

# Recurrent Neural Networks Inspired by Artificial Immune Algorithm for Time Series Prediction

D. Al-Jumeily<sup>1</sup>, A. Hussain<sup>1</sup> and H. Alaskar<sup>2</sup>,

Applied Computing Research Group  
School of Computing and Mathematical Sciences  
Liverpool John Moores University, UK

Email: <sup>1</sup>{D.Aljumeily, A.Hussain}@ljmu.ac.uk, <sup>2</sup>H.Alaskar@2011.ljmu.ac.uk

**Abstract-** This paper presents a novel Dynamic Self-Organised Multilayer Neural Network that can be used for prediction of noisy time series data. The proposed technique is based on the Immune Algorithm for financial time series prediction; combining the properties of both recurrent and self-organised neural networks. The network is derived to ensure that a unique equilibrium state can be achieved to overcome the known stability and convergence problems. Extensive simulations for multi-step prediction in stationary and non-stationary time series were performed. The resulting projection made by the proposed network shows substantial profits on financial historical signals when compared to other neural network approaches. These simulations have suggested that dynamic immunology-based self-organised neural networks have a better ability to capture the chaotic movement in financial signals.

## I. INTRODUCTION

Financial time series forecasting is a difficult task due to the intrinsic complexity and affecting variables of the financial system. While many time series may be approximated with a high degree of confidence, financial time series are considered among the most difficult to analyse and predict [2]. Yao et al identifies several factors that complicate the process in predicting change in large financial data sets, such as the stock market; such systems are affected by many interrelated factors. These range from economic and political factors through to psychological factors affecting both powerful decision makers and individuals or consumers. Even traditional Economics studies inform us that even microeconomic drivers can affect demand consumption patterns and vice versa, thus forming a very complicated highly-interrelated system to model [26].

Decisions regarding investments and trading by large companies, financial institutions, and even the economic policy of governments rely on computer modelling forecasts [16]. Commercial and financial imperatives ensure that financial time-series prediction is awarded a large amount of coverage in research literature and this will no doubt continue to be the case.

Various methods and techniques for the prediction of financial time series have been and continue to be developed following some key principles. There are several distinct approaches used for prediction/forecasting, from statistical to artificial intelligence/machine learning techniques. The

traditional methods for financial time-series forecasting are based around statistical approaches. However, most of the developed prediction methods have very weak scientific support and were completely unsatisfactory due to the nonlinear nature of most of the financial time series data [7, 27, 12].

Accordingly, throughout the last decade, neural networks have gained ground moving from research experimentation tools, through to production and industrial decision-support systems; assisting managerial and advisory staff in making “critical financial decisions” [4]. As such, they are regarded as powerful tools for forecasting data, and have significant academic/research interest from fields such as artificial intelligence, machine learning, and statistics. They are nonlinear models that can be trained from known data sets, automatically mapping input and output data; effectively mapping past to future values in time series data. Such models are particularly powerful in their ability to thereby extract hidden structures and relationships that govern the data mappings [20].

There is clearly a business case for such large financial interests to invest in artificial neural networks as a prediction and forecasting method. Neural networks are considered as superior methods for time-series prediction, as compared to pure statistical methods, owing to their non-linear nature and capacity for data-based training [28, 17, 7]. In summary, the ability of a neural network to approximate and produce complex mappings between non-linear data is particularly fitting to this sort of problem domain.

Widyanto et al introduced a hidden layer inspired by the immune algorithm to the back-propagation networks to improve the way in which the NN could generalise. This took inspiration from the immune system algorithm that controls the construction of antibody-generating *B* cells. In their own experiments, the immunity-inspired NN was used to predict temperature-based food quality; it showed an 18% improvement in correct recognition on standard back-propagation NNs [29].

In this paper, a novel neural network architecture is proposed, based on the immune algorithm and self-organised neural network. The proposed network is called Dynamic Self-organised Multilayer neural network inspired by the Immune Algorithm (DSMIA). The multilayer perceptron (MLP) and

functional link networks are used for single and multi-step-ahead prediction of financial time series.

Ten financial time series are used to test the performance of the various networks such as the exchange rates and oil price time series.

## II. THE NETWORKS

Neural networks are trained to generate functional maps from given input to output data sets. However, differing architectures are better suited to different training dataset types and problem domains. For some problem domains, higher-order input combinations may be more suitable in order that the NN forms a fitting problem mapping.

This section provides an overview and brief introduction to existing Functional Link and Artificial Immune System-inspired Neural Networks.

### A. Functional Link Neural Network (FLNN)

FLNN was first introduced by Giles and Maxwell [10], as an addition to feedforward networks. The FLNN architecture extends the traditional feedforward network; aiming to map non-linear relationships between input and output data. Functional links replace the linear weights creating a degree of class separation.

FLNNs can use higher-order correlations of input components to perform non-linear mappings with fewer layers. As the architecture is simpler, it is intended to reduce the computational cost incurred in the training stage, whilst demonstrating good approximation performance [19]. Despite their otherwise linear nature, providing the input set is suitably descriptive, FLNN-architecture NNs can provide a learning network with greater information capacity and complex learning ability [19, 1, 11].

However, FLNN architectures are not without problems. They suffer from weight explosion, seeing an exponential increase across the number of inputs. As a result, only second or third order networks form practical maps [15, 21]. The output of FL-architecture NNs is specified as follows:

$$Y = \sigma \left( W_0 + \sum_j W_j X_j + \sum_{jk} W_{jk} X_j X_k + \sum_{jkl} W_{jkl} X_j X_k X_l + \dots \dots \dots \right) \quad (1)$$

where  $X$  and  $Y$  are the input and the output of the network, respectively. In this case,  $\sigma$  is a nonlinear transfer function, and  $w_0$  is the adjustable threshold.

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

Natural computation, specifically in the form of artificial immune systems has attracted much research interest; from the augmentation of machine-learning though to descriptive paradigms for self-organising systems [23].

The SONIA (Self-Organised Network inspired by Immune Algorithm) network is an enhancement to back-propagation neural networks that introduces a self-organising hidden layer that aims to improve both the generalisation and recognition capacity [29]. The rules governing self-organisation in the hidden layer are loosely modelled on the B-cell

construction/mutation in antibody production and antigen recognition.

The concept is based on the relationship between antigens and the B-cell; the latter of which produces antibodies when fully active, and mutate when they encounter known antigens in order that the mutated cells may recognise previously unknown antigens.

In the SONIA network, each hidden unit has a centre representative of the number of connections of attached input vectors. To avoid over-fitting, each centre contains a value representing the strength of connections between input units and their corresponding hidden units.

SONIA has been previously applied to financial time-series data prediction previously, with some success [13]. Two time series were used, modelling the daily exchange rates between the US Dollar and the Euro and the IBM common stock closing price. The simulation results showed that the SONIA network produced profit when used for financial time series prediction.

## III. DYNAMIC SELF-ORGANISED MULTILAYER NETWORK INSPIRED BY THE IMMUNE ALGORITHM (DSMIA)

In this section, recurrent neural network architecture is proposed. The new proposed architecture incorporates recurrent links within the structure, the operation of which is creates a self-organising layer; inspired by Artificial Immune System theory.

Previous work in this field includes the SMIA (Self-organised Multilayer NN inspired by the Immune Algorithm) [18] and the Dynamic Ridge Polynomial Higher-Order Neural Network (DRPHONN) [9].

These research works feed into, and are extended by this work to provide a novel network with the following characteristics; dynamic self-organising of hidden-layer units, and output-based feedback into the hidden layers, the justification for which follows:

Feedforward networks can only implement a static mapping of the input vectors. In order to model dynamic functions, it is essential to make use of a system capable of storing internal states and implementing complex dynamics. Neural networks with recurrent connections are dynamic systems with temporal/state representations.

Because of their dynamic structure, they have been successfully used for solving a variety of problems. This work is motivated by the potential of recurrent dynamic systems in solving complex real-world problems. The following section provides an overview of the proposed architecture of the Dynamic Self-organising Multilayer network inspired by the Immune Algorithm (DSMIA).

### A. Properties and Network Structure of the DSMIA

Applications in forecasting and signal processing require systems that can work with dynamic signals. As discussed previously, traditional feedforward networks can only handle static mappings. Including previous inputs in a now-augmented set of input values permits dynamic mapping to occur in the network by including previous state representations.

The DSMIA architecture differs from this dynamic representation; borrowing from biological networks, in common with the Dynamic Ridge Polynomial Neural Network (DRPNN) architecture [9]. By incorporating a self-organising internal feedback circuit, this allows DSMIA to learn the internal signal properties, providing better detection and prediction of precise signal trends.

Therefore, the structure of the DSMIA comprises three layers: the input, the self-organising hidden layer, and the output layers with feedback connection from the output layer to the input layer. The input layer holds copies of the current inputs as well as the previous output of the network; providing the network with memory. As such, previous behaviour of the network is also used as an input affecting current behaviour. Similar to the Jordan recurrent network [14], the output of the network is fed back to the input through the context units.

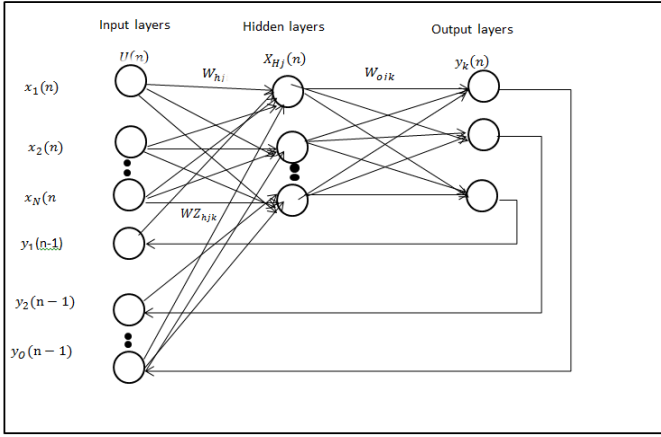


Figure 1. The structure of the proposed DSMIA network.

The hidden layer of the proposed network is trained in a similar manner to recurrent self-organising maps [24]. The feedback connection feeds the activation of the output node to the input context units. This feedback connection, shown in Figure 1, gives the network the ability to “memorise” data for use in its mapping, as described earlier. All the connection weights from the input layer to the self-organised layer and from the self-organised layer to the output layer are learnable.

### B. Mathematically Specifying the DSMIA

This section considers the dynamic equations that form the proposed DSMIA network. Let  $N$  be the number of external inputs to the proposed neural network, while  $O$  is the number of outputs.

The total number of inputs to the network is  $N+O$  where:

$$U(n) = \begin{cases} x_i(n), i=1, \dots, N \\ y_j(n-1), j=1, \dots, O \end{cases} \quad (2)$$

In this equation,  $x$  is representative of external inputs, and  $y$  the final outputs of the network. The function  $U(n)$  describes the  $n^{th}$  input of the network.

The function  $X_{Hj}(n)$  describes the input to the hidden, self-organising layer, and is determined as follows:

$$X_{Hj}(n) = f_{ht} \left( D_{hj}(n) \right) \quad (3)$$

Where  $f$  is a nonlinear function, further specified as:

$$D_{hj}(n) = V_{hj}(n) + Z_{hj}(n) \quad (4)$$

Comprising:

$$V_{hj}(n) = \alpha \sqrt{\sum_{i=1}^{NI} (W_{hji} - x_i(n))^2} \quad (5)$$

$$Z_{hj}(n) = \beta \sqrt{\sum_{k=1}^{NO} (WZ_{hjk} - y_k(n-1))^2} \quad (6)$$

$\alpha, \beta$  are experimentally selected parameters with  $0 < \alpha$  and  $0 < \beta$ .

To summarise; outputs of the self-organised layer form inputs to the nodes in the output layer; which in turn form the final output of the network. Therefore, the (final)  $k^{th}$  output is given by:

$$y_k = f_y \left( \sum_{j=1}^{N_H} w_{ojk} x_{Hj} + B_{ok} \right), \quad k=1, \dots, O \quad (7)$$

The term  $w_{ojk}$  represents the strength of the connection from the  $j^{th}$  hidden unit to the  $k^{th}$  output unit,  $B_{ok}$  is a bias associated with the  $k^{th}$  output unit, and  $f_y$  is a non-linear transfer function.

Turning back to the matter of artificial immune systems, recalling that the immune recognition function utilises the  $B$  cell; each which is responsible for generating antibodies matched to various antigens. In this case, the hidden unit is designed as an implementation of that  $B$  cell recognition algorithm from the immune system. If we assume that  $d(n)$  describes a function illustrative of the desired response of the network at time  $n$ , then the error of the network at time  $t$  can be specified as:

$$E(n) = d(n) - y(n) \quad (8)$$

Therefore, the average squared error between the actual and predicted values of the network is given by:

$$J(n) = \frac{1}{2} [E(n)]^2 \quad (9)$$

The change for any element  $w_{oj}$  of the weights matrix is calculated as follows:

$$\Delta w_{oj}(n) = -\eta \frac{\partial J(n)}{\partial w_{oj}} \quad (10)$$

where  $(i = 1, \dots, N_H, j = 1, \dots, O)$  and  $\eta$  is the learning rate, which is a positive constant.

The change for any particular element  $B_{oj}$  of the bias matrix is given as, where ( $j = 1 \dots, O$ ):

$$\Delta B_{oj}(n) = -\eta \frac{\partial J(n)}{\partial B_{oj}} \quad (11)$$

The self-organised layer is trained in a similar manner to the training algorithm of the recursive self-organising map proposed by Voegtlin [25]. In this case, the weights of the previous context  $WZ_{hjk}$  is updated in the same way as the weight of the external input  $WZ_{hji}$ . This is done by finding  $D$ , which is the distance between the input units and centroid of the  $j^{\text{th}}$  hidden units:

$$D_{hj}(n) = \alpha \sqrt{\sum_{i=1}^N (X_i(n) - W_{hji}(n))^2} + \beta \sqrt{\sum_{k=1}^O (y_k(n-1) - WZ_{hjk}(n))^2} \quad (12)$$

From  $D_{hj}(n)$ , the location of the closest match is determined according to the following equation :

$$c(n) = \text{argmin} D_{hj}(n) \quad (13)$$

If the shortest distance  $D_c$  is less than the simulation level  $s_l$  (between 0 and 1), then both the weight for the external input vector and the context vector are updated according to the following equations:

$$\begin{aligned} W_{hji}(n+1) &= W_{hji}(n) + \gamma D_c(n) \\ WZ_{hjk}(n+1) &= WZ_{hjk}(n) + \gamma D_c(n) \end{aligned} \quad (14)$$

$WZ_{hjk}$ , is the weight of the previous output and  $W_{hji}$  is the weight for the external inputs, and  $\gamma$  is the learning rate which is updated during the epochs.

#### IV. PREDICTION OF FINANCIAL TIME SERIES

Ten noisy financial time series signals are considered as shown in Table 1. All the signals were obtained from the Time Series Data Library [5].

**Table 1:** Time Series Data Used

No	Time Series Data	Size
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 Ind. Average stock opening price (DJIAO) 01/07/2000 - 11/11/2008	1605
5	Dow Jones Industrial Average stock closing price (DJIAO) 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

The networks are tested as predictors of one and five steps ahead of financial time. To evaluate this work, two sets of experiments are performed; on non-stationary and the stationary datasets. For non-stationary signals, all data signals are presented to the networks directly, i.e. without any transformation. Conversely, the stationary set of signals needs a degree of transformation before passing them to the networks, which may be familiar to those working in time series prediction. This is because precise daily prices (the non-stationary signals) are not considered as meaningful to trading, but more significant is the relative magnitude of price changes [3].

This sort of pre-processing is well understood to increase the performance of the NN in its forecasting ability. The processing technique adopted is termed RDP (Relative Difference in Price), and is a used by a number of researchers in this field [3, 22]. It creates a five-day measure of the relative difference in price data, smoothing the data and helping to reduce noise.

Post-transformation, all input and output variables are scaled between upper and lower bounds of the transfer function, as this has been shown in previous work to avoid further computational complexity [9].

#### V. TRAINING AND TESTING THE NETWORKS

The performance of the DSMIA network has been benchmarked against the performance of several other NN architectures. As this work extends the SMIA NN, the method of benchmarking extends that used in the SMIA work [18]. Performance is compared with MLP, FLNN, and SONIA, NN architectures.

The time series data is split into three data sets, forming the training, validation, and finally out-of-sample sets; representative of 25%, 25%, and 50% of total data, respectively. The prediction performance of all networks was evaluated using three financial metrics [7]; judging the network predictions as fit (or otherwise) for profit generation. Three statistical metrics [3] are also evaluated, which provide validation of the output signals. The precise algorithms used as shown in Table 2.

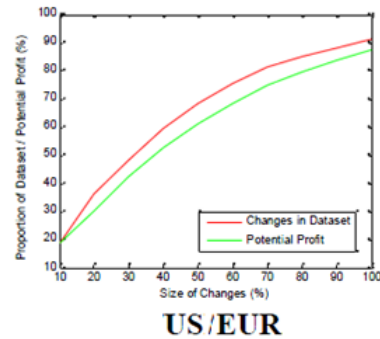


Figure 2. Size of changes in one data datasets compared with the percentage of profit from the whole dataset available within those changes

**Table 2:** Performance Metrics Equations  
 $n$  is the total number of data patterns.

$y$  and  $\hat{y}$  represent the actual and predicted output value, respectively.

Annualized Return (%AR)	Normalized Mean Squared Error (NMSE)
$AR = \frac{\text{Profit}}{\text{All profit}} * 100$ $\text{Profit} = \frac{252}{n} * CR, \quad CR = \sum_{i=1}^n R_i$ $R_i = \begin{cases} + y_i  & \text{if } (y_i)(\hat{y}_i) \geq 0, \\ - y_i  & \text{otherwise} \end{cases}$ $\text{All profit} = \frac{252}{n} * \sum_{i=1}^n \text{abs}(R_i)$	
$NMSE = \frac{1}{\sigma^2 n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ $\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2$ $\bar{y} = \sum_{i=1}^n y_i$	
Maximum Drawdown (MD)	Signal to Noise Ratio (SNR)
$MD = \min \left( \sum_{t=1}^n (CR_t - \max(CR_1, \dots, CR_t)) \right)$ $CR_t = \sum_{i=1}^t R_i, t = 1, \dots, n$ $R_i = \begin{cases} + y_i  & \text{if } (y_i)(\hat{y}_i) \geq 0, \\ - y_i  & \text{otherwise} \end{cases}$	
$SNR = 10 * \log_{10}(\text{sigma})$ $\text{sigma} = \frac{m^2 * n}{SSE}$ $SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2$ $m = \max(y_i)$	
Annualized Volatility (VOL)	Correct Directional Change (CDC)
$VOL = \sqrt{252} * \sqrt{\frac{1}{n-1} \sum_{i=1}^n (R_i - \bar{R})^2}$ $CDC = \frac{1}{n} \sum_{i=1}^n d_i$ $d_i = \begin{cases} 1 & \text{if } (y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0, \\ 0 & \text{otherwise} \end{cases}$	

## VI. SIMULATION RESULTS AND DISCUSSION

Results have been obtained that show average results from 50 simulations, conducted on out-of-sample data, in tables (3&4). This comprises stationary and non-stationary data predicted at one and five steps ahead of time signal.

Non-stationary financial signals are very difficult to predict with a degree of accuracy. Non-stationary signals are highly volatile and provide a signal with great instability and noise – frequent changes of behaviour including massive slumps and rises.

This pattern presents difficulty in training NNs; unable to respond properly to the price values' high-frequency variations. As such, all the networks, the DSMIA included, perform poorly on the AR (Annualised Return) measure on non-stationary signals.

However, for the stationary signals, the networks predicted high percentage of profits. NNs perform better on these signals as they are smoothed and undergo translation into the RDP measure, as described in Section IV. Stable predictions, and therefore, a greater profit measure is possible when evaluating against stationary signals.

Price time-series data acquired from three stock opening and closing (six in total) prices has been used in this study; NASDAQ-O and -C (opening and closing), DJIA-O and -C, and DJUA-O and -C. The inclusion of both opening and closing price signals is to investigate the differences between predictions for opening and closing stock results.

Results for the non-stationary signals show that in most cases, the prediction results for one and five steps ahead show a small difference between opening and closing stock prices. It is worth noting these differences are related to the raw data; itself affected by external factors, such as economic climate, blue-chip company financial announcements, and threats of war/civil unrest.

As can be seen in Tables (5&6), the DSMIA NN generated profit-based predictions on one and five-steps ahead. For stationary data, five-steps-ahead predictions, DSMIA generated greater profits than the majority of benchmarked NN algorithms.

In the DSMIA network, the learning is centred on the local properties of the signal; the self-organising layer aiming 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 any data changes or structural shifts common in non-stationary data.

Figure 2 plots the size of change in the US/EUR dataset. This is shown against the percentage of profit from the dataset available in those changes. It indicates that given larger dataset changes, the potential profit (on correct predictions) increases. For example, even with a 100% correct prediction rate, the available profit is variable. In US/EUR dataset, potential profit is always less than the size of change.

This pointed effect on AR (Annualised Return) has wider ranging implications when considered alongside the different behaviour of the NNs. For several of the benchmarked NN architectures, the behavioural objective of the training is to minimise error over the whole training dataset. However, the training effect on the self-organising layer's learning is localised; the network's concern is primarily with current data values. As such, the network is better placed to predict on these temporally-localised events.

## VII. CONCLUSION

This research work highlights an important contribution of a new neural network structure inspired by artificial immune systems. It is applied to and evaluated against prediction of values in a financial time series; namely its ability to approximate nonlinear financial series values.

The network has shown its advantages in forecasting both stationary and non-stationary signals, particularly regarding temporally-local training behaviour.

Early work on applying a regularization scheme to the DSMIA network is providing encouraging results using a weight-decay between the hidden and output nodes. As such, future work will further investigate and assess an improved, Regularised-DSMIA scheme. While weight-decay is not without its performance-related problems, work by others has shown that it can help to avoid over-fitting the network to



training data; as such, improving the network [6]. Another direction of research is to investigate the best choice of the network architecture, which include the number of inputs and the use of higher order terms in the input units similar to the FLNN. This may improve the performance of the proposed network.

## References

- [1] R. Cass and B. Radl, "Adaptive process optimization using Functional-Link Networks and Evolutionary Algorithm," *Control Eng. Practice*, Vol. 4, No. 11, pp. 1579-1584, 1996.
- [2] F. Castiglione, (2000). Forecasting price increments using an artificial Neural Network. *Adv. Complex Systems*, 1, pp. 1-12, 2000
- [3] L. J. Cao and F. E. H Tay, "Support Vector Machine with Adaptive Parameters in Financial time Series Forecasting". *IEEE Transactions on Neural Networks*, Vol. 14, No. 6, pp. 1506-1518, 2003.
- [4] A. S. Chen and M. T. Leung, Performance Evaluation of Neural Network Architectures: The case of Predicting Foreign Exchange Correlations. *Journal of Forecasting*, 24, pp. 403-420, 2005
- [5] Datastream International Ltd, London, Thompson Financial Limited, retrieved November 14, 2005.
- [6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*: John Wiley & Sons, Inc. Canada, 2001.
- [7] C. L. Dunis & M. Williams, "Modeling and trading the UER/USD exchange rate: Do Neural Network models perform better?" in *Derivatives Use, Trading and Regulation*, Vol. No. 8, 3, pages 211-239, 2002.
- [8] G. Fei, & Y. L. Yu, "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, 1994.
- [9] R. Ghazali, A. J. Hussain, N. Nawi, B. Mohamad, "Non-stationary and stationary prediction of financial time series using dynamic ridge polynomial neural network", *Neurocomputing*, Elsevier, 2009.
- [10] C. L. Giles and T. Maxwell, "Learning, invariance and generalization in high-order neural networks," In *Applied Optics*, vol. 26, no. 23, pp.4972-4978, 1987.
- [11] C. L. Giles, R. D. Griffin, and T. Maxwell, T. "Encoding Geometric Invariances in HONN," *American Institute of Physics*, pp. 310-309, 1998.
- [12] T. Hellstrom, & K. Holmstrom "Predicting the stock market", 1998
- [13] A. J. Hussain, D. Al-Jumeily "How good is the backpropagation neural network using a Self-organised Network Inspired by Immune Algorithm (SONIA) when used for multi-step financial time series prediction?", *LNCS*, vol. 4492, Part II, pp. 921-930, 2007.
- [14] M. I. Jordan, "Serial order: A parallel distributed processing approach," *Institute for Cognitive Science report 8604*, UC:San Diego, 1986.
- [15] T. Kaita, S. Tomita, and J. Yamanaka, "On a Higher-order Neural Network for distortion invariant pattern recognition," *Pattern Recognition Letters* 23, pp 977-984, 2002.
- [16] A. C. Knowles. "Higher-Order and Pipelined Neural Networks for Time Series Prediction of Currency Exchange Rates," MSc Thesis, Liverpool John Moores University, October 2005
- [17] S. L. Ho, M. Xie, & T. N. Goh, "A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction," *Computers & Industrial Engineering*, 42, pp. 371-375, 2002.
- [18] A Mahdi, A.J. Hussain, P. Lisboa and D. Al-Jumeily, "The Application of the Neural Network Model Inspired by the Immune System in Financial Time Series Prediction", 2nd International Conference on Developments in eSystems Engineering (DESE), 2009
- [19] L. Mirea, & T. Marcu, "System identification using Functional-Link Neural Networks with dynamic structure," 15th Triennial World Congress, Barcelona, Spain, 2002.
- [20] Y. Shachmurove, & D. Witkowska, (2000). Utilizing Artificial Neural Network Model to Predict Stock Markets. CARESS Working Paper #00-11. City University of New York, University of Pennsylvania, Tech University of Lodz. ([www.economics.sas.upenn.edu/system/files/00-11.pdf](http://www.economics.sas.upenn.edu/system/files/00-11.pdf))
- [21] G. Thimm, "Optimization of High Order perceptron," PhD Thesis, Swiss Institute of Technology (EPFL), 1995. ([http://publications.idiap.ch/downloads/papers/1997/Georg\\_thesis.pdf](http://publications.idiap.ch/downloads/papers/1997/Georg_thesis.pdf))
- [22] M. Thomason, "The practitioner method and tools," *Journal of Computational Intelligence in Finance*, vol. 7, no. 3, pp. 36-45, 1999.
- [23] J. I. Timmis, "Artificial immune systems: a novel data analysis technique inspired by the immune network theory," University of Wales, 2001.
- [24] T. Voegtlin and P.F. Dominey. Recursive self-organizing maps. In N.Allison, H.Yin, L.Allinson, and J.Slack, editors, *Advances in Self-Organizing Maps*, Vol. 5, pp. 210-215, Springer, 2001
- [25] T. Voegtlin, "Recursive self-organizing Maps", *Neural Networks*, 15, pp. 979-991, 2002.
- [26] J. Yao, H. Poh, & T. Jasic, (1996). Foreign exchange rates forecasting with neural networks. *International Conference on Neural Information Processing*, Hong Kong, Springer-Verlag, pp. 754-759.
- [27] J. Yao, & C. L. Tan "A case study on neural networks to perform technical forecasting of forex," *Neurocomputing* 34, pp. 79-98, 2000.
- [28] S. Yumlu, F. S. Gurgen And N. Okay (2005). A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction. *Pattern Recognition Letters* 26, pp.2093-2103
- [29] M. R. Widyanto, H. Nobuhara, K. Kawamoto. K. Hirota, B. Kusumoputro, "Improving recognition and generalization capability of back-propagation NN using self-organized network inspired by immune algorithm", *Applied Soft computing*, Vol 6, 2005, pp. 72-84.

TABLE 3  
AVERAGE RESULTS ON STATIONARY SIGNALS FOR THE PREDICTION OF 1-STEP AHEAD

Performance Measures	Neural Networks	US/UK	JP/US	US/EU	NASDAQO	NASDAQC	DJIAO	DJIAC	DJUAO	DJUAC	OIL
AR(%)	MLP	65.14634	64.72067	69.91986	60.5218	62.32483	59.49555	58.85693	52.97216	52.01663	51.19201
	FLNN	78.18972	77.67146	78.58212	67.91466	67.16377	74.08653	73.57757	72.81365	70.84033	73.64221
	SONIA	73.15769	76.19371	71.00268	62.66176	63.35985	63.71905	62.29099	67.01270	68.02663	72.16195
	DSMIA	76.40196	76.28788	72.941440	66.466713	62.866065	64.859029	66.071119	69.986696	70.593948	73.352611
MD	MLP	-1.4304	-1.5806	-1.7923	-4.8447	-4.8762	-4.4906	-2.9165	-6.1459	-5.6157	-19.024
	FLNN	-1.14585	-1.03966	-0.98186	-8.37879	-5.31515	-1.87348	-1.88483	-2.51275	-2.974	-8.70562
	SONIA	-1.51535	-1.24388	-2.69518	-8.84982	-8.32449	-6.60512	-7.99077	-3.55542	-4.17998	-8.32751
	DSMIA	-1.29011	-1.50278	-1.91315	-6.89071	-8.230259	-7.57360	-6.60576	-3.66422	-3.41091	-7.08040
AV	MLP	4.526048	5.717462	4.666936	13.27316	12.58836	11.18889	11.27457	12.48506	12.56166	67.3681
	FLNN	4.204148	5.379993	4.428998	12.92789	12.31514	10.48708	10.56609	11.63671	11.78889	59.43905
	SONIA	4.346859	5.424529	4.63746	13.20117	12.53349	11.00119	11.12426	11.93661	11.94297	60.15546
	DSMIA	4.256646	5.420273	4.590001	12.999914	12.560987	10.951396	10.952866	11.787191	11.802643	59.745353
SR	MLP	14.50009	11.36989	15.00254	4.575426	4.955069	5.322315	5.22938	4.267582	4.169173	0.766643
	FLNN	18.59827	14.43758	17.74275	5.253586	5.453892	7.064812	6.963861	6.257906	6.010538	1.246236
	SONIA	16.83866	14.04827	15.34295	4.742449	5.057593	5.800506	5.610039	5.614729	5.678424	1.20381
	DSMIA	17.951165	14.083690	15.897520	5.117712	5.007122	5.928563	6.038112	5.937937	5.981827	1.228245
SNR(dB)	MLP	20.35	21.75	22.29	23.7	22.83	24.45	23.26	23.84	23.9	21.03
	FLNN	22.95	23.99	24.26	25.07	23.51	24.99	24.99	25.06	25.15	23.13
	SONIA	21.93	23.41	23.04	24.14	22.75	23.61	23.53	25.02	25.03	22.2
	DSMIA	22.43	23.36	23.08	24.49	22.65	23.72	23.67	25.31	25.35	22.69
CDC	MLP	67.97	62.59	62.95	65.87	64.53	63.36	64.88	63.34	64.61	61.36
	FLNN	66.46	63.05	62.14	66.94	63.39	63.51	64.44	59.56	60.08	58.88
	SONIA	66.09	62.35	63.07	66.82	62.39	61.92	62.98	60.48	59.95	62.76
	DSMIA	65.88	59.77	62.80	67.62	62.03	60.38	61.65	60.56	61.11	61.67

TABLE 4  
AVERAGE RESULTS ON STATIONARY SIGNALS FOR THE PREDICTION OF 5-STEP AHEAD

Performance Measures	Neural Networks	US/UK	JP/US	US/EU	NASDAQO	NASDAQC	DJIAO	DJIAC	DJUAO	DJUAC	OIL
AR(%)	MLP	84.72936	81.19235	91.46321	77.57114	83.09106	74.77388	65.96346	74.07559	68.93831	75.89575
	FLNN	77.68125	86.30544	86.37381	85.91437	85.81948	88.28172	88.31142	87.44099	86.69399	93.69838
	SONIA	88.15049	87.17208	91.84363	85.16167	85.02744	85.3598	84.89345	81.28562	86.66031	91.82156
	DSMIA	89.047661	87.201969	92.028790	86.323692	85.105993	86.629872	86.106254	87.135184	87.517079	92.039333
MD	MLP	-3.15533	-3.54864	-1.78913	-10.1308	-7.34318	-8.24405	-13.0605	-11.1968	-20.4777	-8.03638
	FLNN	-5.87622	-2.72277	-5.14140	-6.84715	-7.08089	-3.82649	-3.72506	-3.25588	-5.36821	-8.75378
	SONIA	-3.2779	-2.72278	-1.39054	-6.19597	-7.36317	-8.41446	-8.37277	-7.78359	-3.70212	-12.60818
	DSMIA	-2.963665	-2.722772	-1.432039	-6.109268	-7.658861	-9.305728	-9.268669	-3.357676	-3.410785	-13.3041 -11.644154
AV	MLP	16.24499	17.56791	15.69251	37.68809	36.53018	33.66308	35.11867	38.27525	39.02029	194.2871
	FLNN	17.04838	16.88852	16.39043	35.7455	35.85963	30.99444	31.03799	35.79927	36.26686	152.054
	SONIA	15.87492	16.76482	15.6362	35.93796	36.06489	31.63171	31.78481	37.03390	36.27336	157.372
	DSMIA	15.757916	16.760712	15.607786	35.637459	36.041823	31.347378	31.426138	35.864341	36.089621	156.745293
SR	MLP	5.246373	4.62544	5.829595	2.063196	2.278035	2.225729	1.888711	1.941499	1.786946	0.390889
	FLNN	4.569979	5.110325	5.289899	2.403571	2.393323	2.848333	2.845357	2.442637	2.390524	0.61651
	SONIA	5.561351	5.199892	5.873841	2.369793	2.357881	2.633558	2.672691	2.564821	2.389255	0.584465
	DSMIA	5.661860	5.202832	5.896499	2.422628	2.361925	2.766082	2.754751	2.429673	2.425167	0.588284
SNR(dB)	MLP	23.14	22.56	26.71	24.46	24.93	23.73	23.09	24.53	24.21	20.24
	FLNN	24.14	22.61	24.37	26.5	25.9	27.1	27.15	27.55	27.39	25.31
	SONIA	24.98	23.02	23.41	25.22	24.33	24.81	24.68	26.28	27.27	25.37
	DSMIA	24.85	22.30	23.38	24.93	24.00	24.72	25.62	27.10	27.36	25.02
CDC	MLP	63.7	62.18	64.65	61.47	59.92	61.92	61.45	60.86	59.66	56.88
	FLNN	63.47	64.44	63.17	61.23	60.72	63.48	63.84	62.61	63.09	56
	SONIA	65.49	63.16	65.09	62.49	59.48	62.91	62.99	61.72	62.97	60.06
	DSMIA	65.93	63.48	65.23	61.85	59.39	63.27	63.54	62.35	62.41	61.21

TABLE 5  
AVERAGE RESULTS ON NON-STATIONARY SIGNALS FOR THE PREDICTION OF 1-STEP AHEAD

Performance Measures	Neural Networks	US/UK	JP/US	US/EU	NASDAQO	NASDAQC	DJIAO	DJIAC	DJUAO	DJUAC	OIL
AR(%)	MLP	6.669613	-0.473902	7.703251	-6.27845	-10.20173	-9.977635	-12.48747	-7.6654061	-6.666097	4.151457
	FLNN	1.993203	-0.905553	9.82592	-3.3082592	-5.49312	-6.546531	-9.72016	-6.254837	-6.376194	-6.630352
	SONIA	2.073759	-8.271959	2.414358	-3.619703	3.531254	-15.34971	-15.34804	-5.068463	-5.201504	24.689167
	DSMIA	5.075575	-4.069113	5.498935	-1.201741	7.886264	-3.891887	4.078142	-1.449810	0.355431	26.762486
MD	MLP	-14.957188	-22.500975	-11.293929	-15.470936	-70.907413	-63.928973	-73.355125	-67.988187	-67.160675	-13.563781
	FLNN	-18.341226	-21.509273	-10.26734	-46.467443	-51.820357	-49.59282	-66.6875	-51.65052	-51.482152	-100.89227
	SONIA	-19.83157	-32.49779	-13.49779	-64.55476	-49.69712	-88.91595	-92.87567	40.012786	-45.16187	-50.27416
	DSMIA	-16.272571	-15.434371	-9.23998	-30.243216	-40.470801	-31.948202	-14.144426	-51.271605	-44.666601	-51.651684
AV	MLP	11.403352	12.802219	10.106966	30.732116	30.476655	27.480595	27.88965	29.921184	30.590123	33.256952
	FLNN	11.410718	12.802582	10.095021	30.780716	30.565166	27.535189	23.240831	29.992717	30.660917	131.37748
	SONIA	11.399141	12.77795	10.105815	30.733801	30.472487	27.361664	27.773873	29.9737	30.642176	128.19774
	DSMIA	11.383173	12.794142	10.101788	30.738439	30.495996	27.500631	27.869717	30.643680	29.981559	128.001300
SR	MLP	0.585399	-0.037	-0.762452	-0.20439	-0.3351	-0.36342	-0.44844	-0.25648	-0.21824	-0.000123
	FLNN	0.175035	-0.07084	0.974008	0.107544	-0.17996	-0.23811	-0.41859	-0.20877	-0.20809	-0.05142
	SONIA	0.182488	-0.64743	0.239181	-0.1177	0.115987	-0.56231	-0.55418	-0.16912	-0.16979	0.193607
	DSMIA	0.446797	-0.318106	0.544569	-0.039112	0.259041	-0.141980	0.146780	-0.047426	0.011822	0.209142
SNR(dB)	MLP	16.3	17.68	13.41	17.74	19.38	12.7	13.07	14.19	14.94	19.08
	FLNN	29.26	25.72	30.06	31.13	30.84	29.4	24.76	32.86	32.53	9.04
	SONIA	20.5	27.04	13.97	19.83	16.05	16.97	18.51	16.51	16.88	11.44
	DSMIA	17.05	24.24	11.97	15.45	17.87	14.87	13.07	12.72	12.92	11.26
CDC	MLP	52.78	47.7	52.34	47.6	48.06	48.41	47.66	49.04	48.58	51.34
	FLNN	51.31	46.63	52.13	47.54	48.64	49.22	48.47	48.25	47.28	46.29
	SONIA	50.31	45.45	52.79	47.77	50.55	46.1	45.56	48.16	47.63	57.1
	DSMIA	50.80	47.46	51.40	49.20	52.25	51.67	51.10	50.92	51.48	57.32

TABLE 6  
AVERAGE RESULTS ON NON-STATIONARY SIGNALS FOR THE PREDICTION OF 5-STEP AHEAD

Performance Measures	Neural Networks	US/UK	JP/US	US/EU	NASDAQO	NASDAQC	DJIAO	DJIAC	DJUAO	DJUAC	OIL
AR (%)	MLP	-0.559585	-3.59391	1.117382	-2.40269	1.65726	-0.321544	-0.555403	-3.439273	-3.523605	0.01561
	FLNN	-1.2738	-5.9573	-0.1571	-3.8363	-0.915	-1.3329	-4.0139	-2.9823	-2.9541	-6.3419
	SONIA	-1.48508	2.839703	2.819774	-5.12982	-1.00699	2.440176	1.880855	2.857381	1.735058	-6.13704
	DSMIA	1.824697	4.883264	5.924942	0.102613	-0.645413	2.444301	2.555257	4.642005	4.005996	11.290324
MD	MLP	-14.8441	-20.2364	-11.8548	-37.1925	-34.1537	-37.7277	-40.42709	-49.0705	-51.062871	-88.71052
	FLNN	-16.0545	-20.2988	-12.3168	-38.3696	-40.0468	-39.5673	-47.5507	-50.6108	-15.8044	-94.3542
	SONIA	-17.776	-19.8414	-8.63958	-38.6083	-32.3624	-20.558	-22.1424	-35.7076	-34.5416	-90.0799
	DSMIA	-13.364766	-14.536049	-8.593920	-37.366685	-32.550353	-21.343170	-20.984023	-30.055884	-30.602317	-87.792361
AV	MLP	11.42579	12.8158	10.1309	30.8166	30.5985	27.5912	27.5927	29.9932	30.70504	131.6793
	FLNN	11.4246	12.8066	10.1311	30.8154	30.6204	27.5939	23.288	30.0396	30.7071	131.135
	SONIA	11.41287	12.7995	10.11326	30.77346	30.59728	27.58149	28.00402	29.9849	30.67759	130.3251
	DSMIA	11.417538	12.797590	10.109854	30.767597	30.602293	27.579252	28.001119	29.992023	30.662968	130.448316
SR	MLP	-0.049097	-0.28073	0.11028	-0.07803	0.054188	-0.011719	-0.020069	-0.11477	-0.114946	-0.000611
	FLNN	-0.1116	-0.4654	-0.0155	-0.1245	-0.0299	-0.0483	-0.1725	-0.0993	-0.0962	-0.0484
	SONIA	-0.13091	-0.22183	0.378768	-0.16675	-0.1253	0.088481	0.067178	0.095616	0.056716	-0.04803
	DSMIA	0.159847	0.381776	0.586446	0.003337	-0.021085	0.088642	0.091279	0.154933	0.130994	0.087339
SNR(dB)	MLP	14.7	16.34	13.22	16.54	17	12.39	12.42	14.07	15.49	10.6
	FLNN	15.3	16.51	14.02	23.88	27.7	26.59	22.23	29.39	29.51	9.59
	SONIA	16.83	21.66	13.32	19.94	17.1	16.83	17.14	13.38	14.88	10.92
	DSMIA	15.30	22.04	11.57	16.73	18.99	19.07	12.83	13.48	13.76	10.78
CDC	MLP	52.27	47.95	50.79	48.44	49.79	52.28	51.27	49.57	49.44	49.42
	FLNN	52.52	46.98	50.91	48.04	50.99	52.62	49.44	49.3	49.27	48.58
	SONIA	53.2	50.98	51.89	48.05	48.41	52.54	52.36	50.15	49.83	49.03
	DSMIA	51.52	51.52	52.27	50.45	49.45	52.73	51.60	51.13	50.33	53.00