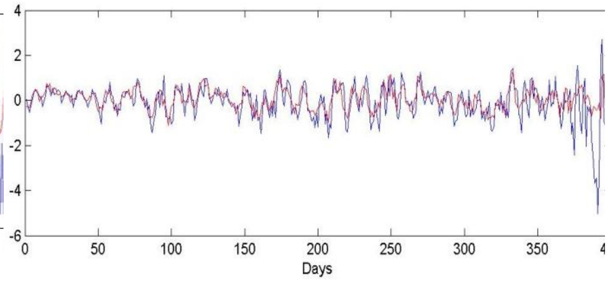
JPYUSD



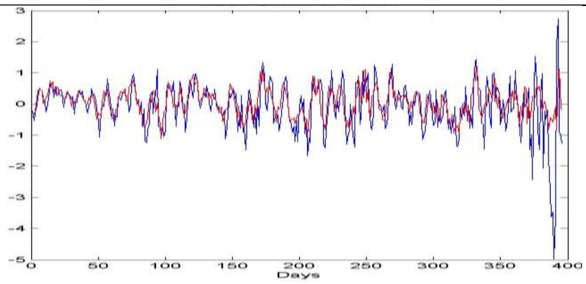
NASDAQC



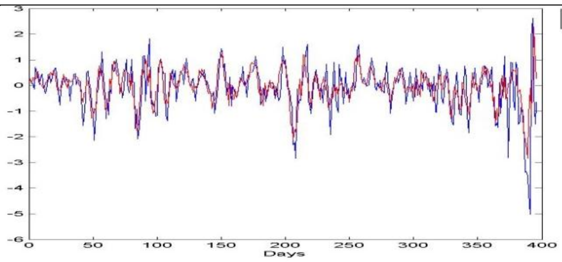
NASDAQO



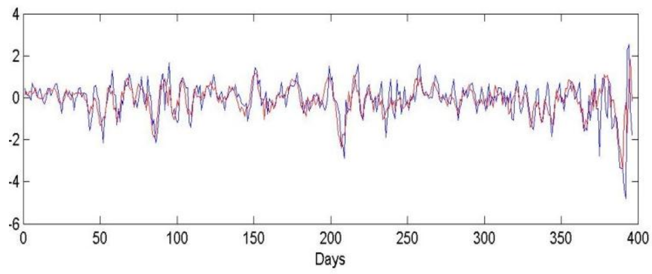
DJIAO



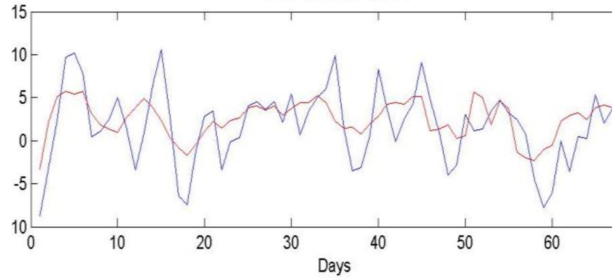
DJIAC



DJUAC

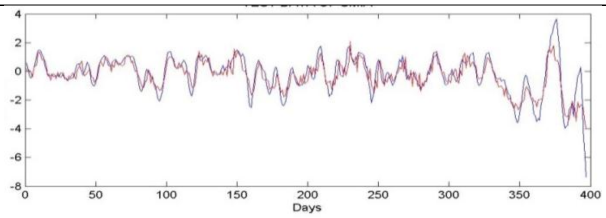


DJUAO

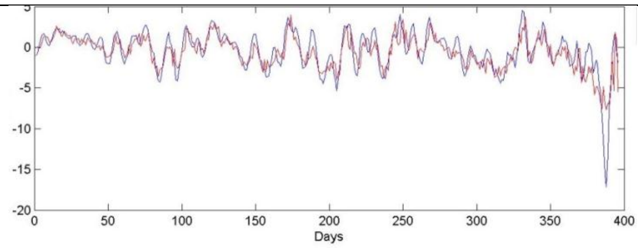


OIL

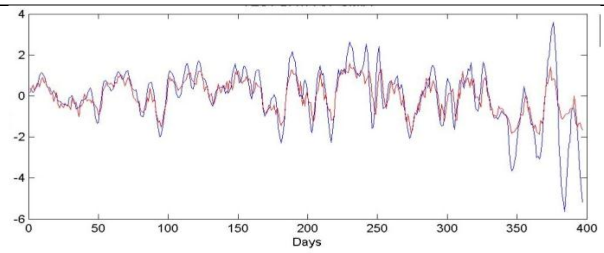
Appendix 11:The best forecasting on stationary data for the prediction of one step ahead



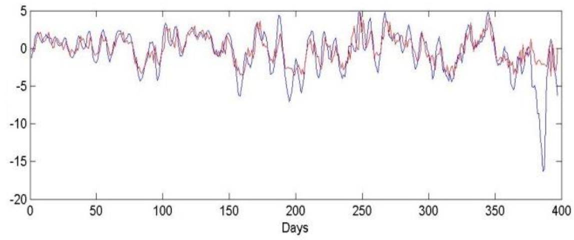
USDUKP



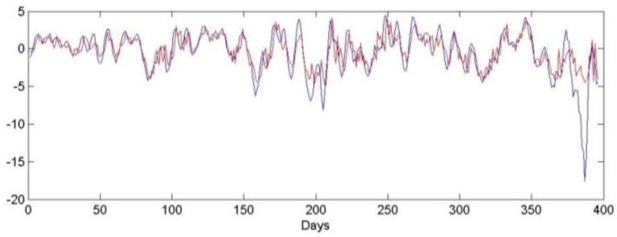
USDEUR



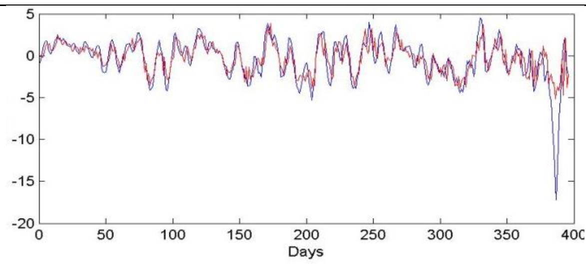
JPYUSD



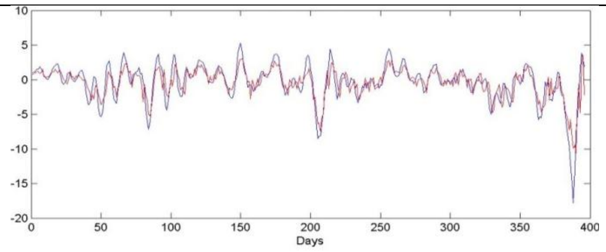
NASDAQC



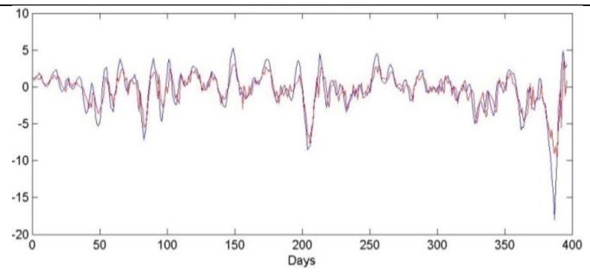
NASDAQO



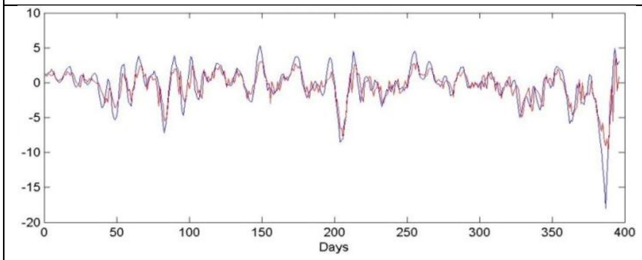
DJIAO



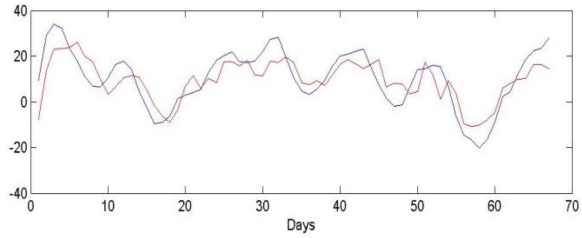
DJIAC



DJUAC

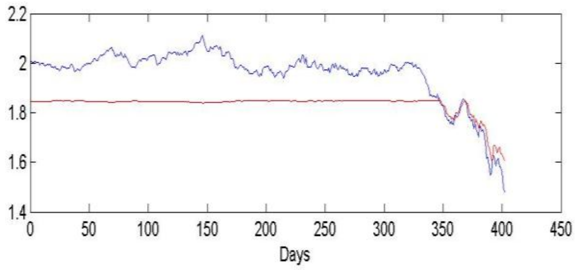


DJUAO

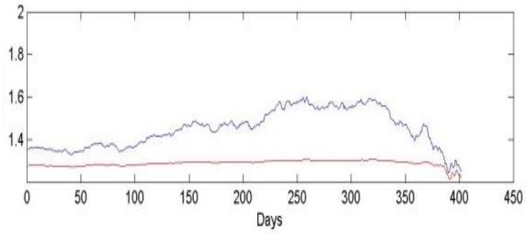


OIL

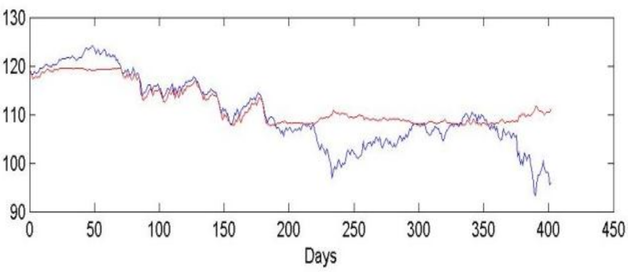
Appendix 12:The best forecasting on stationary data for the prediction of five step ahead



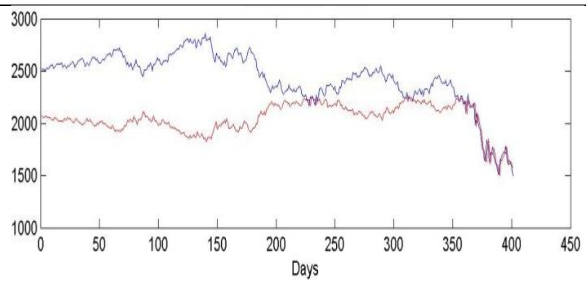
USDUKP



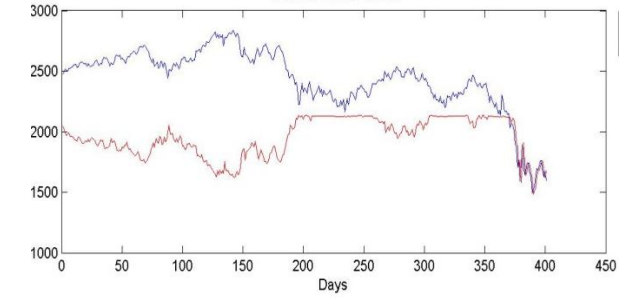
USDEUR



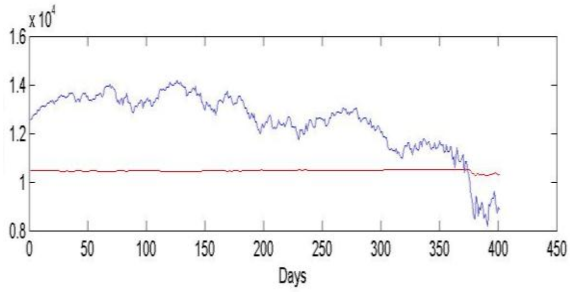
JPYUSD



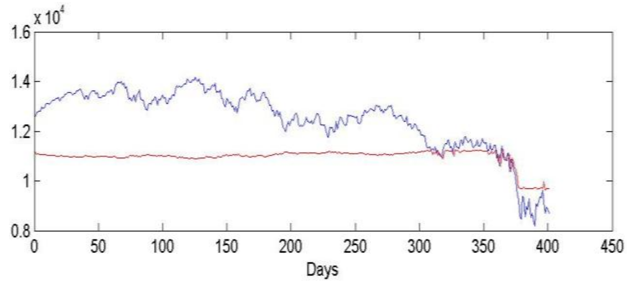
NASDAQC



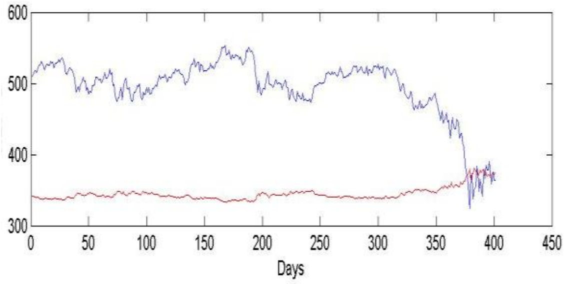
NASDAQO



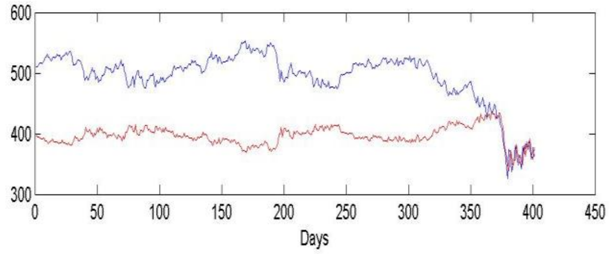
DJIAO



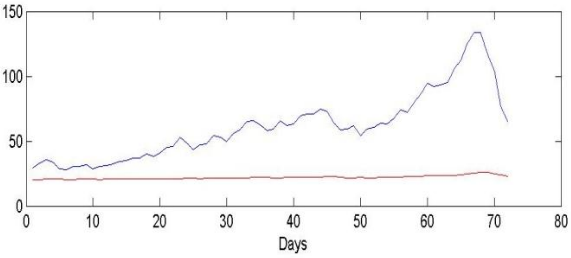
DJIAC



DJUAC

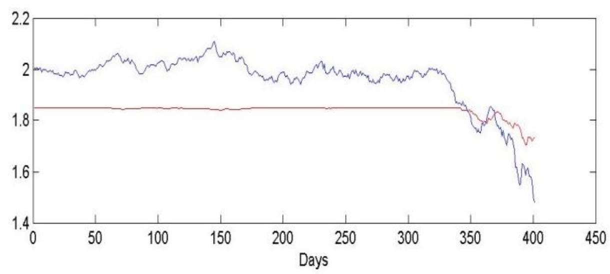


DJUAO

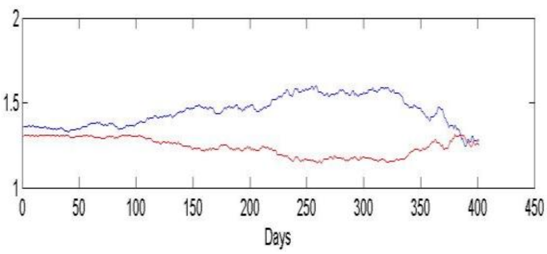


OIL

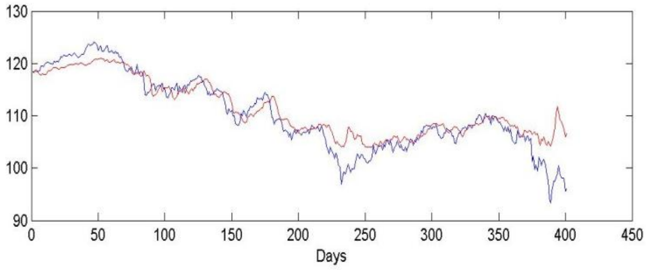
Appendix 12: The best forecasting on nonstationary data for the prediction of one step ahead



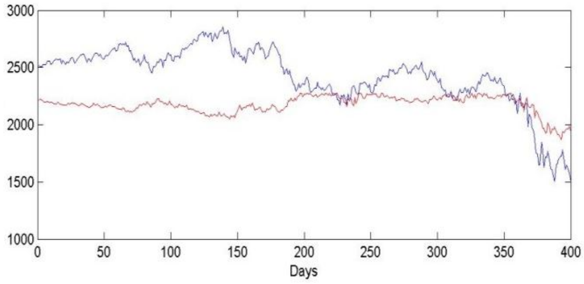
USDUKP



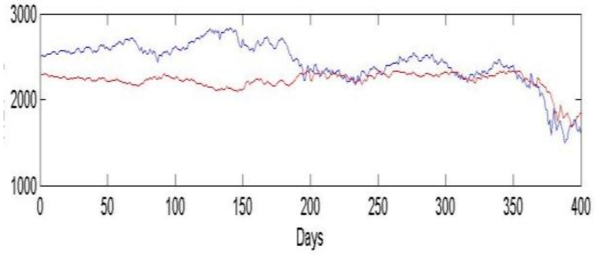
USDEUR



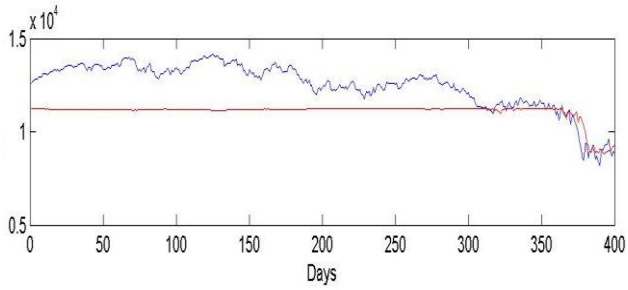
JPYUSD



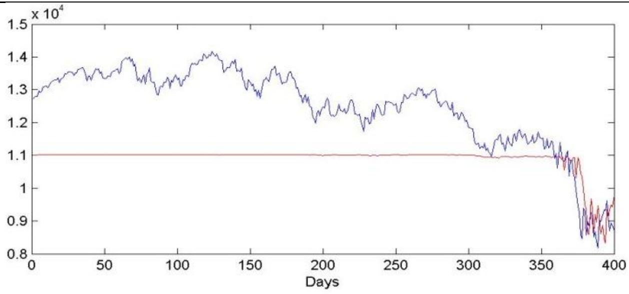
NASDAQC



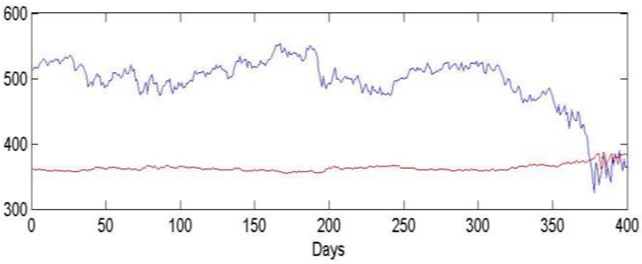
NASDAQO



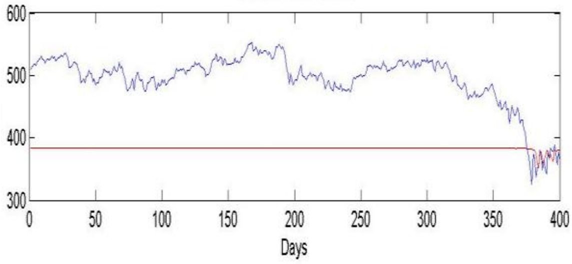
DJIAO



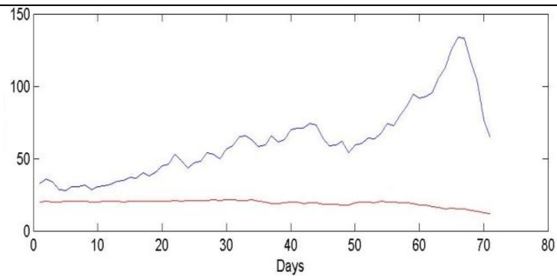
DJIAC



DJUAC

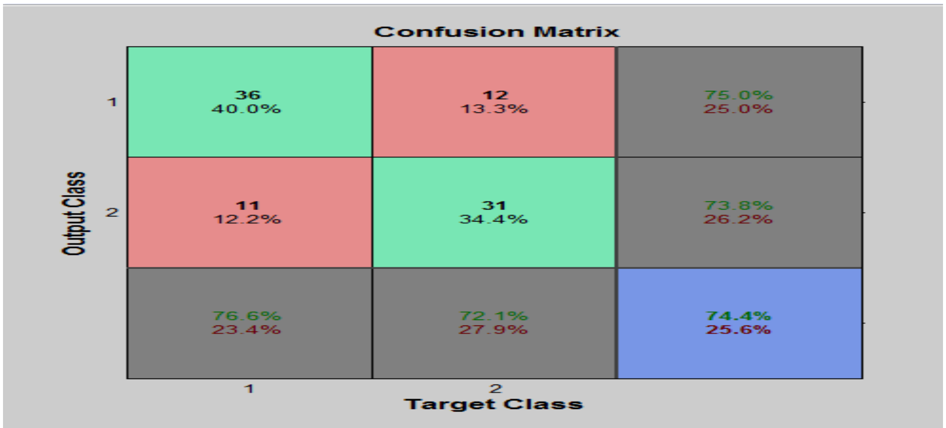


DJUAO

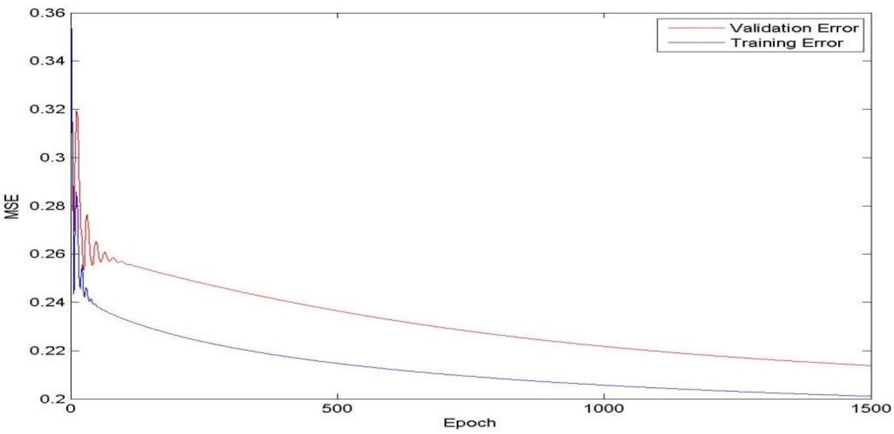


OIL

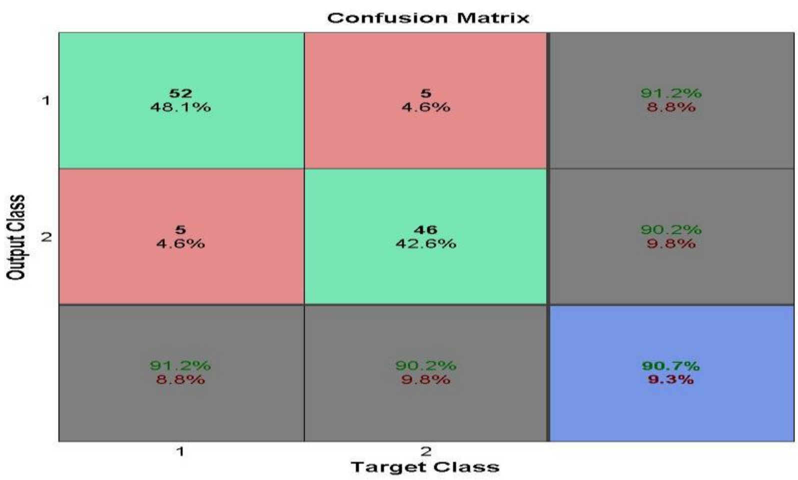
Appendix 13: The best forecasting on nonstationary data for the prediction of five step ahead



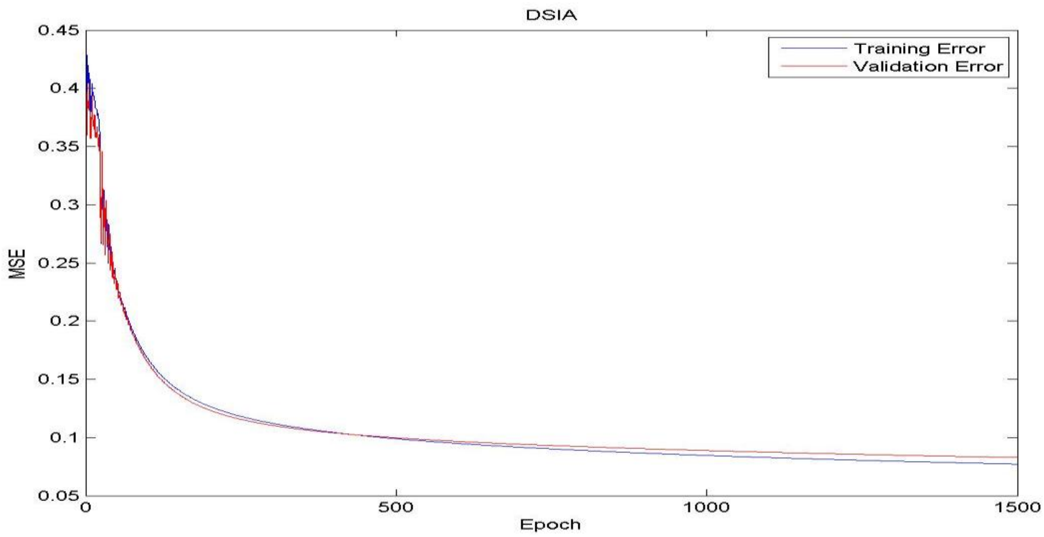
Appendix 14: The confusion Matrix for DSIA classifier for four features



Appendix 15: The MSE value for training and validation set for DSIA classifier



Appendix 16: The confusion Matrix for DSIA classifier for 15 features



Appendix 16: The MSE value for training and validation set for DSIA classifier