The Application of Dynamic Self-organised Multilayer network Inspired by the Immune Algorithm for Weather Signals Forecast

Abir Jaafar Hussain
Applied Computing Research Group,
Liverpool John Moores University
Byrom Street, Liverpool,
L3 3AF, UK
a.hussain@ljmu.ac.uk

Dhiya Al-Jumeily
Applied Computing Research Group,
Liverpool John Moores University
Byrom Street, Liverpool,
L3 3AF, UK
d.aljumeily@ljmu.ac.uk

Haya Al-Askar
Salman Bin Abdulaziz University,
Department of Computer Science,
Al-Kharj - PO Box 151 - ZIP code
11942, Saudi Arabia

***Abstract*—Neural network architecture called Dynamic Self-organised Multilayer Network Inspired by the Immune Algorithm is proposed for the prediction of weather signals. Two sets of experiments have been implemented. The simulation results showed slight improvement achieved by the proposed network when using the average results of 30 simulations. For the second set of experiments, the simulation results indicated that there is no significant improvement over the first set of experiments.**

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, for example, medical diagnostics and pattern recognition for big data. The structure of the proposed network can be modified for clustering tasks by changing the back-propagation algorithm in the output layer. This can extend the application of the proposed network to scientifically analyse different types of big data.

Keywords*—Artificial Immune Systems, Time Series Data, weather data.

I. Introduction

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. [1-6].

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 [7] assumed that time series are generated by a dynamic system. 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, weather signals 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.

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.

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 [8]. In the literature, ANNs are also known as neurocomputer, connectionist network, and parallel distributed processor. 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. [9]. 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 multilayer neural network (MLP). It has been applied extensively in time series prediction. Although there are massive applications of the well-known MLP neural networks, they suffer from difficulties such as the

ISBN: 978-1-4799-5679-1 ©2015 IEEE

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 [10-12]. 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. [12]. They designed a self-organised hidden layer inspired by 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 originally to predict temperature-based food quality; it showed an 18% improvement in correct recognition in comparison to the MLP network [12].

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 [13]. 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 [13-14]. These connections can offer the cognitive function such as memory association, classification or predication of dynamic system. The work of Makarov et al. [14] showed that recurrent networks could 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 a dynamic neural network that can deal with complex time series analysis problems, and this will be achieved by designing a novel dynamic self-organised neural network which is inspired by immune algorithm and the self organized neural network proposed by Widyanto et al. [12]. 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 weather time series signals.

The reminder of this paper is organised as follows. Section II discusses the self-organized neural network inspired by the immune algorithm (SONIA). Section III illustrates the proposed dynamic self organize multilayer neural network inspired by the immune algorithm (DSMIA), while section IV shows time series forecasting. The methodology for the experiments in this paper is illustrated in section V while section VI shows the simulation results. Finally, section VII is dedicated for the conclusion and future directions.

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

The SONIA network which is introduced by Widyanto et al. in 2005 [12] 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 antigen. 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 [15]. 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.

III. The Dynamic Self-organised Multilayer network Inspired by the Immune Algorithm (DSMIA)

In this section, the dynamic self-organised network inspired by the immune algorithm (DSMIA) based on the Jordan recurrent neural architecture [16] 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.

a) 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. [17], 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 [17]. 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 [18]. The structure of this connection is as stated by Hammer, et al. [19]: "The output computed for one time step depends on the current constituent of the structure and the internal model state obtained by previous calculations, i.e., the output of the neurons computed in recursively addressed previous steps". In the proposed network, the structure of the recurrent connection is the same architecture as the Jordan network [16], 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 1. 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.

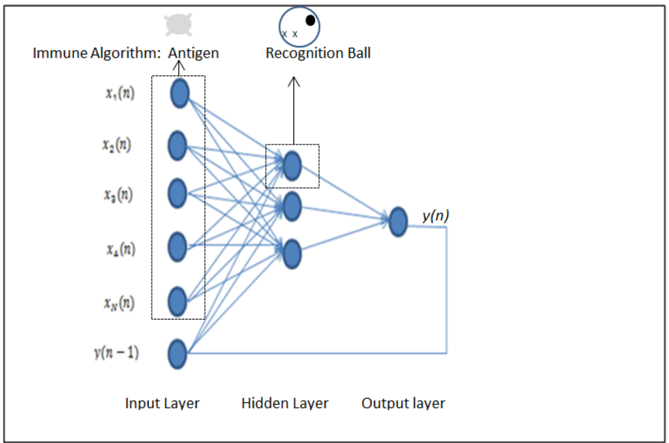


Figure 1. The structure of the proposed DSMIA network

b) 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 $x(t)$ as well as $y_k(t-1)$ and the number of inputs of the network is N_i+N_o defined as U where the output of the input layer:

$$\mathrm{U(n)} = \begin{cases} x_i(n) & i = 1,....N \\ y_i(n-1) & i = 1,....,O \end{cases}$$

The output of the hidden layer is computed as:

$$v_{hj}(n) = \alpha\sqrt{\sum_{i=1}^{N}(w_{hji} - x_i(n))^2}$$

$$z_{hj}(n) = \beta\sqrt{\sum_{k=1}^{O}(wz_{hjk} - y_k(n-1))^2}$$

$$D_{hj}(n) = v_{hj}(n) + z_{hj}(n)$$

$$x_{hj}(n) = f_{ht}(D_{hj}(n))$$

Where f_{ht} is a nonlinear activation function, N_i is the number of external inputs, N_o is the number of output units, w_{hji} is the weight corresponding to the external input while W_{ojk} 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 $0 < \beta$.

$$y_k(t) = f_{ot}\left(\sum_{j=1}^{N_h} x_{hj}(t) \cdot w_{ojk}\right)$$

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

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_{ojk} are also updated in the same way as the weights of the external inputs W_{hji}. This is done by first finding D, which is the distance between the input units and the centroid of the j^{th} hidden units:

$$D_{hj}(t) = \alpha\sqrt{\sum_{i=1}^{N_i}(X_i(t) - W_{hji}(t))^2} + \beta\sqrt{\sum_{k=1}^{N_o}(y_k(t-1) - W_{ojk}(t))^2}$$

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

$$c(t) = \arg\min\{D_{hj}(t)\}$$

Where $c(t)$ minimised $D_{hj}(t)$, 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:

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

$$W_{ojk}(t+1) = W_{ojk}(t) + \gamma D_c(t)$$

Where W_{ojk} is the weight of the previous output and W_{hji} is the weight for the external inputs, and γ is the learning rate that is updated during the epochs.

IV. TIME SERIES FORECASTING

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. Mathematically, for any given system, a time series can be referred to as $x(t)$ or $\{x(t), t$ □ $T\}$, 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 [20-21]. 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 [7] 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. For example, financial market prediction by computations of the next value of trade sales each month [22-24].

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. Therefore, time variables are very important in time series analysis because they show when the measurements were recorded. Hence, Herrera [7] 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 [25], such as climate oscillations [26].

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 [27]. 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. Therefore, this type of time series is very 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. 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 or biomedical signals. 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.

V. METHODOLOGIES

a) *Time Series Data Used in this Research*

Three noisy time series are utilized in our experiments representing 400 data points for weather information from the Valley weather station in Anglesey (North Wales). They are from November 1980 until February 2014 (per month) and represent the maximum temperature, rainfall in mm and sunshine in hours.

b) *Data Pre-processing*

The selection of a suitable forecasting horizon is the first step that must be taken before weather time series forecasting can begin. If the forecasting horizon is very long, that might increase the complexity of the forecasting procedure. This experiment applies five days ahead predictions

The weather 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. 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 using the following equation:

$$R(t) = \frac{S(t)}{S(t-1)} - 1$$

where S(t) is the input signal and R(t) is the one step relative increase in price at time t. This transformation has been shown to achieve better results [28]. R(t) has a relative constant range of values even if the input data represents many days values, while the original data S(t) vary so much which make it very difficult to use a valid model for a long period of time [29]. Another advantage of using this transformation is that the distribution of the transformed data will become more symmetrical and will follow more closely to a normal distribution.

c) *Performance measures*

The prediction performance of our networks was evaluated using four statistical metrics which are used to provide accurate tracking of the signals as shown in Table 1.

TABLE I
PERFORMANCE METRICS AN THEIR CALCULATIONS

Metrics	Calculations
NMSE	$NMSE = \frac{1}{\sigma^2 * N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$ $\sigma^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \overline{y}_i)$
MSE	$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$
MAE	$MAE = \frac{1}{N} \sum_{i=1}^{N} \mid y_i - \hat{y}_i \mid^2$
SNR	$SNR = 10 * \log_{10}(sigma)$ $sigma = \frac{m^2 * n}{SSE}$ $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)$ $m = \max(y)$

n is the total number of data patterns
y and $\hat{y}$ represent the actual and predicted output value

VI. Simulation results

In this section, the simulation results for the prediction of the four weather time series will be presented using various neural networks. The proposed network is benchmarked with the MLP, Elman, Jordan and the SONIA networks.
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.

Two sets of experiments were performed, in the first set of experiments the weather signals are passed directly to the network as nonstationary data while in the second set of experiments the signals are transformed into stationary signals as shown previously.

The number of input units must be selected carefully. 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. Using too many past periods as input will lead to much difficulty in training the Artificial Neural Network (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 [4, 10].

TABLE II
THE PREDICTION OF THE SUNSHINE AT THE ANGLESEY VALLEY FOR FIVE STEPS PREDICTIONS USING NONSTATIONARY SIGNALS

Networks	NMSE	MSE	MAE	SNR
MLP	0.8085	0.0153	0.0889	16.2208
Elman	0.6770	0.0128	0.0814	18.2988
Jordan	0.3410	0.0620	0.0064	19.9732
SONIA	0.274287	0.002432	0.039016	23.53
DSMIA	0.5478	0.0824	0.0104	17.9119

TABLE III
THE PREDICTION OF THE TEMPERTURE AT THE ANGLESEY VALLEY FOR FIVE STEPS PREDICTIONS USING NONSTATIONARY SIGNALS

Networks	NMSE	MSE	MAE	SNR
MLP	0.3877	0.0061	0.0626	19.7504
Elman	1.0950	0.0172	0.0945	17.5980
Jordan	0.1663	0.0026	0.0396	23.4705
SONIA	0.578262	0.010928	0.085222	17.68
DSMIA	0.1481	0.0392	0.0023	23.9300

TABLE IV
THE PREDICTION OF THE RAINFALL AT THE ANGLESEY VALLEY FOR FIVE STEPS PREDICTIONS USING NONSTATIONARY SIGNALS

Networks	NMSE	MSE	MAE	SNR
MLP	0.9289	0.0107	0.0808	17.3117
Elman	1.1137	0.0129	0.0892	16.5567
Jordan	1.8175	0.0210	0.1086	15.8251
SONIA	0.350782	0.005502	0.060503	20.18
DSMIA	1.0692	0.00303	0.0321	21.4520

Tables I, II and III show the average results of 30 simulations obtained on unseen data from the neural networks when the weather signals are passed to the network as non-stationary data for five steps ahead prediction.

TABLE V
THE PREDICTION OF THE SUNSHINE AT THE ANGLESEY VALLEY FOR FIVE STEPS PREDICTIONS USING STATIONARY TRANSFORMATION

Networks	NMSE	MSE	MAE	SNR
MLP	0.91987	0.007105	0.06127	17.65
Elman	1.1306	0.0087	0.0684	16.7721
Jordan	0.8359847	0.00645815	0.058802	18.06
SONIA	0.588017	0.011113	0.085950	17.60
DSMIA	0.900557	0.006956	0.06103	17.74

TABLE VI
THE PREDICTION OF THE TEMPERTURE AT THE ANGLESEY VALLEY FOR FIVE STEPS PREDICTIONS USING STATIONARY TRANSFORMATION

Networks	NMSE	MSE	MAE	SNR
MLP	0.705304	0.003736	0.05018	19.63
Elman	1.388106	0.00735461	0.064681	17.262
Jordan	0.5707405	0.00302395	0.04347525	20.55
SONIA	0.351270	0.005509	0.060488	20.18
DSMIA	0.5777	0.0031	0.0458	20.5021

TABLE VII
THE PREDICTION OF THE RAINFALL AT THE ANGLESEY VALLEY FOR FIVE STEPS PREDICTIONS USING STATIONARY TRANSFORMATION

Networks	NMSE	MSE	MAE	SNR
MLP	1.027654	0.0009102	0.02052	22.75
Elman	1.922730	0.001703	0.0284576	20.51722
Jordan	1.3011231	0.00115243	0.02445824	21.97
SONIA	1.001460	0.000887	0.020383	22.87
DSMIA	0.998607	0.00088	0.02110	22.88

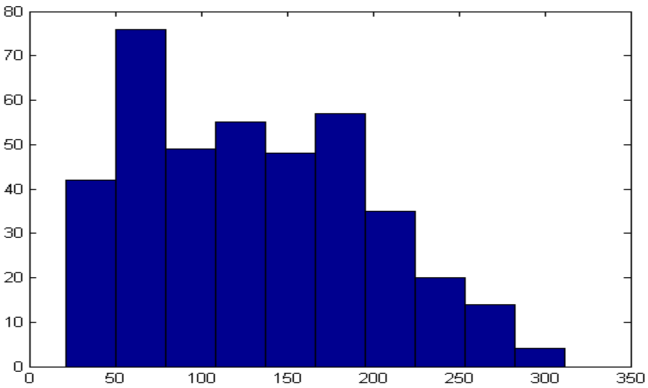
Tables V, VI and VII show the average results of 30 simulations obtained on unseen data from the neural networks when the weather signals are passed to the network as stationary data for five steps ahead prediction.

The simulation results for the stationary and non-stationary prediction indicated that the proposed network showed a slight improvement in comparison with other neural network architectures.

In terms of the signal to noise ratio, the proposed network achieved high values in comparison to the SONIA network except for the prediction of the sunshine signal for non-stationary prediction. In terms of the MAE, the proposed signals achieved better results than the SONIA network except for the prediction of the rainfall signal using stationary prediction.

To further analysis the significant of the results, we have conducted a paired t-test [30] on the best simulation results to determine if there is any significant difference among the proposed DSMIA and the other neural network architectures based on the absolute value of the error. The calculated t-value showed that the proposed technique outperforms the other ANNs with $\alpha = 5\%$ significance level for a one tailed test.

a)



b)

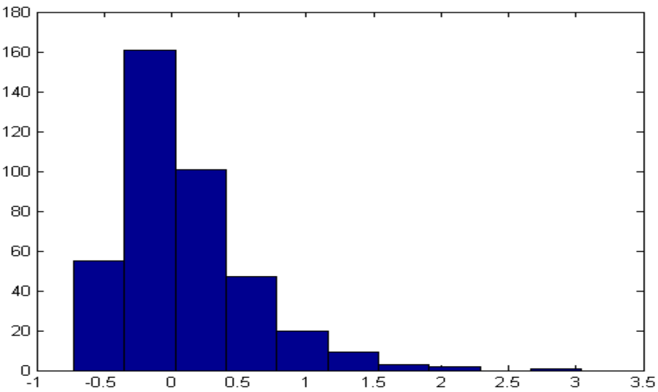


Figure 2. The histogram of a) the original sunshine signal of the Anglesey valley b) the transformed sunshine signal of the Anglesey valley.

Furthermore, the simulation results showed that there is no significant improvement when transforming the data from non-stationary to stationary. To investigate the reasons for these results, the histogram of the transformed data was checked which indicated that the distribution of the transformed data did not become symmetrical and did not follow a normal distribution as shown in Figure 2.

VII. CONCLUSION AND FUTURE WORKS

The dynamic self organized neural network inspired by the immune algorithm is proposed for the prediction of weather signals. Two sets of experiments have been performed, in the first set of experiments the weather signals are passed to the neural networks in which the last 5 values are used to predict the weather in the next 5 days. The simulation results showed slight improvement achieved by the proposed network when using the average results of 30 simulations. For the second set of experiments, the nonstationary weather signals have been transformed to stationary. The simulation results indicated that there is no significant improvement over the first set of experiments.

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.

Since the literature review showed that RNN can be used to reduce noise from biomedical signals and improve the quality of signals, this can broaden the scope of the proposed network to medical applications that can benefit from recurrent neural network models to filter and model the biomedical signals. Another direction of research will investigate the ability of the proposed network to be used as signals pre-processing methods.

REFERENCES

[1]. Mirea, L. & Marcu, T., 2002. System identification using Functional-Link Neural Networks with dynamic structure. In 15th Triennial World Congress. Barcelona, Spain

[2]. 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.

[3]. 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.e77154. Available at: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3810473&tool=pmcentrez&rendertype=abstract [Accessed December 19, 2013].

[4]. 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. 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp.1–4.

[5]. 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].

[6]. Chowdhury, R.H. Reaz, M. B. I. , Ali, M., Bakar, A., Chellappan, K., Chang, T.G.,, 2013a. 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].

[7]. Herrera, J.L., 1999. Time Series Prediction Using Inductive Reasoning Techniques. Instituto de Organizacion y Control de Sistemas Industriales.

[8]. Makeshwar, M.S., Rajgure, N.K., Pund, M., 2010. Object Identification using Neural Network. International Journal of Computer Science and Application , pp.29–32.

[9]. Hertz, J., Palmer, R. & Krogh, A., 1991. Introduction to the Theory of Neural Computation, Redwood City: Addison-Wesley.

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

[11]. 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.

[12]. 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.

[13]. 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.

[14]. 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.

[15]. Shen, X., Gao, X.Z. & Bie, R., 2008. Artificial Immune Networks:Model and Application. International Journal of Computational Intelligence System, 1, pp.168–176.

[16]. 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.

[17]. 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.

[18]. Voegtlin, T., 2002. Recursive self-organizing maps. Neural networks: The Official Journal of the International Neural Network Society, 15(8-9), pp.979–91.

[19]. 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.

[20]. 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.

[21]. Mirea, L. & Marcu, T., 2002. System identification using Functional-Link Neural Networks with dynamic structure. In 15th Triennial World Congress. Barcelona, Spain.

[22]. 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.

[23]. Shachmurove, Y. & Witkowska, D., 2000. Utilizing Artificial Neural Network Model to Predict Stock Markets. 1090 (September), pp.0–25.

[24]. 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].

[25]. Pedersen, M.W., 1997. Optimization of Recurrent Neural Networks for Time Series Modeling. PhD thesis. Technical University of Denmark.

[26]. 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].

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

[28]. Dunis, C. and Williams, M. Modelling and trading the EUR/USD exchange rate: Do neural network models perform better?, Derivatives Use, Trading and Regulation, 8, 3, pp. 211-239.

[29]. Hellstrom, T. and Holmstrom, K. Predicting the Stock Market. Technical Report IMa-TOM-1997-07, Center of Mathematical Modeling, Department of Mathematics and Physis, M· ·alardalen University, V· ·asteras, Sweden, August 1998.

[30]. Montgomery, D. C. and Runger, G. C. Applied Statistics and Probability for Engineers. New York: Wiley, 1999.