*International Journal of Engineering & Technology*, 7 (4.11) (2018) 99-105

**International Journal of Engineering & Technology**

*Website: www.sciencepubco.com/index.php/IJET*

***Research paper***

# Effectiveness of Artificial Neural Networks in Solving Financial Time Series

**Marwan Abdul Hameed Ashour[1] , Arshad Jamal[2], Rabab Alayham Abbas Helmi[3]***

*College of Administration and Economics, University of Baghdad, Iraq*
*[2]Faculty of Information Sciences and Engineering, Management and Science University, Malaysia*
*[3]Faculty of Information Sciences and Engineering, Management and Science University, Malaysia*
**Corresponding author E-mail: rabab_alayham@msu.edu.my*

## Abstract

This research aims to study and analyze which type of Artificial Neural Network (ANN) is more efficient and suitable in handling non-homogenous variance for financial series. Apart from addressing the behavior and efficiency of ANN, the paper also aims to present an advanced methodology for ANN, as a replacement of GARCH and ARCH models in crisis management decision makers. The application part was applied to the Egyptian exchange market, to study the local currency exchange rate volatility (1/1/2009-4/6/2013) in order to develop a model describing those changes in the exchange rate. The research concludes that the best network type to represent such financial series is the Back Propagation. Moreover, comparing the result with general regression and probabilistic networks rendered the later two inefficient at handling such series.

***Keywords***: *Back propagation; general regression network; GARCH model; Heteroscedastic variance financial series; neural networks.*

## 1. Introduction

The financial markets are important pillars of an economy in any country. In the Middle East there is a number of financial markets, the Egyptian financial market has a key importance in the region, on which the international financial crisis had its impact. The current instability in the region had its toll on the Egyptian financial market as well. In order to handle such situations that take place in the financial markets, there is a need for a model that can describe such anomalies and find a solution for them. For the model to be suitable and correct, it needs to take into account the changes that take place over a time period. This research addresses ANN due to the increase in its importance in general and in handling time series data in specific, as well as addressing GARCH and ARCH models. The research studies exchange rate fluctuations for the Egyptian financial market for the period between 1/1/2009 and 4/6/2013, in which the market was under the effect of the political situation in the region as well as the effect of the global financial crisis.

## 2. Methodology

### 2.1. ARCH and GARCH Models

The study of time series is based on analyzing a set of data, in order to study the characteristics of the data and conduct meaningful statistics for the prediction of future values for the series [1]. There are two methods in analyzing time series, the first method is frequency-domain method which is based on Fourier Transfer. While the second method is the time-domain which uses the verification of the autocorrelation of the series, those series are very beneficial in Box-Jenkins and ARCH/GARCH methods for time series prediction evaluation [2]. From the name Generalized Autoregressive Conditional Heteroscedastic (GARCH), the concepts of the method can be recognized. Autoregressive refer to the regressive method of feeding the series that includes the combination of past and present data. While Heteroscedastic refer to the change in the data throughout the time period, referring to the volatility of the variable through time. Therefore, the GARCH method includes the past volatility of the past to explain future volatilities. To be more specific, it is the time series technique that uses a multi volatility model. So, when a series is referred to as enjoying the GARCH effect, this means that the series is heteroscedastic and that the change or volatility defers through time. In the cases where the change is stable through time, the series would be homoscedastic [3, 4].

### 2.2. GARCH

Time series are considered one of the statistical methods used to analyze models based on historical data series. One of the most important of those models is the ARIMA models, which are used in various daily life related fields. The following random error conditions need to be met for the ARIMA model to be used.

$$i)\,E[\ ]_t\ [\ ]\ 0$$

$$ii)\,E[\ ]^2\ [\ ]^2$$

$$iii)\,E[\ ]_t[\ ]_s\ [\ ]0 \qquad for\ s\ [\ ]t$$

It is hard to meet the first two conditions in practical real world applications. Therefore, research to overcome this condition problem and improve the implementation of the model on the data as a result the GARCH model was reached as a solution for volatility problem in time series. The model was presented in 1982 as described by the following form:

$r_1 = \mu + a_t$
$a_t = \sigma_t \varepsilon_t;\ \varepsilon_t \sim \text{iidN}\ (0,1)$
$\sigma_t^2 = \alpha_0 + \alpha_1\ \alpha_{t-1}^2 + \alpha_2\ \alpha_{t-2}^2 + \ldots + \alpha_q\ \alpha_{t-q}^2$

In 1986, the autoregressive condition GARCH model was introduced by Bollersley by adding the limits of autoregressive for the model to be as follows:

$r_1 = \mu + a_t$
$a_t = \sigma_t \varepsilon_t;\ \varepsilon_t \sim \text{iidN}\ (0,1)$
$\sigma_t^2 = \alpha_0 + \alpha_1\ \alpha_{t-1}^2 + \alpha_2\ \alpha_{t-2}^2 + \ldots + \alpha_q\ \alpha_{t-q}^2 + \beta_1 \sigma_{t-1}^2 + \ldots + \beta_p \sigma_{t-p}^2$
$\sigma_t^2 = \alpha_0 \sum_{i=1}^{q} \alpha i\ \ \alpha t - i^{\,2} + \sum_{j=1}^{p} \beta t \sigma t - i^{\,2}$

with the use of

$$\sum_{i=1}^{q} \alpha i \quad + \sum_{j=1}^{p} \beta j \quad < 1$$

$\alpha_0 > 0$
$\alpha_i \geq 0\ \ i = 1,2,3\ldots, q$
$\beta j \geq 0\ \ j = 1,2,3\ \ldots, p$

The GARCH model becomes more rational in using the parameters and increase the calculation efficiency.

## 2.3. Artificial Neural Networks

The name of ANN refers to the network being a combination of an interconnected and communicating units, those unites were original inspired by the study of the nerves systems in living creatures [5]. The manner in which the ANN work is based on the idea of reaching a model through simulating the data in order to use that model for analysis, classification, estimation and other data processes, without the need for developing a model for the data. In other words, the no model is needed but rather a model is generated from the data. Therefore, ANN succeeded in gaining the attention of considerable number of scientists and researchers. This increasing attention and focus on ANN is due to its high flexibility in the data model learning process as well as inherited ease in storage and distribution of data, in comparison with other mathematical methods used in the same process [6].
In this sense, ANN can be defined as data processing system which is based on simple mathematical techniques, to achieve certain performance aspects in a way that is similar to that of a biological neural network like the nerve system. The ANN is considered one of the non-linear models [7].

## 2.4. Back Propagation Network

This type of ANN must have at least one back propagation cycle. At the same time, it could include one or more layer, the output from each neuron returns as input to all other remaining neurons. There is also an auto back propagation, meaning that the neuron output becomes input for the same neuron. This type of network is not largely used in affective domains. It is easier to achieve the same goals using front propagation networks [8, 9].

### 2.4.1. Error Back Propagation Methodology

The basic steps of the Back Propagation Methodology (BPM) is to calculate the error in the output layer, then use it to update the weightage for the hidden-output layer. Similarly, the error for the hidden layer is calculated to update weightage for the input-hidden layer. After that, the output for the network using the new weightage is calculated and the same process steps of calculating error and updating weightage are repeated until minimal error for the network is reached.

The purpose of training the ANN using BPM is to reach optimal weightage, which generates minimal error between the network output and the data model section criterion. So that, those weightage values can be used to estimate new data that the network has not used in previous training. During the first step of the network training, the initial weightage used in the beginning of the training process are random values generated from statistical distributions. Guaranteeing a minimal mean error square is one of the characteristics of using this method as well other characteristics such as its ability to deal with non-linear functions and systems.

### 2.4.2. Error Back Propagation Algorithm

Training the ANN using the error back propagation includes the three steps, forwarded stage, back pass stage and update weights stage. In the first stage, the forward stage, the input signal is distributed to every node in the hidden layer, and then the activation value is for each node is calculated. Those nodes will then send its signal to each node in the output layer. Then, the activation value for each node in the output layer is calculated, thus resulting in the ANN respond to the input sample given.
Throughout the training phase, each node in the output layer compare its calculated activation values with the actual output to find the error value for this node. Based on this error value, the error correction coefficient (δk) is calculated. This error correction coefficient is then used to distribute the error on the output layer nodes, only for the error to be send back to the previous layer. Similarly, the coefficient is used to adjust the weightage in the hidden layers. In a similar manner, the correction coefficient (δj) is calculated for every node in the hidden layer. This coefficient is used to calculate the weightage for the hidden layer. After finding all error correction coefficient the weightage for the whole network is conducted at the same time. The methodology can be described using the following steps:

1. Generating primary weightage – from one of the statistical distributions.
2. The input signal for each node in the input layer is received then sent to all the nodes in the hidden layer.
3. Each node in the hidden layer accumulates its weighted signal according to the following equation:

$$hj = 2 / (1 + \exp (\sum xiwij)) - 1$$

The activation equation is applied to estimate the output of the hidden layer, and then the activation values are sending to all nodes in the output layer.

4. Each node in the output layer accumulates its weighted signal according to the following equation:

$$yk = 2 / (1 + \exp (\sum hjwjk)) - 1$$

5. Calculate the error value for the output node by finding the difference between the activation values, or output values yk, and the target value, or actual value tk. This means:

$$E_k = t_k - y_k$$

where the output of the network is compared against the actual value to estimate the error through this equation:

$$\delta k = (t_k - y_k)\ .\ f(v)$$

where f(v) is the logistic equation when the non-linear output node is equal to 1 and the equation is linear. Then, the change in the error value is calculated using the following formula:

$$\Delta wjk = \alpha\ .\ \delta k\ .\ hj$$

6. All the nodes in the hidden layer, accumulate the signaled weighted input values to □ using the following equation:

$$\Delta j = \delta \sum_{j} kwjk$$

This value is then multiplied by the activation equation to calculate the $\delta_j$, then the change in the error value $\Delta v_{jj}$ using the following equation:

$$\Delta v_{jj} = \alpha \cdot \delta_j \cdot x_i$$

Update the weightage for all the nodes in the output layer, according to the following equation:

$$w_{jk}\,(new) = w_{jk}\,(old) + \Delta_{wjk}$$

Then, update the weightage for all the nodes in the hidden layer according to this equation:

$$v_{ij}\,(new) = v_{ij}\,(old) + \Delta v_{ij}$$

Repeat the steps of calculation, update and recalculate. In other words, the training process until optimal weightage is reached, thus the targeted outputs are reached. Meaning reaching the best compatibility with the model being processed, where the network input is xi and the weightage between the layers is W.

## 2.5. General Regression Neural Network

This type of ANN is considered a front feed neural networks. It was developed by Donald Specht 1991. For system modeling and diagnosis, it actual can be considered as the probabilistic ANN main-stream and is generally referred to using the short form GRNN. Apart from the general purpose of pattern classification for the probabilistic neural network, the GRNN have a lot of other different applications.

The basic idea behind developing General Regression Neural Network (GRNN) is based on the ability to approximate any function that is given inputs and data pair outputs. Let the function have (n) of inputs ($x_1, \ldots, x_n$) and one output (y). The value of the expected average output for a given input x, can be found using the following equation from probability theory:

$$\bar{y} = \frac{\int_{-\infty}^{\infty} y\,pdf(x,y)\,dy}{pdf(x)}$$

This equation contains the Probabilistic Density Function (pdf(x)), which is the joint probabilistic density function. Thus, meaning the probability of output becomes (y) and the probability of input becomes (x). Using the same method of probability density functions that approximated by accumulating Gaussian function in the probabilistic network PNN, therefore functions probabilistic condition can also have approximated. Last equation can be approximated using the following equation:

$$\bar{y} = \frac{\sum_{p=1}^{P} Y_p e^{-d^2}/2\sigma^2}{\sum_{p=1}^{P} e^{-d^2}/2\sigma^2}$$

In (15) assumes P data types in the training set.

The GRNN architecture is shown in Fig. 1, it is clear from the figure that the GRNN share great similarity with the Probabilistic Neural Network (PNN), except that in the first type the value of 1 is not assigned to weightage of the output layer. The GRNN is also shares similarities with the Radial Basis Function (RBF). The main difference is in the second "hidden" layer.

This type of network has two main stages in data processing, those are:

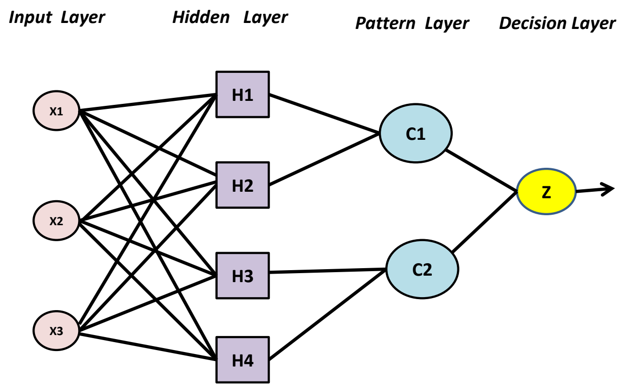


**Fig. 1:** GRNN Architecture

### 2.5.1. First Stage

In this stage, the first step is to find the distance between the input values, which are often a vector. Also, the values of the weights (which are often also a vector) are found which the second step is. Then, the result is multiplied by the gross value of bias, which is a small amount. The activation function is of Gaussian type, which is one of the non-linear functions that follow the formula:

$$f(x) = e^{-X^2}$$

### 2.5.2. Second Stage

This stage is called the special linear layer through which the result of the one-to-one multiplication of the input vector and the weightage vector that were the results of the previous stage. The multiplication result is than divided by the summation of the input values. The value of bias is not calculated and the conversion function for this stage is a linear function that takes the formula:

$$f(x) = x$$

Although there seem to be a lot of similarities between the two types, but there are a number of differences. Some of the basic functional differences are:

Firstly, in GRNN all the components of the training set are set and represented by a single vector. Therefore, the weightage between the input layer and hidden layer is part of the time series estimating component of training model. Secondly, the weightage in the GRNN do not undergo training. Moreover, there is no back feeding like in BPN to correct the weightage.

The GRNN and the probabilistic networks also differ from the BPN by its fast calculation. It also differs by overcoming one of the major problems in non-linear models parameters estimation, thus is local minima problem.

Neural networks are gaining increasing importance in processing and analyzing time series, due to its ability to self-learning and training. Most researchers use theorems based on shifting the time series by one degree or more, in order to find the input for the artificial neural network that is solving the time series.

## 2.6. Comparative Statistical Measures

Statistical measures are used to compare different methods, those measures are based on the random in the sense that error is difference between the estimated value and the actual value [8]. The different comparative statistical measures used are shown in Table 1:

$$MSE_f = \frac{1}{N} \sum_{i=1}^{N} (f_i \;\; O_i)^2$$

$$MSE_{Ref} = \frac{1}{N} \sum_{i=1}^{N} \bar{O} - O_i)^2$$

where $f_i$ is the estimated values and $O_i$ is the sample.

**Table 1:** Comparative Statistical Measures

| Statistical Maser | Mathematical Formula |
|---|---|
| Mean Absolute Percentage Error (MAPE) | MAPE = $\frac{\sum_{t=1}^{N} \lvert\frac{E_t}{Y_t}\rvert}{N}$ |
| Mean Squared Error (MSE) | MSE = $\frac{\sum_{t=1}^{N} E_t^2}{N}$ |
| Root Mean Squared Error (RMSE) | RMSE = $\sqrt{\frac{\sum_{t=1}^{N} \Sigma_t^2}{N}}$ |
| Forecast Skill (SS) | SS = 1 - $\frac{MSE_{forecast}}{MSE_{ref}}$ |
| Coefficient of Determination ($R^2$) | $R^2 = \frac{\Sigma\,(y(i) - \mu(y))^2}{\Sigma\,(\hat{Y}(i) - \mu(y))^2}$ |

As for the Forecast Skill (SS) in Table 1, the value has to be between zero and one (0-1). If the value is 1, then the forecast skill is perfect. On the other hand, if the value is 0, then the forecast skill is low. In other words, it has negative skill. Generally, ($0 < SS < 1$) and the closer the value gets to 1 then the forecast skill is higher, similarly the closer the number is to 0 then the lower the forecast skill is. When using the measures to compare among methods, the method that generates the lowest value of error and highest value of the mentioned coefficients is the best among the compared methods.

## 3. Results and Discussion

The research addresses the changes that take place in the Egyptian financial market, in order to develop a model to describe the changes in this financial market. Therefore, the Egyptian currency (Egyptian Pound) exchange rate for the period (1/1/2009 – 4/6/2013) was studied because of the changes that happened during this period in Egypt in general and the financial market in specific. Fig. 2 represents the Egyptian pound exchange rate time series over the mentioned period. It is clear from the figure that the series tend to generally decrease, as well as it showing instability.

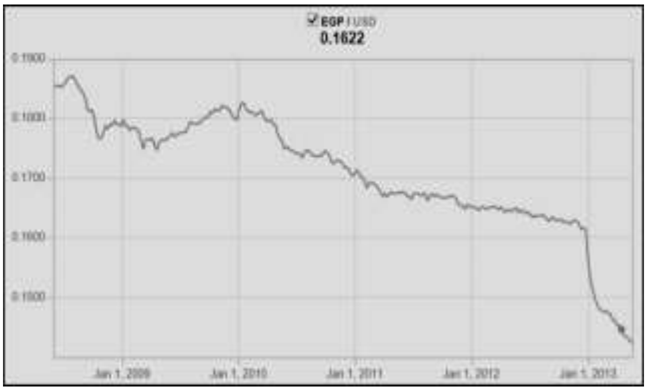

**Fig. 2:** Egyptian pound exchange rate time series

In order to analyze and process this time series, GARCH models and ANN is used. The results are as follow.

### 3.1. GARCH Model

In order to obtain a stable series, the time series is changed to the Retune Series as shown in Fig. 3. It is clear from the figure the retune series suffer from volatility, as well as the instability in the rate. The behavior for Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) for the retuned series is shown in Fig. 4.

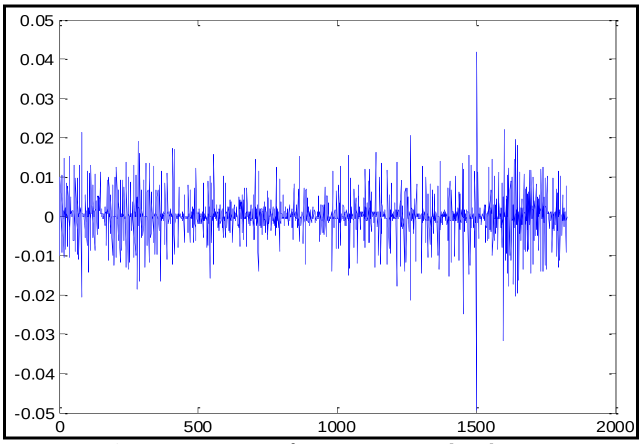

**Fig. 3:** Retune Series for Egyptian Pound exchange

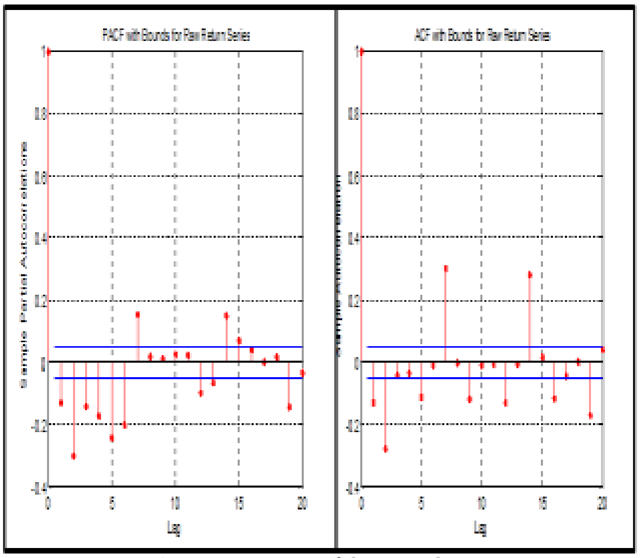

**Fig. 4:** ACF- PACF of the retuned series

From the behavior of the ACF and PACF shown in Fig. 4, a correlation in the retuned series. While, the behavior of the ACF for the squared retuned series in Fig. 5 shows a sequential correlation in more than one shift. In the same figure and from the PACF behavior of the squared retuned series, it is shown that there is a linear dependence. Table 2 shows the values for the model significance tests. From the significance tests for ARCH and P-value in Table 2, the rejection of the null hypothesis and accepting the alternative hypothesis at a level of 0.05 significance because the value is less than 0.05.

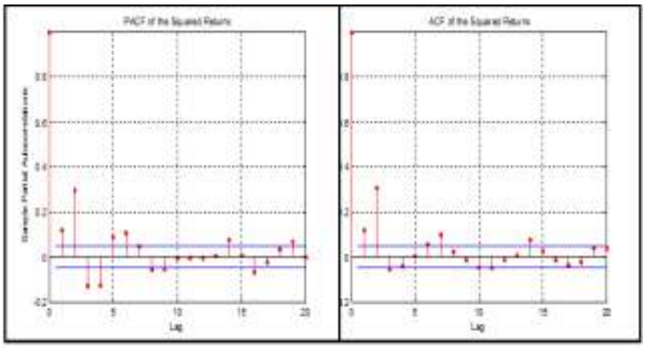

**Fig. 5:** ACF- PACF of the squared retuned series

This means that the series suffer from sequential autocorrelation, at (20, 15, 10) shifts. It is also concluded that the best GARCH model to represent this series is ARCH (1, 1) and the results for the model estimation are shown in Table 3 where G1 = GARCH (1) = 0.13747 and A1 = ARCH (1) = 0.39164. In addition, C = -8.3285e-005 and κ = K = 1.4577e-005.

**Table 2:** Model's Significance Tests

| H | P-Value | Stat | Critical Value | Lag |
|---|---|---|---|---|
| 1 | 0 | 275.4213 | 18.307 | 10 |
| 1 | 0 | 283.7579 | 24.9958 | 15 |
| 1 | 0 | 300.1897 | 31.4104 | 20 |

**Table 3**: Model Estimation Results

| Standard | | T | |
|---|---|---|---|
| **Parameter** | Value | Error | Statistic |
| **C** | -8.3285e-005 | 0.00011478 | -0.7256 |
| **K** | 1.4577e-005 | 5.563e-007 | 26.2037 |
| **GARCH (1)** | 0.13747 | 0.027762 | 4.9519 |
| **ARCH (1)** | 0.39164 | 0.037386 | 10.4755 |

From Table 3 results, it is concluded that the parameters values are significant and the model is suitable for the time series. The mathematical model is as follow:

$$y_t=-8.328*10^{-5}+£_t$$
$$\sigma^2_t=1.4577e\text{-}005+0.13747\sigma^2_{t-1}+0.39164£_{t-1}$$

In order to compare and evaluate the model, the errors, Conditional standard deviation and the returns are compared. Fig. 6 shows the comparison results, while Table 4 shows the parameters indicators for the model error. The curve for those results is shown in Fig. 7.

**Table 4:** Error Measures results for Different Methods

| | RMSE | MAPE | $R^2$ | S.S |
|---|---|---|---|---|
| GARCH | 0.33 | 0.015 | 0.90 | 0.90 |
| BP | 0.29 | 0.0008 | 1 | 1 |
| GRNN | 7.2 | 0.25 | 0.52 | 0.54 |

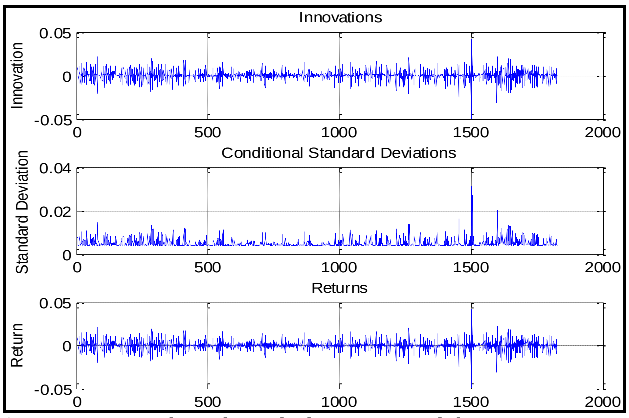


**Fig. 6:** Errors, Conditional Standard Deviation and the Returns comparison

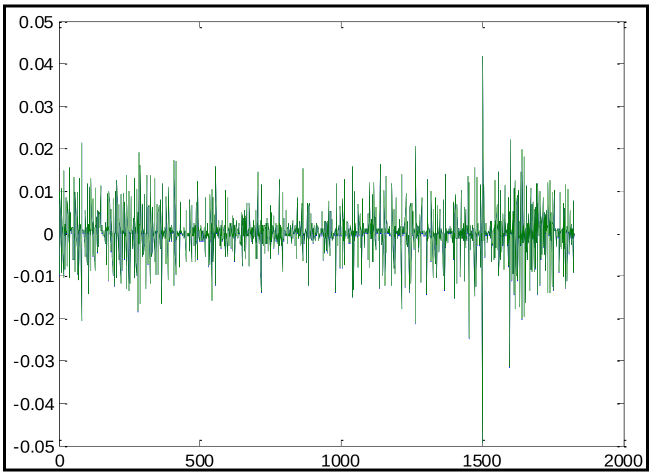

**Fig. 7:** CARCH-Model Compatibility Curve

## 3.2. Artificial Neural Network

The selected network consists of three layers, which are input layer, hidden layer and output layer. The layers are interconnected with weightage; the priority comparison is decided based on MSE. Therefore, the input for the network is decided through to degree shift of the network (Lag1, Lag2), in this way the network input is represented with two variables ($X_{t-1}$, $X_{t-2}$), while the output is represented with $X_t$, as shown in Fig. 8. Where $X_t$ is the original time series.

The network architecture is as follow:

- Input layer: The number of nodes in this layer is two nodes, with two degrees' time series shift, represented with the variables ($X_{t-1}$, $X_{t-2}$).
- Network Training: In the training phase, a logistic function and a sigmoid function, and the maximum number of iterations is 1500, unless the one of the training condition is met. The best performance for the network and best training is decided based on finding the value of Regression (R).
- Hidden Layer: For this layer the optimal number of nodes is 25 nodes, with one layer.
- Output Layer: this layer consists of one node only, represented by the variable t. the results for the BP network for the time series under study is shown in Fig. 8.

In the figure the value of R is 0.99, the number of iterations are 1500. The gradient degree is 1.3718e-06 of the 1500 iterations; Mu is 0.0001 of the iterations. A validation check is 0. While the best training performance is NaN. It is found that the network has concluded all the conditional iterations, thus showing its compatibility and optimality of this neural network for this time series. The errors statistical measures result for the BP network are shown in Table 4, while the compatibility curve is shown in Fig. 9.

## 3.3. GRNN

The architecture of the ANN is used to set the inputs and outputs for this network, the results that are shown in Fig. 4 are calculated using Mathlab, the "newgrnn" function in Mathlab is used to develop the network, while the results are reached using "Sim" function in the same program.

## 3.4. Results Analyses

The results of comparing the two methods in question, that is the neural network represented by BP network and the GARCH model, are summarized in Table 4. From studying the results in the table it is clear that:

- The GRNN is not suitable for this time series and that as expected the BP network is suitable because it has the training to improve the performance.

- It is also important to highlight that the BP network is the best in non-linear models because it has overcome the local minima problem due to its use of back-feeding.
- The results also show that the GARCH model has high prediction skill with 0.90 which means that it is efficient, but the BP network has the best results by achieving perfect prediction skill.

Thus, BP network in such time series based on the error measure and prediction skill indicators.

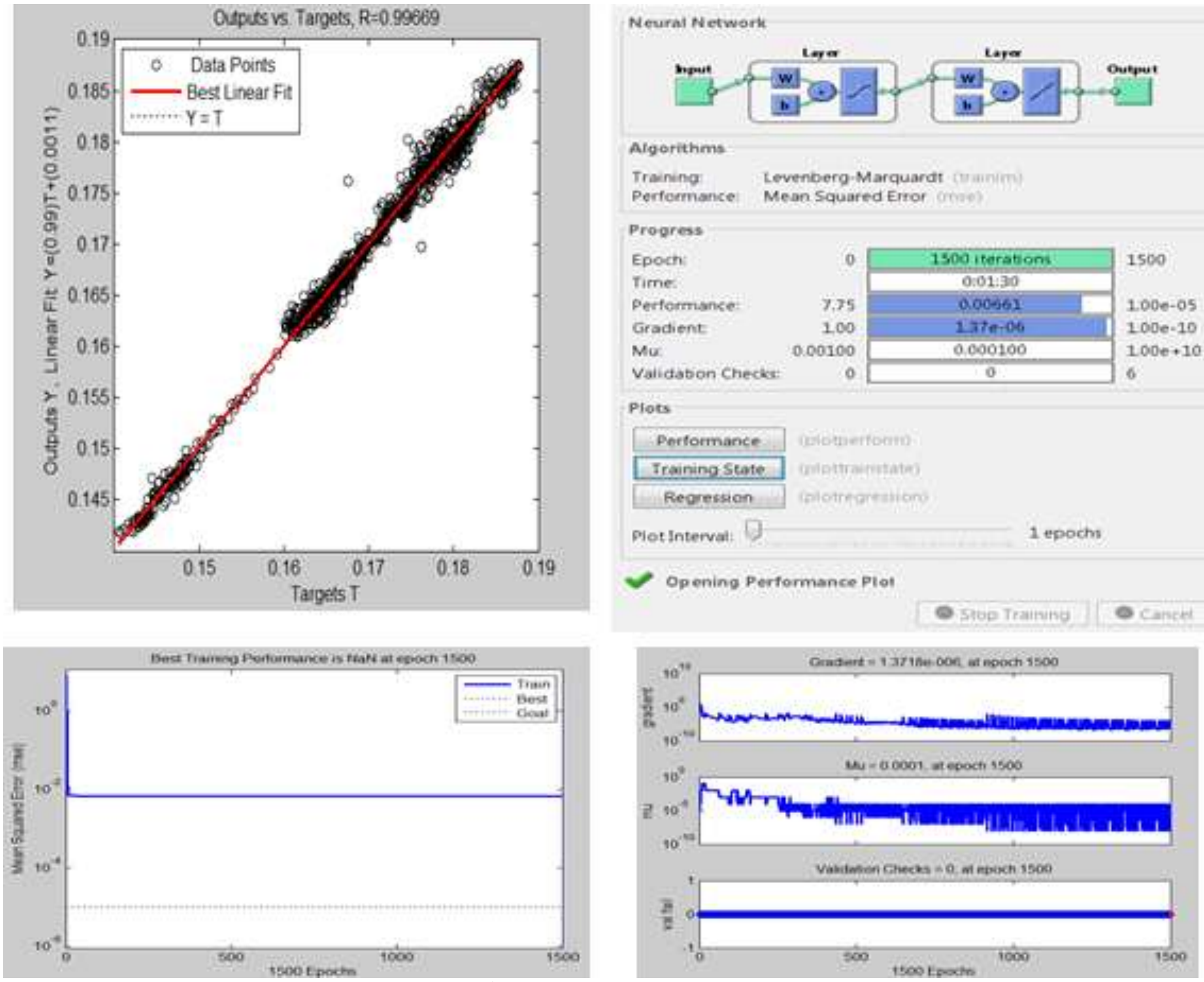


**Fig. 8:** ANN Training Measures

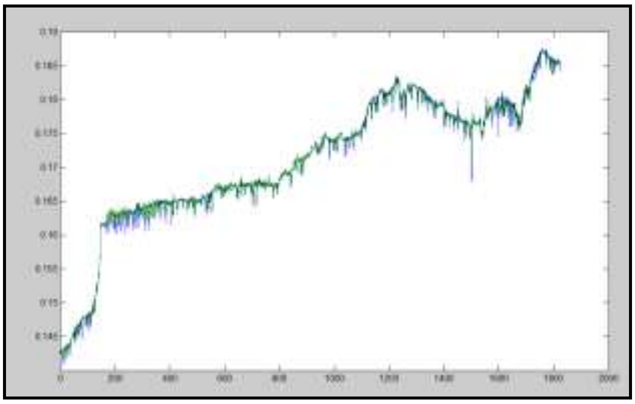

**Fig. 9:** ANN-Time Series Compatibility Curve

## 4. Conclusion

Throughout the course of this research a number of conclusions are reached, they are summarized as follows:

1. The results show that GRNN model is not efficient for handling heterogeneous contrast time series and non-linear time series. Because it does not include the back-feeding in its training methodology.
2. The adoption of BP networks, because of its high efficiency in processing heterogeneous contrast time series, non-linear time series and financial time series. Due to using back-feeding in its training methodology BP networks have the ability to overcome problems like "Local Minimal".
3. The results also show the efficiency of GARCH model in handling back-feeding in its training methodology.
4. The statistical tests' result shows that the best method to represent Egyptian currency exchange rate time series is the BP network.
5. Due to its ability in learning and self-adjusting to any model, the BP network are chosen to replace ARCH, GARCH and other models, in processing heterogeneous, non-linear and semi-linear time series.

Based on the results and conclusions of this paper, it is highly advised for the Egyptian financial market management to adopt the BP network in studying, analyzing and predicting the exchange rate of the currency, in order to be able to guide the sound decision making during crisis. This paper comes as a part of a continues work on the use of ANN to improve forecasting accuracy, the results are consistent with results from other research papers on applying ANN for forecasting other types of time series [10].

## References

[1] G. Bekaert, and R. Hodrick, International financial management. Cambridge University Press, 2017.

[2] M. J. Muhammed, "The use of GARCH model for Saudi financial market forecasting," Proceedings of the Al-Rafidain University Second Scientific Conference, 2010.

[3] R. S. Tsay, Analysis of financial time series. John Wiley and Sons, 2005.

[4] R. Engle, "Risk and volatility: Econometric models and financial practice," American Economic Review, 94(3), 405-420, 2004.

[5] S. Benkachcha. J. Benhra, and H. El Hassani, "Seasonal time series forecasting model based on artificial neural network," International Journal of Computer Applications, 116(20), 9-14, 2015.

[6] A. K. Dhamijam, and V. K. Bhalla, "Financial time series forecasting: Comparison of neural networks and ARCH models," International Research Journal of Finance and Economics, 49, 185-202, 2010.

[7] O. Erdogan and A. Goksu, "Forecasting Euro and Turkish Lira exchange rates with Artificial Neural Networks (ANN)," International Journal of Academic Research in Accounting, Finance and Management Sciences, 4(4), 307-316, 2014.

[8] S. Malik and A. K. Bhatt, "Developing a model for financial forecasting through artificial neural networks,"

[9] O. Stetter, D. Battaglia, J. Soriano, and T. Geisel, "Model-free reconstruction of excitatory neuronal connectivity from calcium imaging signals," Plos Computational Biology, 8, 1-26, 2012.

[10] M. A. Ashour, and R. A. Abbas, “Improving time series' forecast errors by using recurrent neural networks,” Proceedings of the ACM 7th International Conference on Software and Computer Applications, pp. 229-232, 2018.