Al-Jumeily & Hussain, *Cogent Engineering* (2015), 2: 985005
http://dx.doi.org/10.1080/23311916.2014.985005

cogent engineering

CrossMark

COMPUTER SCIENCE | RESEARCH ARTICLE

The performance of immune-based neural network with financial time series prediction

Dhiya Al-Jumeily[1]* and Abir J. Hussain[1]

Received: 19 June 2014
Accepted: 31 October 2014
Published: 08 January 2015

*Corresponding author: Dhiya Al-Jumeily, Faculty of Engineering and Technology, Applied Computing Research Group, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK
E-mail: d.aljumeily@ljmu.ac.uk

Reviewing editor:
Duc Pham, University of Birmingham, UK

Additional information is available at the end of the article

Abstract: This paper presents the use of immune-based neural networks that include multilayer perceptron (MLP) and functional neural network for the prediction of financial time series signals. Extensive simulations for the prediction of one-and five-steps-ahead of stationary and non-stationary time series were performed which indicate that immune-based neural networks in most cases demonstrated advantages in capturing chaotic movement in the financial signals with an improvement in the profit return and rapid convergence over MLPs.

Subjects: Artificial Intelligence; Computation; Software Engineering & Systems Development; Technology

Keywords: financial signals; immune-based neural network; time series prediction

1. Introduction

Neural networks have been shown to be a promising tool for forecasting financial times series. Numerous research and application of neural networks for business applications have proven their advantage in relation to classical methods that do not include artificial intelligence. What makes this particular use of neural networks so attractive to financial analysts and traders is the fact that government sectors and companies have used this technique to make decisions on investment and trading. However, when the number of inputs to the model and the number of training examples becomes extremely large, the training procedure for ordinary neural network architectures becomes

ABOUT THE AUTHOR

Dhiya Al-Jumeily is a principal lecturer in applied computing and leads the applied computing research group at the faculty of Engineering and Technology. He is also the Head of Enterprise for the faculty and he has already developed fully the first online MSc and BSc Courses for Liverpool John Moores University. He has published numerous referred research papers in multidisciplinary research areas including: Applied Artificial Intelligence, Neural Networks, Signal Prediction, Telecommunication Fraud Detection, Health Informatics, Personalised Health Care, Image Compression and Technology Enhanced Learning. He is a PhD supervisor and an external examiner for the degree of PhD. He has been actively involved as a member of editorial board and review committee for a number of peer-reviewed international journals, and is on program committee or as a general chair for a number of international conferences.

PUBLIC INTEREST STATEMENT

As businesses grow in size and scale it becomes increasingly difficult to predict how its' stocks and revenue will fluctuate. This can be sorted by using artificial neural networks, data processes that monitor and react to data according to how they were trained, it allows them to not only recognise patterns but also, in certain types of artificial neural network, teach itself to react differently to that pattern. The major limitation of artificial neural networks is the time it takes to train them, which increases exponentially as you increase the amount of data they must process. The purpose of this study is to compare a new form of artificial neural network against the current forms and evaluate the outcome of the tests, in hopes of seeing an improvement in performance for the new version over the old.

cogent • oa

tremendously slow and unduly tedious. To overcome such time-consuming operations, this research work proposes the use of immune-based neural network to improve the recognition and generalisation capability of the backpropagation neural networks.

The efficient market hypothesis states that a stock price, at any given time, reflects the state of the environment for that stock at that time. That is, the stock price is dependent on different variables, such as news events, other stock prices and exchange rates. The hypothesis suggests that future trends are completely unpredictable and subject to random occurrences. Thus making it infeasible, to use historical data or financial information, to produce above average returns. However, in reality, market responses are not always instantaneous. Markets may be slow to react due to poor human reaction time or other psychological factors associated with the human actors in the system. Therefore, in these circumstances, it is possible to predict financial data, based on previous results (Jensen, 1978). There is a significant body of evidence showing that markets do not work in a totally efficient manner. Much of the research shows that stock market returns are predictable by various methods such as; time series data analysis on financial and economic variables (Fama & French, 1989; Fama & Schwert, 1977; Ferson, 1989).

Various studies have been carried out on the use of neural networks for financial time series prediction; these include the forecasting behaviour of the financial market using neural networks. Multiple decisions, each of which affects the performance of the neural networks forecasting model, must be made, including; which data to use, the size and the architecture of the neural network systems (Zhang, 2003). The following are some of the difficulties of using neural networks in financial time series applications:

- There are infinitely many models which fit the training data well, but few of them generalise well. Supplementary degrees of freedom may lead to a better fitting of the model during the training of the network, but to worse generalisation ability on the out-of-sample data (Lendasse, de Bodt, Wertz, & Verleysen, 2000).
- In order to form a more accurate model, it is desirable to use as large training set as possible. However, for the case of highly non-stationary data, increasing the size of training set results in more data with statistics that are less relevant to the task at hand being used in the creation of the model.
- The high noise and too many parameters (compared to the number of data available) make the models prone to overfitting (Dorffner, 1996; Lendasse et al., 2000).
- The requirement of large number of sample data, due to their large number of free parameters (Dorffner, 1996). The limitation exists for the fact that some new founded companies do not have much of the previous data.

To improve the recognition and generalisation capability of the backpropagation neural networks, Widyanto, Nobuhara, Kawamoto, Hirota, and Kusumoputro (2005) used a hidden layer inspired by immune algorithm (SMIA) for the prediction of sinusoidal signal and time temperature-based quality food data. Their simulations indicated that the prediction of sinusoidal signal showed an improvement of 1/17 in the approximation error in comparison to the backpropagation and 18% improvement in the recognition capability for the prediction of time temperature-based quality food data.

In this paper, we propose the use of a multilayer perceptron (MLP), the functional link networks and the self-organised neural network inspired by the SMIA for single and multi-step ahead prediction of financial time series. Furthermore, a novel application of the regularisation technique is used with the self-organised MLPs network that is inspired by the immune algorithm (R_SMIA). The aim is to increase the generalisation ability of the SMIA network for financial time series prediction and to avoid the problem of overfitting for the purpose of improving the prediction ability of the self-organised multilayer neural network which is inspired by SMIA.

Ten financial time series are used to test the performance of the various networks such as the exchange rates time series and the oil price. In these extensive experiments, our primary interest is to concentrate on the profitable value contained in the predictions for all neural network models and hence during generalisation. The work focuses more on how the network generates the profits. For this reason, the neural networks structure, which provides the highest percentage of annualised return (AR) on out-of-sample data, is considered to be the best. A new training algorithm was utilised with the self-organised neural network that is inspired by the SMIA using weight decay; the simulation results indicated significant improvement of the proposed training algorithm over the standard network.

2. Financial time series forecasting

Time series forecasting is the process of predicting future values using current values. Forecasting the behaviour of the financial market is a non-trivial task due to its non-linear and non-stationary behaviour, furthermore it has been suggested that some financial time series are not predictable (Thimm, 1995).

Dunis and Wiliams (2002) implemented Neural Network Regression to forecast foreign exchange rates on UER/USD time series data. The study was benchmarked against several traditional forecasting techniques including Naïve Strategy, MACD Strategy, ARMA Methodology and Logit Estimation. Their observations have confirmed the applicability of neural network for financial forecasting.

Yao and Tan (2000) examined the forecasting performance of neural network on the exchange rates between American Dollar and five other major currencies; Japanese Yen, Deutsch Mark, British Pound, Swiss Franc and Australian Dollar. The results showed that without the use of extensive market data or knowledge, useful prediction can be made and significant paper profits can be achieved for out-of-sample data. They also concluded that a backpropagation network used in their study has proved to be adequate for forecasting and simple technical indicators such as moving average (MA) are enough.

Another approach for time series forecasting can be found in (Lawrence & Giles, 2000) which analysed the predictability of major world stock markets such as Canada, France, Germany, Japan, United Kingdom (UK) the United States (US), and the world excluding US (World) using MLP models. They found that MLP models with logistic activation functions predict daily stock returns better than the traditional ordinary least squares and general linear regression models. Neural networks are promising tool for forecasting financial times series. They have been widely used to model the behaviour of financial time series and to forecast future values (Yao & Tan, 2000).

3. Traditional approaches to time series prediction

The standard method for time series prediction is the statistical linear approach. In this approach, the signal S_n is considered the output of a system with unknown input u_n and its value is determined by the linear combinations of previous outputs and inputs according to the following equation (Makhoul, 1975):

$$S_n = \sum_{k=1}^{P} a_k S_{n-k} + G \sum_{m=0}^{q} b_m u_{n-m}, \quad b_0 = 1 \tag{1}$$

where a_k, b_m and G are the model parameters. Usually, the input u_n is modelled by a zero mean Gaussian noise source. The above equation can be specified in the frequency domain by taking the Z transform of both sides of the equation. Let $H(Z)$ represent the transfer function of the system in the Z domain, then:

$$H(Z) = \frac{S(Z)}{U(Z)} = G \frac{1 + \sum_{m=1}^{q} b_m z^m}{1 + \sum_{k=1}^{p} a_k z^k} \tag{2}$$

And the Z transform of the signal is:

$$S(Z) = \sum_{n=\infty}^{\infty} s_n z^n \tag{3}$$

In this case, the roots of the numerator and the denominator of the transfer function $H(Z)$ are the zeros and the poles of the model, respectively. When $a_k = 0$, the model is considered as all zeros and called the Moving Average (MA) model, when $b_m = 0$, the model is considered as all poles and known as Autoregressive (AR) model, while a model that has pole sand zeros values is referred to as an autoregressive moving average (ARMA) model.

For the non-linear model, we have:

$$g(S_n, S_{n-1}, S_{n-2}, \ldots) = u_n \tag{4}$$

In this case, u_n is a zero mean white noise. The function g is a highly non-linear and very complicated. Non-linear prediction can be determined using either the Volterra or the bilinear models, where the process is assumed to be inevitable, i.e. u_n can be approximated using a finite number of terms (Manikopoulos, 1992) and in which:

$$S_n = \sum_i a_i u_n + \sum_i \sum_j a_{ij} u_{ni} u_{nj} + \sum_i \sum_j \sum_k a_{ijk} u_{ni} u_{nj} u_{nk} + \ldots \tag{5}$$

Using the discrete Volterra series expansion. Where $\{u_i\}$, $\{u_{ij}\}$, $\{u_{ijk}\}$ are Gaussian random variables and $\{a_i\}$, $\{a_{ij}\}$, $\{a_{ijk}\}$ are sets of constant coefficients.

Using the bilinear model, we can determine S_n as follows:

$$S_n = \sum_{i=1}^{P} a_i S_{ni} + \sum_{j=1}^{q} a_j u_{nj} + \sum_{l=1}^{P} \sum_{m=1}^{q} b_{lm} S_{nl} u_{nm} \tag{6}$$

where c_0=0, and u_n is a white noise process.

To solve the non-linear model, it is required to determine the unknown parameters, which are usually very difficult to determine using traditional methods. Neural networks can be used to solve this problem in which the parameters (weights and biases) are determined implicitly using suitable training algorithms.

4. The networks

Although most neural network models share a common goal in performing functional mapping, different network architectures may vary significantly in their ability to handle different types of problems. For some tasks, higher order combinations of some of the inputs or activations may be appropriate to help form good representation for solving the problems.

This section is concerned with introducing Functional link neural network, and the Immune-based neural networks.

4.1. Functional link neural network (FLNN)

FLNN was first introduced by Giles and Maxwell (1987). It naturally extends the family of theoretical feedforward network structure by introducing non-linearities in inputs patterns enhancements (Durbin & Rumelhart, 1989). These enhancement nodes act as supplementary inputs to the network. FLNN calculates the product of the network inputs at the input layer, while at the output layer the summations of the weighted inputs are calculated.

http://dx.doi.org/10.1080/23311916.2014.985005

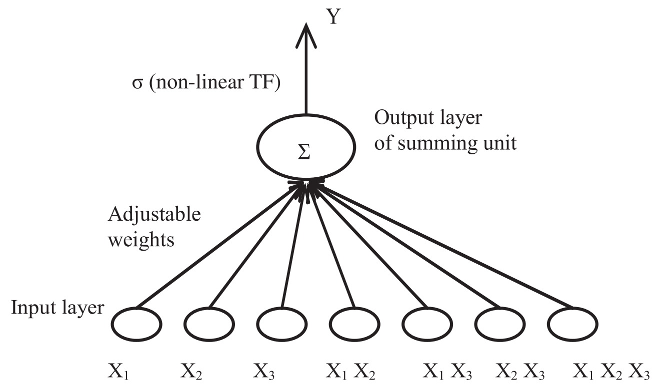


Figure 1. Functional link neural network.

FLNN can use higher order correlations of the input components to perform non-linear mappings using only a single layer of units. Since the architecture is simpler, it is suppose to reduce computational cost in the training stage, whilst maintaining good approximation performance (Mirea & Marcu, 2002). A single node in FLNN model could receive information from more than one node by one weighted link. The higher order weights, which connect the high order terms of the input products to the upper nodes, have simulated the interaction among several weighted links. For that reason, FLNN could greatly enhance the information capacity and complex data could be learnt (Cass & Radl, 1996; Giles & Maxwell, 1987; Mirea & Marcu, 2002).

Fei and Yu (1994) showed that FLNN has a powerful approximation capability than conventional Backpropagation network, and it is a good model for system identification (Mirea & Marcu, 2002). Cass and Radl (1996) used FLNN in the optimisation process and found that FLNN can be trained much faster than MLP network without scarifying computational capability. FLNN has the properties of invariant under geometric transformations (Durbin & Rumelhart, 1989). The model has the advantage of inherent invariance, and only learns the desired signal. Figure 1 shows an example of third-order FLNN with three external inputs X_1, X_2 and X_3, and four high order inputs which act as supplementary inputs to the network.

The output of FLNN is determined as follows:

$$Y = \sigma\left(W_0 + \sum_j W_j X_j + \sum_{j,k} W_{jk} X_j X_k + \sum_{j,k,l} W_{jkl} X_j X_k X_l + \dots\right) \tag{7}$$

where σ is a non-linear transfer function, and W_0 is the adjustable threshold. Unfortunately, FLNN suffers from the explosion of weights which increase exponentially with the number of inputs. As a result, second- or third-order functional link networks are considered in practice (Kaita, Tomita, & Yamanaka, 2002; Thimm, 1995).

4.2. The self-organised network inspired by the SMIA

The SMIA which was first introduced by Timmis (2001) has attracted many interests. Widyanto et al. (2005) introduced a method to improve recognition as well as generalisation capability of the backpropagation by suggesting a self-organisation hidden layer inspired by SMIA network. The input vector and hidden layer of SMIA network are considered as antigen and recognition ball, respectively. The recognition ball which is the generation of the immune system is used for hidden unit creation.

In time series prediction, the recognition balls are used to solve overfitting problem. In the immune system, the recognition ball has a single epitope and many paratopes. In which, the epitope is attached to *B* cell and paratopes are attached to antigen, where there is a single *B* cell that represents several antigens.

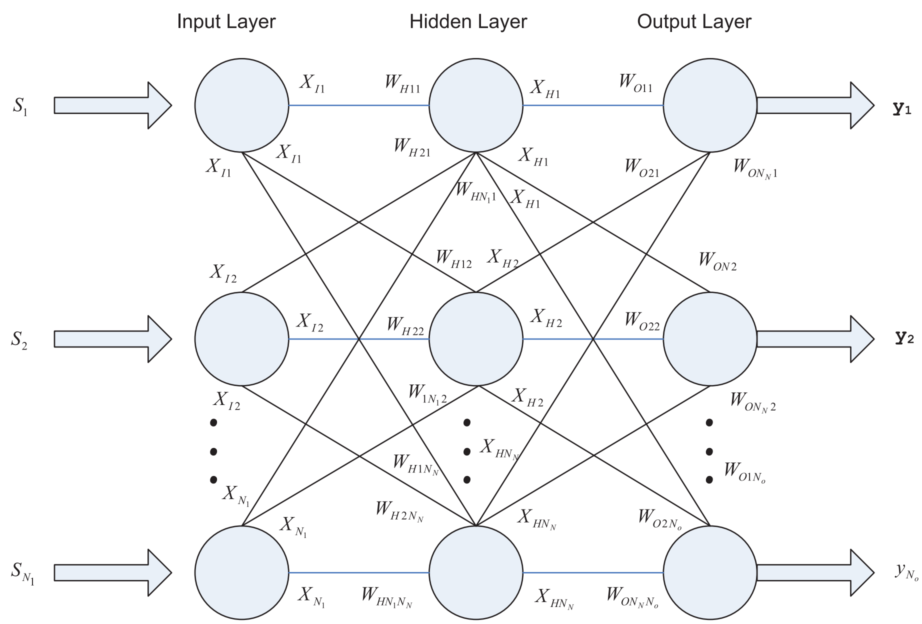


Figure 2. The structure of the SMIA network (Widyanto et al., 2005).

For SMIA network, each hidden unit has a centre that represents the number of connections of the input vectors that are attached to it. To avoid the overfitting problem, each centre has a value which represents the strength of the connections between input units and their corresponding hidden units. The SMIA network consists of three layers which are input, self-organised and output layers as shown in Figure 2.

In what follows the dynamic equations of SMIA network are considered. The ith input unit receives normalised external input S_i where $i = 1, \ldots, N_I$ and N_I represents the number of inputs. The output of the hidden units is determined by the Euclidean distance between the outputs of input units and the connection strength of input units and the jth hidden unit. The use of the Euclidian distance enables the SMIA network to exploit locality information of input data. This can lead to improve the recognition capability. The output of the jth hidden unit is determined as follows:

$$X_{Hj} = f\left(\sqrt{\sum_{i=1}^{N_I}\left(W_{Hij} - X_{Ii}\right)^2}\right) \quad j = 1, \ldots, N_H \tag{8}$$

where W_{Hij} represents the strength of the connection from the ith input unit to the jth hidden unit, and f is a non-linear transfer function.

The outputs of the hidden units represent the inputs to the output layer. The network output can be determined as follows:

$$y_k = g\left(\sum_{j=1}^{N_H} W_{ojk} X_{Hj} + b_{ok}\right) \quad k = 1, \ldots, N_o \tag{9}$$

where W_{ojk} represents the strength of the connection from the jth hidden unit to the kth output unit and b_{ok} is the bias associated with the kth output unit, while g is the non-linear transfer function.

4.3. Training the SMIA network

In this subsection, the training algorithm of the SMIA network will be shown. Furthermore, a *B* cell construction-based hidden unit creation will be described.

For the SMIA, inside the recognition ball, there is a single *B* cell which represents several antigens. In this case, the hidden unit is considered as the recognition ball of SMIA. Let $d(t + 1)$ represents the desired response of the network at time $t + 1$. The error of the network at time $t + 1$ is defined as:

$$e(t+1) = d(t+1) - y(t+1) \tag{10}$$

The cost function of the network is the squared error between the original and the predicted value, that is:

$$J(t+1) = \frac{1}{2}[e(t+1)]^2 \tag{11}$$

The aim of the learning algorithm is to minimise the squared error by a gradient descent procedure. Therefore, the change for any specified element W_{oij} of the weights matrix is determined according to the following equation:

$$\Delta W_{oij}(t+1) = -\eta \frac{\partial J(t+1)}{\partial W_{oij}} \tag{12}$$

where ($i = 1, \ldots, N_H, j = 1, \ldots, N_o$) and η is a positive real number representing the learning rate.

The change for any specified element b_{ok} of the bias matrix can is determined as follows:

$$\Delta b_{oj}(t+1) = -\eta \frac{\partial J(t+1)}{\partial b_{oj}} \tag{13}$$

where ($j = 1, \ldots, N_o$). The initial values of W_{oij} are set to zero and the initial values of b_{oj} are given randomly.

4.4. Regularised SMIA network (R_SMIA)

In this section, the regularisation technique has been introduced in order to improve the performance of the SMIA network. Regularisation is the technique of adding a penalty term Ω to the error function which can help obtaining a smoother network mappings. It is given by:

$$\tilde{A} = E + \lambda\Omega \tag{14}$$

where *E* 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 $\tilde{A}$ (Bishop, 1995). One form of regularisation is called weight decay. This form is based on the sum of the squares of the adaptive parameter in the network.

$$\Omega = \frac{1}{2}\sum_i W_i^2 \tag{15}$$

Although the use of weight decay in some cases leads to degraded performance of the network, it has been proven in most cases that it can avoid the overfitting problem and as a result enhance the network performance (Duda, Hart, & Stork, 2000).

The reason behind the popularity of weight decay approach is the simplicity of using this method. The idea is that every weight once updated, is simply decayed or shrunk as follows:

$$W^{\text{new}} = W^{\text{old}}(1-\lambda) \tag{16}$$

where $0 < \lambda < 1$. The weight decay is performed by adding a bias term to the original objective function E, thus the weight decay cost function is determined as follows (Bishop, 1995):

$$Ewd = E + (\lambda/2)B \tag{17}$$

where λ is the weight decay rate, B represents the penalty term.

The simplest form of calculating the penalty term B is:

$$B = \sum W_{ij}^2 \tag{18}$$

where W_{ij} is the weight connections between the ith units and jth nodes in the next layer. In R_SMIA network, the weight decay was used to adjust the weights between the hidden nodes and output units. The change of weights using weight decay method could be calculated as follows:

$$\Delta W_{ojk} = -\eta \frac{\partial E}{\partial W_{ojk}} = -\eta \frac{\partial}{\partial W_{ojk}} \left(E_{\text{std}} + \frac{\lambda}{2} \sum W_{ojk}^2 \right) \tag{19}$$

$$\Delta W_{ojk} = \eta \left(\sum e\grave{f}_{ot} f_{ot} - \lambda \sum W_{ojk} \right) \tag{20}$$

where ΔW_{ojk} is the updated weights between hidden units and output unit. The R_SMIA network is used to examine the effect of the regularisation technique and to enhance the performance of the SMIA network in the prediction.

5. Prediction of financial signals

Ten noisy financial time series signals are considered as shown in Table 1. All the signals were obtained from a historical database provided by Datastream®, forepart from the IBM common stock closing price time series, which was taken from the Time Series Data Library (Datastream, 2005). The networks are tested for the prediction of one- and five-steps-ahead predictions of financial time series in which two methods are utilised; in the first method, the data are passed directly to the neural network as non-stationary signals; while in the second method, the financial data are transformed into stationary signals.

For non-stationary signals, the data are presented to the networks directly without any transformation. The data are scaled between the upper and lower bounds of the transfer function. On the other hand, the stationary version of the signals needs some series of transformations before passing them to the networks. For the stationary signals, we systematically investigate a method of pre-processing the financial signals in order to reduce the influence of their trends. To smooth out

Table 1. Financial 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 Ind. 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

Table 2. Calculations for input and output variables

	Indicator	Calculations
Input variables	EMA15	$P(i) - \overline{\text{EMA}_{15}(i)}$
	RDP-5	$(p(i) - p(i-5))/p(i-5) \times 100$
	RDP-10	$(p(i) - p(i-10))/p(i-10) \times 100$
	RDP-15	$(p(i) - p(i-15))/p(i-15) \times 100$
	RDP-20	$(p(i) - p(i-20))/p(i-20) \times 100$
Output variables	RDP+5	$(\overline{p(i+5)} - \overline{p(i)})/\overline{p(i)} \times 100$
		$\overline{p(i)} = \overline{\text{EMA}_3(i)}$

Notes: $EMA_n(i)$ is the n-day exponential moving average of the ith day.
$p(i)$ is the closing price of the ith day.

the noise and to reduce the trend, the original raw data was pre-processed into a stationary series by transforming them into measurements of relative difference in percentage of price (RDP) (Thomason, 1999). The calculations for the transformation of input and output variables are presented in Table 2. Subsequent to transformation, all the input and output variables in Table 2 were scaled between the upper and lower bounds of the transfer function in order to avoid computational problems and to meet algorithm requirements.

6. Training the networks

The performances of the SMIA and the R_SMIA are benchmarked against the performance of MLP, the regularised MLP (R_MLP) and the FLNN network. Early stopping was utilised and each signal was divided into three data-sets which are the training, validation and the out-of-sample data which represent 25, 25 and 50% of the entire data-set, respectively. For FLNN, the higher order terms were empirically selected between two and five. The MLP were trained with hidden units varies from three to eight. The prediction performance of all networks was evaluated using three financial metric (Dunis & Wiliams, 2002), where the objective was to use the networks predictions for profit purpose, and three statistical metrics (Cao & Tay, 2003) which provide accurate tracking of the signals, as shown in Table 3.

7. Simulation results

As we are concerned with financial time series prediction, in these extensive experiments, our primary interest is to concentrate on the profitable value contained in the predictions for all neural network models. For this reason, the neural networks structure, which provides the highest percentage of the AR on out-of-sample data, is considered to be the best model. Tables 4–7 summarise the average results of 50 simulations obtained on out-of-sample data for the prediction of both stationary and non-stationary signal, when used to predict one- and five-steps-ahead predictions.

7.1. One-step-ahead prediction (stationary)

For the AR, the simulation results indicated that the R_SMIA network has outperformed the MLP and R_MLP prediction for all the ten stationary signals. Conversely, the R_SMIA set of results shows lowest profits when compared with FLNN. While the R_SMIA network outperformed the SMIA network for forecasting all the signals apart from the JPY/USD exchange rate and the DJIAO stock opining.

Although using the regularisation technique with the standard MLP network results in an improvement in the performance of the R_MLP, the SMIA network has shown the highest profit in all 10 series data than the R_MLP network except for the USD/EUR.

It could be observed that the results of the maximum drawdown demonstrate higher values were obtained using the R_SMIA network when used to predict the USD/UKP, NASDAQO, NASDAQC and OIL time series. The FLNN produced better results in comparison to multilayer networks for the remaining time series.

Table 3. Performance metrics and their calculations

Annualised return (%AR)	Normalised mean squared error (NMSE)
$AR=\frac{\text{Profit}}{\text{All profit}}\times 100$ $\text{Profit}=\frac{252}{n}\times \text{CR},\quad \text{CR}=\sum_{i=1}^{n}R_i$ $R_i=\begin{cases}+\lvert y_i\rvert & \text{if}\,(y_i)(\hat{y}_i)\geq 0,\\ -\lvert y_i\rvert & \text{otherwise}\end{cases}$ $\text{All profit}=\frac{252}{n}\times\sum_{i=1}^{n}abs(R_i)$	$\text{NMSE}=\frac{1}{\sigma^2 n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$ $\sigma^2=\frac{1}{n-1}\sum_{i=1}^{n}(y_i-\bar{y})^2$ $\bar{y}=\sum_{i=1}^{n}y_i$
Maximum drawdown (MD)	Signal to noise ratio (SNR)
$\text{MD}=\min\left(\sum_{t=1}^{n}(CR_t-\max(CR_1,\ldots,CR_t))\right)$ $CR_t=\sum_{i=1}^{t}R_i,\,t=1,\ldots,n$ $R_i=\begin{cases}+\lvert y_i\rvert & \text{if}\,(y_i)(\hat{y}_i)\geq 0,\\ -\lvert y_i\rvert & \text{otherwise}\end{cases}$	$\text{SNR}=10\times\log_{10}(\sigma)$ $\sigma=\frac{m^2\times n}{\text{SSE}}$ $\text{SSE}=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$ $m=\max(y_i)$
Annualised volatility (VOL)	Correct directional change (CDC)
$\text{VOL}=\sqrt{252}\times\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(R_i-\bar{R})^2}$	$\text{CDC}=\frac{1}{n}\sum_{i=1}^{n}d_i$ $d_i=\begin{cases}1 & \text{if}\,(y_i-y_{i-1})(\hat{y}_i-\hat{y}_{i-1})\geq 0,\\ 0 & \text{otherwise}\end{cases}$

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

Nevertheless, the R_SMIA networks outperformed all multilayer networks in most of the time series. For the volatility, the comparison between the multilayer networks clearly represents that the R_SMIA has the lower values than the other networks except for the prediction of the JPY/USD and DJIAO time series as the values slightly rising. However, the FLNN produces lower volatility than all other networks for predicting all the 10 signals.

When evaluating the Sharpe Ratio (SR) measure, it can be noticed that higher value is preferable. Table 5 indicated that the FLNN provides the best SR.

Figure 3 shows the value of the AR which has been forecasted by all networks used in this research.

In order to compare the rate of the weight decay (decay rate), that were utilised in the prediction of the R_MLP and R_SMIA networks, Figure 4 represents the best decay rate used in this experimental work.

7.2. Five-step-ahead prediction (stationary)

The simulation results indicated that using eight hidden nodes in the MLP and R_MLP network can produce the best average of profits. While four order FLNN model can obtain the highest profits. The simulation results for the prediction of the exchange rate time series using the percentage of AR indicated that the SMIA network outperforms the MLP and the FLNN models by 0.38–10.47%, respectively. These results show that the SMIA network made the best profits on average for all exchange rate data signals when compared to MLP and FLNN networks.

Table 4. Average results on stationary signals for the prediction of one-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
	SMIA	73.15769	76.19371	71.00268	62.66176	63.35985	63.71905	62.29099	67.01270	68.02663	72.16195
	R_MLP	69.87597	69.11974	72.43294	61.31236	63.32054	60.396	57.27957	53.52269	56.41693	54.96625
	R_SMIA	73.63894	75.08265	74.79281	63.4887	66.11718	62.14083	63.69784	69.75802	69.74614	73.578
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
	SMIA	−1.51535	−1.24388	−2.69518	−8.84982	−8.32449	−6.60512	−7.99077	−3.55542	−4.17998	−8.32751
	R_MLP	−1.14692	−1.38279	−1.3342	−4.58009	−5.12025	−2.82338	−2.98009	−3.85035	−3.54951	−17.3305
	R_SMIA	−1.14585	−1.37047	−1.24481	−4.19713	−4.14573	−3.10446	−5.62573	−3.29048	−3.3862	−6.76682
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
	SMIA	4.346859	5.424529	4.63746	13.20117	12.53349	11.00119	11.12426	11.93661	11.94297	60.15546
	R_MLP	4.434009	5.621507	4.604267	13.23235	12.53594	11.1544	11.34621	12.51297	12.45137	65.8069
	R_SMIA	4.334805	5.457777	4.539692	13.15954	12.37758	11.07906	11.06588	11.79905	11.84642	59.63936
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
	SMIA	16.83866	14.04827	15.34295	4.742449	5.057593	5.800506	5.610039	5.614729	5.678424	1.20381
	R_MLP	15.76153	12.30291	15.73393	4.649596	5.053141	5.416199	5.051273	4.28039	4.531404	0.857579
	R_SMIA	16.98832	13.75739	16.47533	4.82642	5.341885	5.610178	5.760461	5.912536	5.888317	1.234447
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
	SMIA	21.93	23.41	23.04	24.14	22.75	23.61	23.53	25.02	25.03	22.2
	R_MLP	21.64	22.5	23.38	23.68	23.22	22.98	23	24.27	24.5	21.46
	R_SMIA	22.28	23.4	23.42	24.59	23.18	23.97	23.95	25.13	25.12	23.23
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
	SMIA	66.09	62.35	63.07	66.82	62.39	61.92	62.98	60.48	59.95	62.76
	R_MLP	68.42	63.74	62.78	67.07	65.07	56.24	66.01	64.33	64.67	61.33
	R_SMIA	66.74	62.46	61.54	68.73	62.7	63.6	63.84	60.7	61.1	61.73

Table 5. Average results on stationary signals for the prediction of five-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
	SMIA	88.15049	87.17208	91.84363	85.16167	85.02744	85.3598	84.89345	81.28562	86.66031	91.82156
	R_MLP	87.1223	83.74337	90.66374	77.94086	84.49206	74.35885	69.99318	76.14721	75.9389	81.74823
	R_SMIA	90.07944	86.87133	91.62076	85.4554	86.29109	83.8411	85.13061	86.84489	86.55567	91.28559
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
	SMIA	−3.2779	−2.72278	−1.39054	−6.19597	−7.36317	−8.41446	−8.37277	−7.78359	−3.70212	−12.60818
	R_MLP	−2.45425	−2.72277	−2.555499	−9.173755	−7.024514	−8.166544	−10.711946	−12.706323	−13.174437	−57.254185
	R_SMIA	−1.77722	−2.72277	−1.35824	−6.3989	−7.08089	−7.00081	−6.72123	−3.4824	−4.75658	−13.3041
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
	SMIA	15.87492	16.76482	15.6362	35.93796	36.06489	31.63171	31.78481	37.03390	36.27336	157.372
	R_MLP	16.01374	17.240726	15.812136	37.578259	36.195788	33.752804	34.536964	37.978064	38.347615	180.79057
	R_SMIA	15.64255	16.80812	15.67015	35.86336	35.73558	31.95809	31.74464	35.92529	36.2965	158.8773
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
	SMIA	5.561351	5.199892	5.873841	2.369793	2.357881	2.633558	2.672691	2.564821	2.389255	0.584465
	R_MLP	5.441671	4.857615	5.735123	2.079828	2.335423	2.20523	2.032267	2.005844	1.98104	0.46131
	R_SMIA	5.758673	5.168465	5.846852	2.382872	2.414784	2.623515	2.681791	2.417534	2.384731	0.575535
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
	SMIA	24.98	23.02	23.41	25.22	24.33	24.81	24.68	26.28	27.27	25.37
	R_MLP	26.59	23.46	25.87	25.27	25.5	24.04	23.77	25.07	24.94	22.02
	R_SMIA	25.68	23.5	23.61	23.82	24.67	24.99	24.99	27.42	27.03	25.36
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
	SMIA	65.49	63.16	65.09	62.49	59.48	62.91	62.99	61.72	62.97	60.06
	R_MLP	66.02	63.58	64.57	61.52	60.69	62.12	62.3	62.05	61.1	54.7
	R_SMIA	66.16	64.07	65.02	61.59	59.75	62.31	62.86	62.59	61.98	60.03

Table 6. Average results on non-stationary signals for the prediction of one-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
	SMIA	2.073759	−8.271959	2.414358	−3.619703	3.531254	−15.34971	−15.34804	−5.068463	−5.201504	24.689167
	R_MLP	0.470592	−1.08558	13.39764	−5.15092	−2.75681	−2.12597	−5.46303	−7.75481	−8.15845	9.532212
	R_SMIA	13.222424	0.109788	10.548035	5.048035	−1929973	−1.448578	−3.431984	−7.377525	−0.69999	24.824701
MD	MLP	−14.957188	−22.500975	−11.293929	−15.470936	−70.907413	−63.928973	−73.355125	−67.988187	−67160675	−13.563781
	FLNN	−18.341226	−21.509273	−10.26734	−46.467443	−51.820357	−49.59282	−66.6875	−51.65052	−51.482152	−100.89227
	SMIA	−19.83157	−32.49779	−13.49779	−64.55476	−49.69712	−88.91595	−92.87567	40.012786	−45.16187	−50.27416
	R_MLP	−22.729432	−20.852478	−8.21238	−49.749368	46.533708	−36.2518	−48.166501	−52.472487	−55.995268	−70.085349
	R_SMIA	−11.377406	−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
	SMIA	11.399141	12.77795	10.105815	30.733801	30.472487	27.361664	27.773873	29.9737	30.642176	128.19774
	R_MLP	11.418611	12.804681	10.077155	30.774937	30.59627	27.56866	27.96083	29.985878	30.625883	130.31572
	R_SMIA	11.370933	12.807689	10.084079	30.777562	30.597727	27.580758	27.984762	29.992723	30.664225	126.72161
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
	SMIA	0.182488	−0.64743	0.239181	−0.1177	0.115987	−0.56231	−0.55418	−0.16912	−0.16979	0.193607
	R_MLP	0.041302	−0.084872	1.329773	−0.16743	−0.090141	−0.077248	−0.195888	−0.258771	−0.266283	0.074606
	R_SMIA	1.163817	0.00857	1.046945	0.164053	−0.6314	−0.05255	−0.12285	−0.24607	−0.02314	0.196018
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
	SMIA	20.5	27.04	13.97	19.83	16.05	16.97	18.51	16.51	16.88	11.44
	R_MLP	15.04	19.38	16.23	20.86	21.62	13.56	13.77	17.44	17.81	10.88
	R_SMIA	16.4	20.34	11.51	15.17	17.07	16.88	15.65	14.3	13.91	11.09
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
	SMIA	50.31	45.45	52.79	47.77	50.55	46.1	45.56	48.16	47.63	57.1
	R_MLP	50.48	46.06	53.37	47.42	49.8	50.6	49.58	47.64	46.55	52.29
	R_SMIA	53.43	47.92	52.33	51.45	50.55	50.25	49.62	47.28	51.48	65

Table 7. Average results on non-stationary signals for the prediction of five-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
	SMIA	−1.48508	2.839703	2.819774	−5.12982	−1.00699	2.440176	1.880855	2.857381	1.735058	−6.13704
	R_MLP	−3.09729	−3.309134	3.733809	−4.589262	−1.772365	−4.65767	−4.517743	−3.079075	−3.133686	0.113113
	R_SMIA	1.79951	2.034985	3.813749	−1.92625	−3.73281	−1.96245	−1.52512	0.054411	2.001813	−1.24209
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
	SMIA	−17.776	−19.8414	−8.63958	−38.6083	−32.3624	−20.558	−22.1424	−35.7076	−34.5416	−90.0799
	R_MLP	−16.877584	−19.066487	−8.871813	−39.692504	−39.37541	−42.126048	−14.89301	−49.765621	−25.975362	−82.687761
	R_SMIA	−15.08815	−14.67284	−11.57839	−39.49456	−40.71476	−40.93965	−37.91129	−48.78379	−44.45037	−76.71338
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
	SMIA	11.41287	12.7995	10.11326	30.77346	30.59728	27.58149	28.00402	29.9849	30.67759	130.3251
	R_MLP	11.43124	12.807072	10.130352	30.813032	30.635332	27.587913	28.014315	30.050515	30.717461	131.81674
	R_SMIA	11.42942	12.8153	10.13094	30.82329	30.62114	27.59616	28.01929	30.06198	30.72487	132.1194
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
	SMIA	−0.13091	−0.22183	0.378768	−0.16675	−0.1253	0.088481	0.067178	0.095616	0.056716	−0.04803
	R_MLP	−0.271036	−0.258545	0.368653	−0.148991	−0.057879	−0.168913	−0.161395	−0.102547	−0.10209	0.000935
	R_SMIA	0.157756	0.159264	0.376467	0.062527	−0.1219	−0.07122	−0.05443	−0.001792	−0.065155	−0.00939
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
	SMIA	16.83	21.66	13.32	19.94	17.1	16.83	17.14	13.38	14.88	10.92
	R_MLP	15.85	16.15	13.41	20.5	20.61	13.31	13.41	17.32	17.63	10.82
	R_SMIA	14.11	16.19	11.37	16.13	16.56	14.9	12.47	17.15	16.52	10.99
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
	SMIA	53.2	50.98	51.89	48.05	48.41	52.54	52.36	50.15	49.83	49.03
	R_MLP	54.27	48.69	51.38	48.22	51.01	51.87	51.09	49.58	49.05	49.71
	R_SMIA	51.4	46.34	51.93	47.64	51.41	51.19	51.91	49.74	49.67	52.61

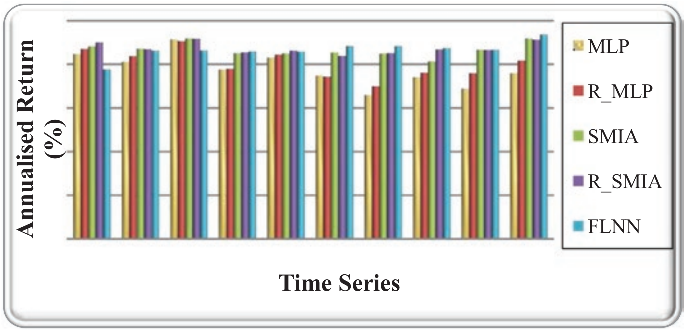


Figure 3. The average of AR predicted from all networks.

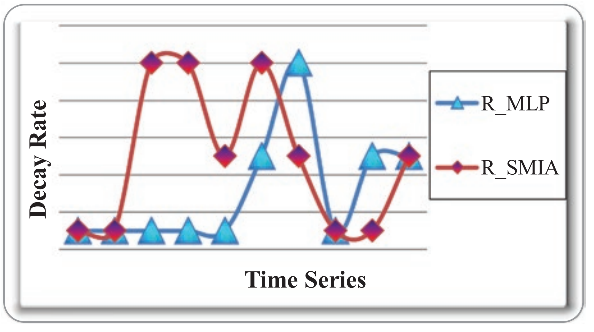


Figure 4. Best decay rate used in prediction of all financial signals.

The comparison between the performance of the SMIA network and the R_SMIA network based on the percentage of AR detect an increasing on profits obtained with R_SMIA network. The R_SMIA successfully reaches the highest profits than SMIA network when forecasting the following five financial time series: USD/UKP, NASDAQO, NASDAQC, DJIAC and DJUAO.

The overall performances of the five networks which are utilised in forecasting the various signals using the AR is depicted in Figure 5. The five-steps-ahead prediction for all networks indicated that the SMIA and R_SMIA networks produce better percentage of AR than the other multilayer networks. Meanwhile, it complements the FLNN in some stock prices data.

For the value of the decay rate, it can be noticed from Figure 6 that the signals can reach the best ratio of profits by using small values of decay rate (which is 0.0001) when predicting the five-step-ahead prediction for R_MLP and R_SMIA networks.

7.3. One-step-ahead prediction using non-stationary signals

The number of hidden units or network order used to obtain the best prediction showed that the performance of MLP network produces the best results of profits using six or eight hidden nodes

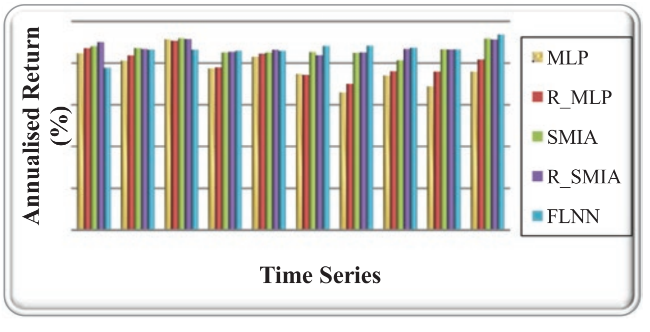
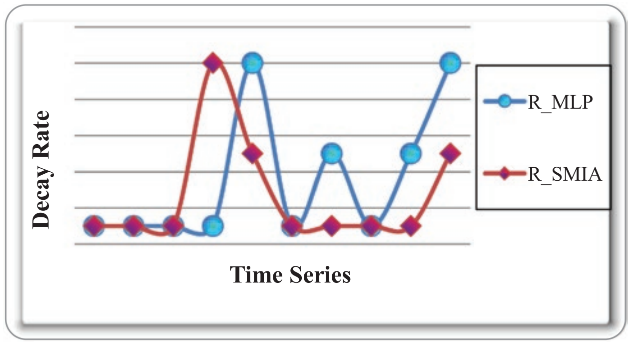


Figure 5. The average of AR predicted from all networks.

Figure 6. Best decay rate used in the prediction of all financial signals (five-steps ahead).



while the R_MLP network gives better profits using seven or eight hidden nodes. Furthermore, the SMIA and R_SMIA networks could reach high values of profits using seven or eight hidden units and above. The FLNN reaches the best performance when using only the third order in most cases.

For the AR, the R_SMIA shows higher values than all other network for the USD/UKP, JPY/USD, NASDAQO, DJIAO, DJIAC, DJUAC and OIL time series. The SMIA network achieved the highest profit on two signals the NASDAQC and the DJUAO signals. Meanwhile the R_MLP can obtain the best average of profit only when it is used to forecast the USD/EUR signal. Figure 7 illustrates the performance of the AR for the forecast of the five network models that are used in this research work, while Figure 8 shows the rate decay values which were used for the prediction of all data signals.

Figure 7. The best average of AR predicted from all networks.

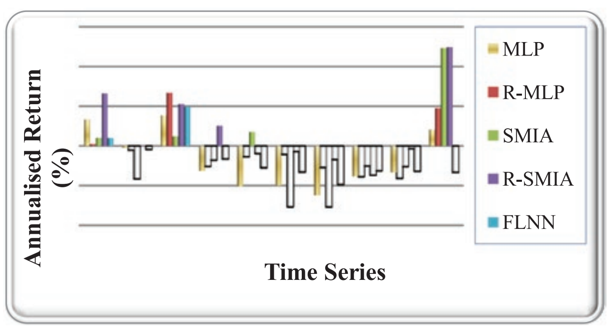
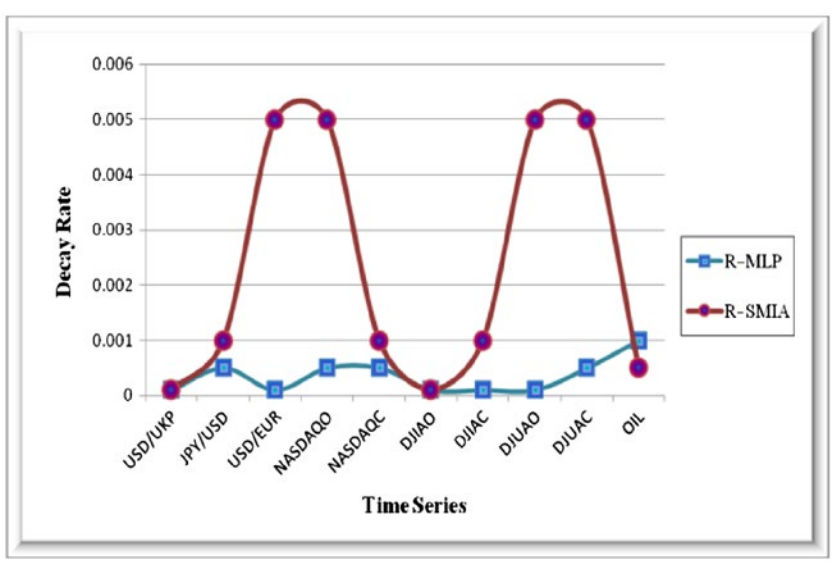


Figure 8. Best decay rate used in prediction of all financial signals.



7.4. Five-step-ahead prediction using non-stationary signals

Although the prediction for the non-stationary signals usually give inconsistent results, the extensive experiments of this research proved that the proposed application of the SMIA and R_SMIA for the prediction of financial time series showed the best profit values when compared to other neural networks.

The MLP and R_MLP networks can produce the best average results of profits with seven or eight hidden nodes. The SMIA network gives the best results using only five or seven hidden units. However, the R_SMIA network attains the highest percentage of AR with four hidden nodes and above. For the FLNN most prediction results indicated that the best profits can be achieved using two or three network order.

The comparison between all networks demonstrated that the high ratio of the AR is achieved using the SMIA and R_SMIA networks. Meanwhile, for the MLP and R_MLP networks, each network can attain higher profit value for only one signal namely NASDAQC and OIL signals, respectively. Furthermore, the FLNN produced the worst profits in comparison to the multilayer networks.

Figure 9 shows the values of the AR for the prediction of the various networks. The simulation results indicated that the SMIA and R_SMIA networks produced better percentage of AR than the other networks in most cases. Figure 10 represents the best decay rate values that are used for the R_SMIA and R_MLP neural networks.

Figure 9. The average of AR predicted from all networks.

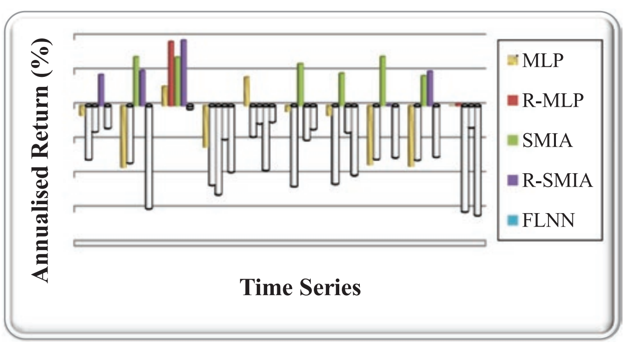


Figure 10. Best decay rate used in prediction of all financial signals.

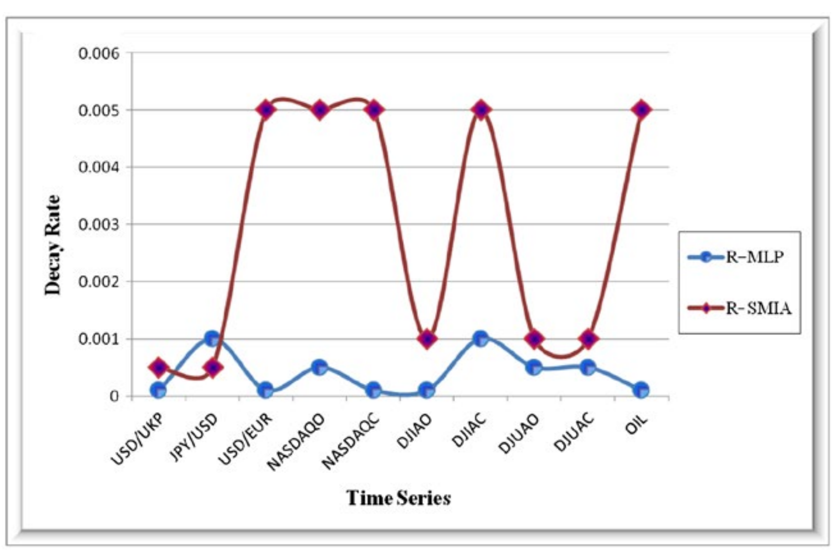


Table 8. The standard deviation for the exchange rate between the UK/US Dollar time series over 50 simulations with respect to the profit value

Network	One-step-ahead prediction using stationary data	Five-steps-ahead prediction using stationary data	One-step-ahead prediction using non-stationary data	Five-steps-ahead prediction using non-stationary data
SMIA	0.4709	0.3538	6.5302	7.1744
R_SMIA	0.1795	0.2654	5.9049	6.4938
MLP	13.7166	6.9672	7.4127	7.7695
R_MLP	1.1148	2.6221	3.2671	4.0839

8. Discussion

Simulation results demonstrate that all the neural networks models used in this research work were potentially profitable, the non-stationary financial signals are very difficult to predict due to its instability behaviour. The non-stationary signals are highly volatile and noisy and that is why they often change their behaviour and fall sharply at some point during the training. The networks are trying to learn the price values of the non-stationary signals during the training phase where they are unable to respond well, since the prices values include high-frequency components. Therefore, the networks generate unpromising prediction using the AR measure.

For the stationary signals, the networks predicted high percentage of profits. The non-stationary signals are smoothed and transferred into Relative Different in Price (RDP) and the neural networks generate better forecasting and profit. Consequently, neural networks can attain stable prediction and higher profits for stationary signals than the non-stationary signals.

In this research study, six stock opining prices and stock closing prices time series data have been used which includes NASDAQO, NASDAQC, DJIAO, DJIAC, DJUAO and DJUAC. Three of these time series are stock opening prices and the others are stock closing prices. The aim of these signals is to investigate the differences between the predictions of the opening stock price and closing stock results.

For stationary signals, the simulation results showed that for all networks used in this work, there is a slightly differences in the results when using these signals in one-step-ahead prediction. While the prediction results for five-steps-ahead illustrate variances between these series.

The non-stationary signals show that in most cases the prediction results for one-step-ahead and five-steps-ahead have small difference between the opening and closing stock prices for all networks which have been utilised in the current work.

It is worth to notice that these differences related to the raw data, since the data are affected by several factors such as the threats of war, good or bad economic climate, announcements of company earning and the advertisements of economic statistics.

As it can be noticed from Table 8, the simulation results indicated for the prediction of the US/UK exchange rate time series that the standard deviations for the SMIA, R_SMIA, MLP and the R_MLP have significantly different values which indicate that the results achieved by each network is strategically different.

9. Conclusion

This research work underlines an important contribution of a new application of the self-organised multilayer neural network inspired by the SMIA for the prediction of the financial time series; namely, its elegant ability to approximate non-linear financial time series. The network has shown its advantages in forecasting both stationary and non-stationary signals. A considerable profitable value does exist in the proposed network when compared to other networks and the network demonstrated a vast speed in convergence time. Hence, it is anticipated that the self-organised multilayer neural network inspired by the SMIA can be used as an alternative method for predicting financial variables and thus justified the potential use of this model by practitioners. The superior property hold by the network could promise more powerful applications in many other real world problems.

Funding
The authors received no direct funding for this research.

Author details
Dhiya Al-Jumeily[1]
E-mail: d.aljumeily@ljmu.ac.uk
Abir J. Hussain[1]
E-mail: a.hussain@ljmu.ac.uk
[1] Faculty of Engineering and Technology, Applied Computing Research Group, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK.

Citation information
Cite this article as: The performance of immune-based neural network with financial time series prediction, D. Al-Jumeily & A.J. Hussain, *Cogent Engineering* (2015), 2: 985005.

References

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Cass, R., & Radl, B. (1996). Adaptive process optimization using functional-link networks and evolutionary optimization. *Control Engineering Practice, 4*, 1579–1584. http://dx.doi.org/10.1016/0967-0661(96)00173-6

Cao, L. J., & Tay, F. E. H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks, 14*, 1506–1518. http://dx.doi.org/10.1109/TNN.2003.820556

Datastream International Limited all rights reserved/ Datastream database, [machine-readable data]. (2005). London: Thompson Financial Limited [Producer and Distributor]. Retrieved November 14, 2005 from http://financial.thomsonreuters.com/en.html

Dorffner, G. (1996). Neural networks for time series processing. *Neural Network World, 6*, 447–468.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd ed.). New York, NY: Wiley-Interscience.

Durbin, R., & Rumelhart, D. E. (1989). Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Computation, 1*, 133–142. http://dx.doi.org/10.1162/neco.1989.1.1.133

Dunis, C. L., & Wiliams, M. (2002). Modeling and trading the UER/USD exchange rate: Do neural network models perform better? *Derivatives Use, Trading and Regulation, 8*, 211–239.

Fama, E. F., & French, K. R. (1989). Business conditions and expected returns on stocks and bonds. *Journal of Financial Economics, 25*, 23–49. http://dx.doi.org/10.1016/0304-405X(89)90095-0

Fama, E. F., & Schwert, G.W. (1977). Asset returns and inflation. *Journal of Financial Economics, 5*, 115–146. http://dx.doi.org/10.1016/0304-405X(77)90014-9

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, 2243*, 508–515.

Ferson, W. (1989). Changes in expected security returns, risk, and the level of interest rates. *The Journal of Finance, 44*, 1191–1217. http://dx.doi.org/10.1111/j.1540-6261.1989.tb02650.x

Giles, C. L., & Maxwell, T. (1987, December). Learning, invariance and generalisation in high-order neural networks. *Applied Optics, 26*, 4972–4978.

Jensen, M. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics, 6*, 95–101. http://dx.doi.org/10.1016/0304-405X(78)90025-9

Kaita, T., Tomita, S., & Yamanaka, J. (2002). On a higher-order neural network for distortion invariant pattern recognition. *Pattern Recognition Letters, 23*, 977–984. http://dx.doi.org/10.1016/S0167-8655(02)00028-4

Lawrence, S., & Giles, C. L. (2000). Overfitting and neural networks: Conjugate gradient and backpropagation. In *International Joint Conference on Neural Network, Italy* (pp. 114–119). Los Alamitos, CA: IEEE Computer Society.

Lendasse, A., de Bodt, E., Wertz, V., & Verleysen, M. (2000). Non-linear financial time series forecasting—Application to the Bel 20 stock market index. *European Journal of Economic and Social Systems, 14*, 81–91. http://dx.doi.org/10.1051/ejess:2000110

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE, 63*, 561–580. http://dx.doi.org/10.1109/PROC.1975.9792

Manikopoulos, C. N. (1992). Neural network approach to DPCM system design for image coding. *IEE Proceedings-I, 139*, 501–507.

Mirea, L., & Marcu, T. (2002). System identification using functional-link neural networks with dynamic structure. In *15th Triennial World Congress*, Barcelona, Spain.

Thimm, G. (1995). *Optimisation of high order perceptron*. Lausanne: Swiss federal Institute of Technology (EPFL).

Thomason, M. (1999). The practitioner method and tools. *Journal of Computational Intelligence in Finance, 7*, 36–45.

Timmis, J. I. (2001). *Articial immune systems: A novel data analysis technique inspired by the immune network theory*. Aberystwyth: University of Wales.

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*, 72–84. http://dx.doi.org/10.1016/j.asoc.2004.10.008

Yao, J., & Tan, C. L. (2000). A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing, 34*, 79–98. http://dx.doi.org/10.1016/S0925-2312(00)00300-3

Zhang, G. P. (2003). *Neural networks in business forecasting*. Hershey, PA: Idea. http://dx.doi.org/10.4018/978-1-59140-176-6