An Intelligent Model for Stock Market Prediction

Ibrahim M. Hamed[1]
Ibrahim.hamed@gmail.com

Ashraf S. Hussein[1]
ashrafh@acm.org

Mohamed F. Tolba[1]
fahmytolba@gmail.com

[1] *Department of Scientific Computing*
Faculty of Computer and Information Sciences, Ain Shams University
Abassia, Cairo, 11566, Egypt

Received 6 June 2011
Accepted 13 June 2012

Abstract

This paper presents an intelligent model for stock market signal prediction using Multi-Layer Perceptron (MLP) Artificial Neural Networks (ANN). Blind source separation technique, from signal processing, is integrated with the learning phase of the constructed baseline MLP ANN to overcome the problems of prediction accuracy and lack of generalization. Kullback Leibler Divergence (KLD) is used, as a learning algorithm, because it converges fast and provides generalization in the learning mechanism. Both accuracy and efficiency of the proposed model were confirmed through the Microsoft stock, from wall-street market, and various data sets, from different sectors of the Egyptian stock market. In addition, sensitivity analysis was conducted on the various parameters of the model to ensure the coverage of the generalization issue. Finally, statistical significance was examined using ANOVA test.

Keywords: stock market prediction; technical indicator; artificial neural networks; blind source separation;

1. Introduction

Financial market prediction has been one of the most challenging goals of the Artificial Intelligence (AI) research community. This research is meant to go far beyond the capabilities of traditional AI research, which primarily focuses on developing systems that are supposed to emulate human intelligence[1], because stock market is generally non-linear and volatile. The fluctuation rate may depend on many factors; equity, interest rate, securities, options, warrants merger and ownership of large financial corporations or companies. Still, no one can consistently predict the stock market movement. That is why; this kind of AI prediction requires an iterative process of knowledge discovery and system improvement through knowledge engineering, data mining, theoretical and data-driven modeling, as well as trial and error [2].

The stock market has always been one of the most popular investments due to its high returns[3]. In this market, predictions are based on either fundamental analysis[a] (See Refs. 4 and 5) or technical analysis[b] (See Refs. 6 and 7). Fundamental analysis is related to analyzing the assets and the economical values of a security[c]. Recently, technological analysis[d] has been used in the prediction, as it aims at more accurate results, higher performance and broad calculations. It was motivated by the fact that one cannot perform any of technical or fundamental analysis manually on more than 2-3 securities at a time or even per trading session[8].

It is an interesting topic to predict the trend of a security price in the stock market. The prediction process is complicated due to its nonlinearity and uncertainty. In developing a stock market prediction system, one of the most important tasks is to select the

[a] Fundamental analysis of a business involves analyzing its financial statements and health, its management and competitive advantages, and its competitors and markets.
[b] Technical analysis is a security analysis discipline for forecasting the direction of prices through the study of past market data, primarily price and volume
[c] Security is a financial asset such as a share or bond. In stock market, it is referred to as a stock symbol.
[d] Technological analysis is based on using new technology and computation power to substitute old hand operated process. It can be an alternate for technical analysis, fundamental analysis or a hybrid of both.

Published by Atlantis Press

input variables. For example, only one-day return of the closing price of a stock was used in Ref. 9 while the difference between the price and the moving average, highest and lowest prices were used in Ref. 10. In addition to price series, volume of transactions, macroeconomic data and market indicators that were considered as input variables in Ref. 11.

However, there is no one technique or combination of techniques, which has been successful enough to consistently "beat the market"[12]. What works for a security will not fit another. With the development of the ANN, researchers and investors are hoping that the market mysteries can be revealed.

In this paper, a proposed model is presented to overcome the existing problems of accuracy and generalization in the stock market prediction. The proposed model uses MLP ANN, which was selected based on the literature survey of the previous models that use neural networks with one hidden layer[13]. This ANN is augmented through using the blind source separation technique in the learning phase. KLD[14] is used as the learning algorithm for the selected ANN.
The proposed model is based on technical analysis. It uses a set of technical indicators for the prediction, Simple Moving Averages (SMA), Exponential Moving Average (EMA) and Average Directional Index (ADX) for trend detection. A simple rule based system is added to classify the signals for “Buy”, “Sell” or “Hold”.

The rest of the paper is organized as follows: in section 2, an overview of the previous work is given. In section 3, a detailed description of the proposed prediction model is explained. In section 4, the implementation details are elaborated. In section 5, one provides one’s experimental results and analysis. Finally, conclusions are given in section 6.

2. Stock market prediction models

The area of stock market prediction has been pursued by many research groups. These work groups focused on the prediction problem from the perspective of prediction accuracy. Basically, two main approaches were used by early researchers to tackle this problem, which are based on either ANNs or Fuzzy Logic (See Refs. 15-20). Many applications were created based on these two approaches. These applications covered a variety of techniques. Examples from early work done using the first method include the work of White[9], who applied ANN based models to detect nonlinear patterns in the price movement of IBM assets, Yoon et al.[21], who successfully adopted ANN methods, for stock prediction, in comparison to the multiple discriminate analysis methods, proving their accuracy and effectiveness and Kimoto et al.[22], who used modular ANN to the Tokyo Stock Exchange Prices Indexes (TOPIX). All of these methods and models achieved accurate predictions, but yet this was performed on the market index not certain stock with actual “Buy” and “Sell” signals.

Reaz et al.[23] implemented the back propagation ANN on the Altera FLEXlOK FPGA, utilizing the parallelism in the ANN architecture to achieve better performance than regular PCs at that time. Liao et al.[24] utilized the stochastic time effective series ANN model on the Chinese stock index (HIS) and some of the US Stock Market indices. This model was found to be effective when verified against the data of HSI, Dow Jones (DJI), NASDAQ Composite (IXIC) and S&P500.

Wang et al.[25] used Time Delay Neural Networks (TDNN) with the intention of investigating the influences of trading volume on the short term predictions. It is interesting to point out that both theoretical and empirical studies have proven the nonlinear relation between stock return and trading volume. Their study emphasized on whether trading volume can improve the forecasting performance of ANN or whether ANN can take advantage of such nonlinearity to get more accurate results. The results showed that the trading volume cannot significantly improve the forecasting performance of S&P 500 when applied to the data of S&P 500 and DJI Market indices since 1990 till 2002[25].

There have been fewer studies on the second method. First, Sugeno et al.[26] proposed a general qualitative model based on fuzzy logic and applied it to stock market. Next, Hiemstra[27] introduced architecture for fuzzy logic forecasting support system for Stock Market Prediction. Application of this architecture facilitated the process of knowledge base update and simulation. In addition, Wang et al.[28] proposed a fuzzy personalized stock information agent based on fuzzy logic, intelligent agent and personalization. Stocks set selection was based on user input, but the agent was able to define the top 10 ranked stocks among the selected set. Finally, Wang[20] constructed a data mart to minimize the size of the stock market data and incorporated fuzzification techniques with the Grey theory and applied the Fuzzy Grey prediction on the Taiwan Stock market, from September 2000 to April 2001.

Combination between the aforementioned methodologies has been considered in the context of Stock Market Predictions (See Refs. 15 and 19). Fuzzy systems and neural networks were combined by developing neural network architecture capable to stimulate the behavior of fuzzy systems. Each layer in the neural network acts as a stage in the fuzzy system, for example, the first layer will be responsible for mapping the input to its membership function while the last layer performs the defuzzification stage. Not much work was invested in this direction, but some relevant examples are mentioned below.

Quah[15] proposed an Adaptive Neuro Fuzzy Inference System (ANFIS) for stock market prediction. ANFIS system, which is an instance of the more generic form of Takagi-Sugeno-Kang (TSK) model[30], replaces the fuzzy sets in the implication with a first order polynomial equation of the input variables. Esfahanipour et al.[29] applied a model that contains TSK fuzzy rule based system[30], which is based on selected technical indicators as input variables from the Tehran stock market. Their contribution focused on the variable selection method using ANFIS, and it was found to outperform other approaches such as Back Propagation Neural Network (BPNN) or Multiple Regression Analysis (MRA). Yet, this was a basic step in the prediction process that missed adding various factors such as fundamental analysis and macroeconomic change to be able to predict the price trend movement.

Novel paradigms such as Genetic Programming (GP) and Markov Model have been investigated in the stock market prediction. For example, El-telbany[31] proposed a GP based approach for stock market prediction. The population consists of individuals represented by specific data structure - trees. Inner nodes of the trees can represent functions (e.g. arithmetic operators, conditional operators or problem specific functions) and leaf nodes would be terminals - external inputs, constants, and zero argument functions. On the other hand, Zhang et al.[32] applied Markov Chain Model. It was found that, using empirical results, this model is able to express the probability of a certain state of stock prices in the future. Yet, this model was not verified against live data to figure out the real performance of the system and its behavior for generalization.

Comparative studies on various techniques discussed in this section have been presented in Refs. 15, 16 and 31. The purpose of these studies was to define the best technique to be used, since a vast number of techniques and methods were proposed in this context. For example, Egeli et al.[16] compared the MLP-ANN and the Generalized Feed-Forward (GFF) ANN, for stock market prediction. This study was performed with the data of the Central Bank of the Republic of Turkey, for the period between 2001 and 2003. In addition, simple sensitivity analysis was provided to select the best parameters for the ANN architecture. The MLP provided acceptable results and was found to outperform the GFF. El-telbany[31] successfully compared his proposed GP results with a three layered GFF ANN. The input data was the Egyptian stock market index CASE30, from 2001 to 2003.

Finally, Quah[15] compared three techniques: optimized MLP ANN, ANFIS and General Growing and Pruning Radial Basis Function (GGAP-RBF). He applied these techniques on data from the DJIA symbol, Singaporean stock market, in the period from 2003 to 2004. This comparison revealed that the optimized MLP ANN was better than both ANFSI and GGAP-RBF.

The aforesaid accuracy optimization techniques were quite promising when applied to real life historical data. The major problems encountered in these techniques were related to accuracy and generalization of the model.

The goal of this research work is not only to improve the prediction accuracy, but also to build a general model that can realize securities from different sectors and stock markets. The model should adapt to nonlinearity in the stock market and un-correlated data of different securities in different stock markets. The proposed technique adopts the ANN architecture (MLP), based on the literature survey. The KLD learning algorithm is used to enhance the performance of the proposed ANN. KLD, being a blind source separation technique, helps in solving the generalization issue of the prediction problem. The proposed model was found to provide more accurate results, converges faster and generalizes to different stocks.

3. Proposed Prediction Model

The proposed model comprises of several stages as shown in Fig. 1. The first stage is concerned with the input selection. Next, the appropriate preprocessing is performed on the selected input data. Such preprocessing may be indicators calculation, fundamental assets evaluation or even data classification for the supervised learning of the ANN. The data is then passed to the ANN to be trained for the classification purposes. The main objective of the learning algorithm is to update the weights between the neural network neurons in order to minimize the error of the prediction results. The proposed technique uses a blind source separation technique; KLD[33-34], as a learning algorithm for the MLP ANN.

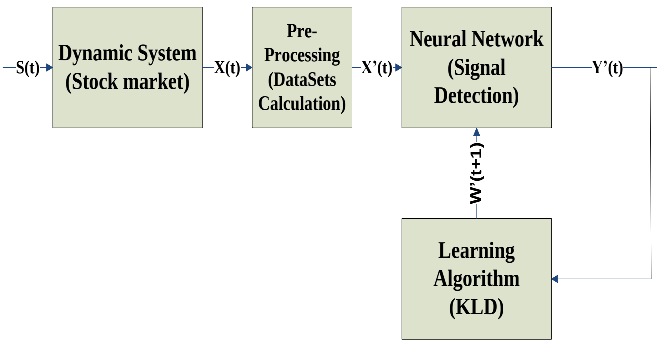


Fig. 1. Block diagram illustrating the proposed model

3.1. *Data*

The selected data is the stock market's daily data. Daily data is the data used in the stock market daily transactions, and it may be called market summary. The daily data is composed of (a) open price, which is the first trading price for a security on a given trading session, (b) close price, which is the final trading price for a security on a given trading session, (c) high price, which is the highest trading price for a security during a given trading session, (d) low price, which is the lowest trading price for a security during a given trading session, (e) volume, which is the amount of trades transacted for a security on a given trading session, as shown in Table 1.

Table 1 The used data

Field	Description
Open	the opening price for a trading session
Close	the closing price for a trading session
High	the highest trading price achieved during a session
Low	the lowest trading price reached during a session
Volume	the total amount of trades preformed during a session

3.2. *Preprocessing*

The input to this stage is the stock market daily data for a chosen security as described in the section above. The preprocessing step is concerned with computing the indicators.

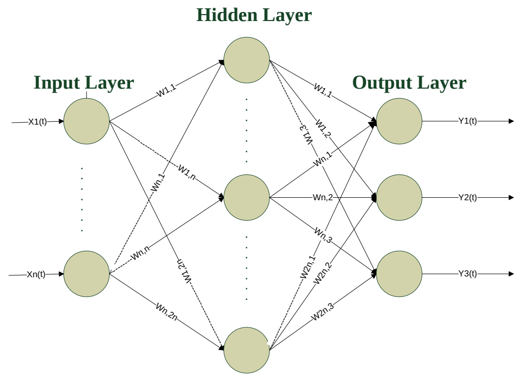


Fig. 2. The implemented Neural Network Architecture

3.3. *Neural Network*

The used MLP ANN architecture has one hidden layer. Sigmoid function with range [-1, +1] was used as the activation function for each neuron, as shown in Fig. 2. The number of input neurons is equal to the number of variables in the data set, while the number of hidden neurons equals to twice the input neurons. This was found to perform better after several trials of different hidden neurons combination. The signal is being classified into three classes "Buy", "Sell" or "Hold". So, three output neurons are being used in the output layer. Each neuron should have the value [0 – 1] indicating the class it belongs to. For a given run, the output (0.8 0.12 0.08) means it is a "Buy" signal, since it is closer to the buy class, while (0.05 0.85 0.1) is a

"Sell" signal. For any output to be valid, it should belong only to one class.

3.4. *Learning Algorithm*

Through the iterative supervised learning of the ANN, the data is processed through an intermediate stage to normalize the output of the current stage before entering the next iteration. Due to the nature of the KLD, the input to this function must be in the form of a probability distribution function, i.e. the magnitude of the output vector must be one. So, the output vector is normalized to match this criterion. Then, it is passed to the KLD to compute the divergence between the output signal and the desired signal. The weights are updated according to (1).

$$W' = W + \gamma * \mathrm{KLD}(Y_{\mathrm{desired}}, Y_{\mathrm{actual}}) \quad (1)$$

where γ is the learning rate, which we set to 0.1.

KLD is also known as information divergence[35], is an asymmetric difference measure between two probability distribution functions P and Q. It measures the expected number of extra bits needed to encode samples from P while using a code based on Q. It is illustrated in (2).

$$\mathrm{KLD}(P, Q) = \sum_{i} P(i) * \log \frac{P(i)}{Q(i)} \quad (2)$$

The KLD satisfies some properties known as divergence properties:

- Self Similarity: $\mathrm{KLD}(P, P) = 0$
- Self Identification: $\mathrm{KLD}(P, Q) = 0$ only if $P = Q$
- Positivity: $\mathrm{KLD}(P, Q) \geq 0$ for all $P \& Q$

In this experiment, KLD was implemented using MatLab for (2). The inputs to this function, P and Q, are mapped directly to the desired and expected results respectively, from (1) Y_{desired} and Y_{actual}. Then, this MatLab function is called during the learning process to update the weights as illustrated in (1).

KLD is used in many domains and applications. In image processing, Goldberger et al.[36] used it to compute the best matching images using their histogram model. In the field of computation theory, Quint[37] used it to find the similarity between two weighted automata. In information retrieval (IR), it was used by Lafferty et al.[38] to measure the distance between queries and members of document collections. Also, Ismail et al.[39] used KLD as an application for population prediction. They also proposed using KLD in the domain of stock market prediction as a future work.

4. Implementation

In this paper, MLP with KLD[39] was implemented, as the prediction unit for the technical indicators, as described in sections 3. A set of technical indicators were used in this implementation as described below.

4.1. *Moving Average*

The Simple Moving Average (SMA) is the average price of a given security over a certain period (Window size) as shown in Fig. **3**. It is common to be calculated based on the closing price of the security as shown in (3). A 14-day simple moving average is the 14 day sum of closing prices divided by 14. As its name implies, a moving average is an average that moves. Old data is skipped as new data is available. This moves the average along the time scale.

$$\mathrm{SMA}_{t,n} = \frac{\sum_{k=t-n}^{n} \mathrm{Close}_k}{n} \quad (3)$$

Although moving averages are lagging indicators, since they are computed from the previous data, Exponential Moving Average (EMA) reduces that lag by applying more weight to recent prices. The weighting applied to the most recent price depends on the window size of the moving average. This makes the EMA follow the short term price movement while SMA follows long term price movement, which is obvious in Fig. 3. Exponential moving average is calculated as shown in (4).

$$EMA_{t,n} = \left\{ \left[\mathrm{Close}_k - \mathrm{EMA}_{t-1,n-1} \right] \mathrm{X}\ \frac{2}{n+1} \right\} + \mathrm{EMA}_{t-1,n-1} \quad (4)$$

Where, $EMA_{t-n,n} = SMA_{t-n,n}$

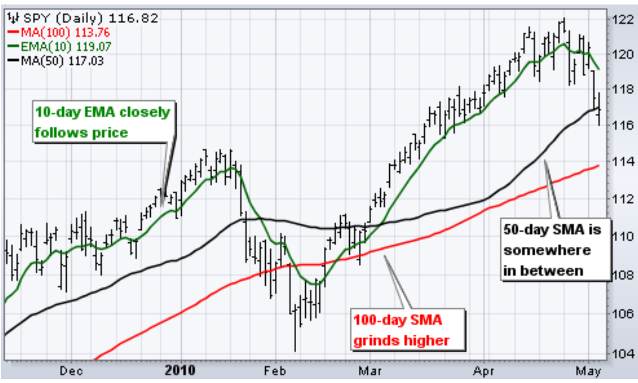


Fig. 3. Moving Average

4.2. *Average Directional Index*

The ADX was developed by Wilder[40] to indicate the strength of a trend, as the trend of the security is crucial, since it affects the "Buy" and "Sell" decisions based on whether it is trending up or down or moving sideways. The ADX is an oscillator that ranges between 0 and 100. Readings above 60 are relatively rare. Readings below 20 emphasize a weak trend and readings above 40 emphasize a strong trend. The indicator does not grade the trend as bullish or bearish, but shows the strength of the current trend. A reading below 20 indicates a weak downtrend as well as a weak uptrend Wilder[40].

Fig. 4. Average Directional Index chart

The ADX is calculated from the Positive Directional Indicator (+DI) and the Negative Directional Indicator (-DI) as shown in (5. ADX is the thick black line with less fluctuation, +DI is green and -DI is red as shown in Fig. 4. +DI indicates the strength of the up moves and -DI indicates the strength of the down moves over a given period. The ADX combines +DI with -DI, and then smoothes the data with a moving average to provide a measurement of trend strength. ADX does not show any indication of trend direction, since it uses +DI and -DI. Readings above 40 indicate a strong trend and readings below 20 a weak trend.

$$
\begin{aligned}
ADX_{\mathrm{t}} &= DX_{\mathrm{t}} \times \frac{2}{\mathrm{n}+1} + \left(1 - \frac{2}{\mathrm{n}+1}\right) \times ADX_{\mathrm{t}-1} \\
DX_{\mathrm{t}} &= \frac{|DI^{+} - DI^{-}|}{DI^{+} + DI^{-}} \times 100 \\
DI^{+} &= \frac{ADM_t^{+}}{ATR_t} \times 100 \\
DI^{-} &= \frac{ADM_t^{-}}{ATR_t} \times 100 \\
ATR_t &= TR_t \times \frac{2}{\mathrm{n}+1} + \left(1 - \frac{2}{\mathrm{n}+1}\right) \times ATR_{\mathrm{t}-1} \\
\mathrm{TR_t} &= \max\,(High_t - Low_t\,|Hight_t \\
&\qquad - Close_{t-1}|Close_{t-1} - Low_t) \\
ADM_t^{+} &= DM_t^{+} \times \frac{2}{\mathrm{n}+1} + \left(1 - \frac{2}{\mathrm{n}+1}\right) \\
&\qquad \times ADM_{t-1}^{+} \\
ADM_t^{-} &= DM_t^{-} \times \frac{2}{\mathrm{n}+1} + \left(1 - \frac{2}{\mathrm{n}+1}\right) \\
&\qquad \times ADM_{t-1}^{-} \\
\Delta High_t &= High_t - High_{t-1} \\
\Delta Low_t &= Low_{t-1} - Low_t \\
DM_t^{+} &= \begin{cases} \Delta High_t & \Delta High_t > \Delta Low_t \\ 0 & otherwise \end{cases} \\
DM_t^{-} &= \begin{cases} \Delta Low_t & \Delta High_t < \Delta Low_t \\ 0 & otherwise \end{cases}
\end{aligned}
\tag{5}
$$

This input is used to calculate the indicators mentioned above (SMA, EMA & ADX) with a given window and filter size. The selection of the appropriate window size and filter size will be illustrated in the next section. After the indicators are being calculated, they are processed through a rule based system. This rule based system classifies the input signals into "Sell", "Hold" or "Buy". These classifications are used later on in the learning stage. A sample of the rules used in this system is shown below:

```
If(ADX is trending)
{
        If(Close Cross MA to Downside)
                Filter(Sell)
        If(Close Cross MA to Downside)
                Filter(Buy)
        If(Short MA Cross Long MA to Downside)
                Filter(Sell)
        If(Short MA Cross Long MA to Downside)
                Filter(Buy)
        If(Close < support & valid Buy Signal)
                Fire(Stoploss)
        If(Close > Resistance & valid Sell Signal)
                Fire(Reentry)
        If(Short MA< support & valid Buy Signal)
                Fire(Stoploss)
        If(Short MA > Resistance & valid Sell Signal)
                Fire(Reentry)
         If(Short MA tests Long MA)
                Filter(Last Valid Signal)
        If(ADX Direction Changed)
                Filter(Last Valid Signal)
}
Other wise Fire(Hold)
```

5. Results and Discussion

Early researches, focusing on ANN and expert systems, faced a lot of challenges in the growing markets, such as the Egyptian market. Lack of historical data and mistakes found in the stock system represent the major weak points when training the neural network. For example, the opening price was never saved since 1999. Now, the availability of such data and the updates in the stock system opened the space for soft computing to produce useful information for the traders. In this research work, the proposed model was tested using the daily data for EFG Hermes Holding (HRHO), El Nasr Clothing - Textiles Co. (KABO), Egyptian Electrical Cables (ELEC) and Microsoft (MS) from March 1999 to August 2008 with a total of 1900 records, except MS was 2600 records[8]. A random 15% of this data was used for testing.

The selected data was meant to cover a large margin of the stock market in terms of sectors, currency, trading volume and session type. First, HRHO was selected as a very active security with a large trading volume, in the investments sector and traded with the local currency. KABO was selected from the industrial sector, which has large trading volumes and is traded in USD. ELEC was selected from the Off-Trading Session (OTS), which is a 30 min session by the end of the trading day for corporates, which have some financial or legal violations and have a small trading volume. MS was chosen as a test case from an international mature stock market.

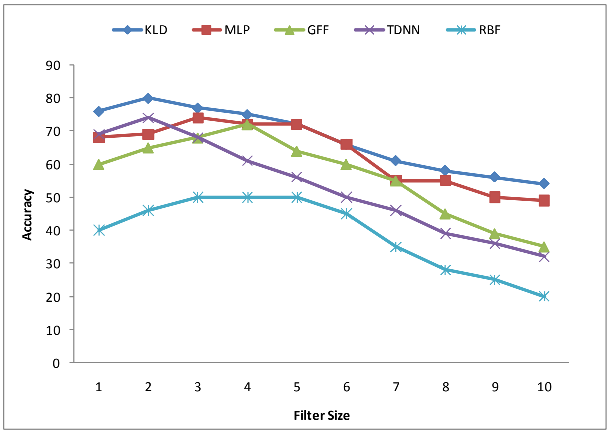
Most of the previous work concerning stock market predictions emphasizes the movement in the market index itself as in Refs. 15-16, 22, 24, 29, 31 and 41. This gives more insight into the market status, but such insight is not enough for an investor to make successful trading decisions. Even though the market may be trending upwards, investors can still lose money due to mistaken analysis of the security's current situation. As such, choosing security data from within the market, rather than the market index, increases the chances that the resultant "Buy", "Sell" or "Hold" signals are more lucrative to the investor.

The appropriate window size, for the selected indicators in the data set, was selected based on the conducted sensitivity analysis. Beside the window size for the indicators, there are two other filters. The first is for the ADX signal. The purpose of this filter is to ensure the strength of the trend, i.e. to remove noise and false sudden moves in the price. The second one is for the "Buy", "Sell" or "Hold" signals. The purpose of this filter is to validate the signal and remove noise due to spikes in the price movement. Since rumors can tamper with a security price movement, especially in growing markets like Egypt, this generates noise that lasts over a relatively short period: one to two trading sessions. After that, the signal is corrected again to match the actual value of the assets represented by the security.

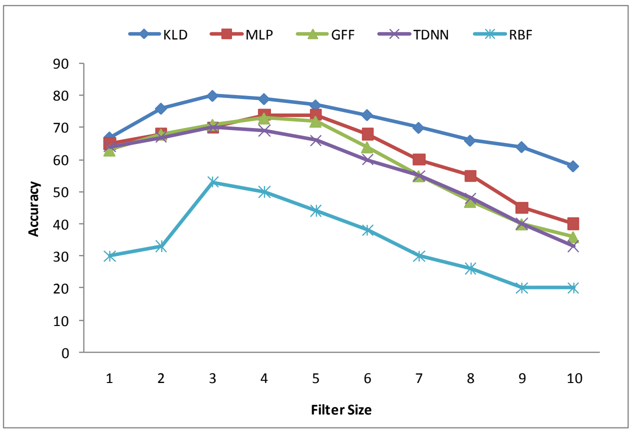
Sensitivity analysis was conducted for four variables of the selected implementation data set: SMA window size, EMA window size, ADX filter and signal filter. For the window size, one tried the size range [2-72] while one tried the range [1-10] for the filters. For each variable, one tried each value from its given range with all possible combination of the other 3 variables. Variable ranges could not be larger, as this makes short term and medium term signals, trends and movements disappear. So, the neural network will capture only long term trades and will fail to detect short term and medium term trades. Therefore, the prediction model accuracy will decrease. Results from the proposed model were compared to that of the efficient ANN architecture from the previous work of early researchers mentioned in Refs. 15, 16, 18 and 41.

According to this experiment, the best value for the ADX filter was 2 bars as shown from Fig. 5.a to Fig. 8.a. A bar represents the basic unit of the data aggregation. Since the data set used was aggregated on a daily level, a bar in this case means one trading day. The best value for the signal filter was 3 bars as shown from Fig. 5.b to Fig. 8.b. After 3 bars the signals detected are still of high accuracy. But, due to large filter size some true, short term, signals were ignored because they were passed by such a long filter over the signal validity. The best values for SMA and EMA are shown from Fig. 5 to Fig. 8 (c & d) respectively. As shown, the prediction accuracy increases with increasing the parameters value until a certain threshold, at which some classifications are missed.

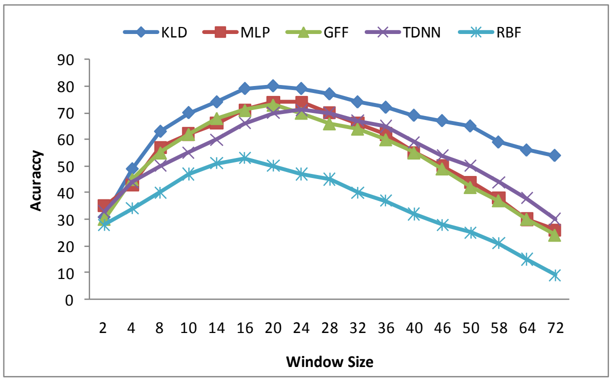
This sensitivity analysis was performed for the 4 securities mentioned before. The results for these securities were nearly the same for the proposed solution but the other techniques behavior was inconsistent. For example, the MLP was being outperformed by the TDNN for the data of ELEC while they were nearly equivalent for the data of KABO. Besides, the RBF results were much closer to the MLP for the data of KABO and it dropped way far for the data of ELEC.



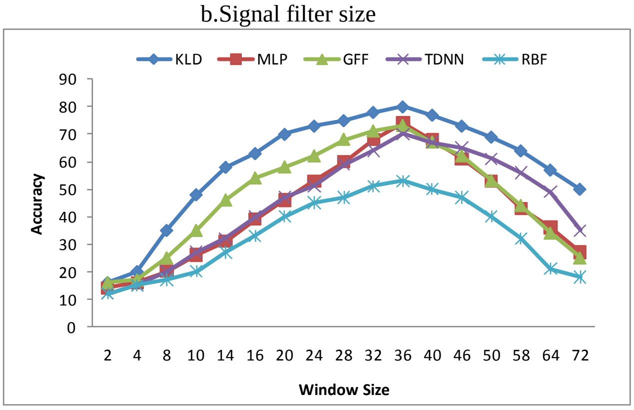
a.ADX filter



b.Signal filter size

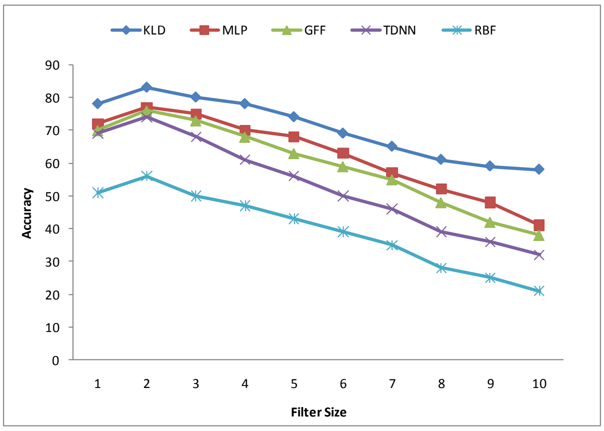


c.SMA window size

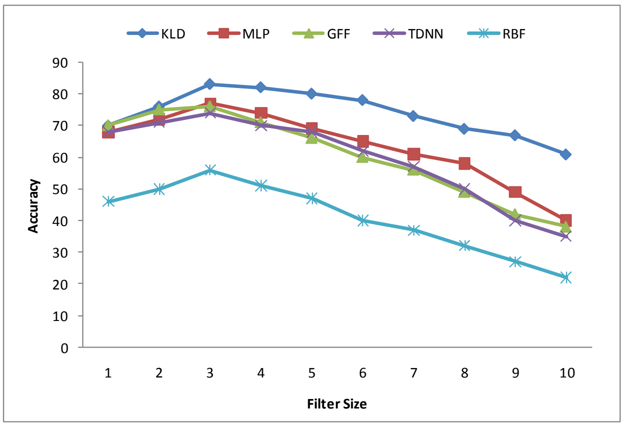


d.EMA window size

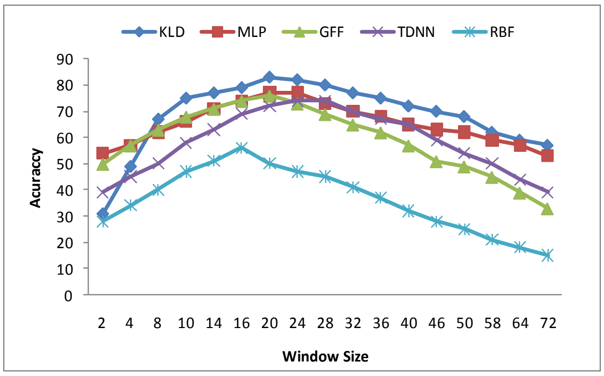
Fig. 5. Sensitivity analysis for data set parameters using HRHO data



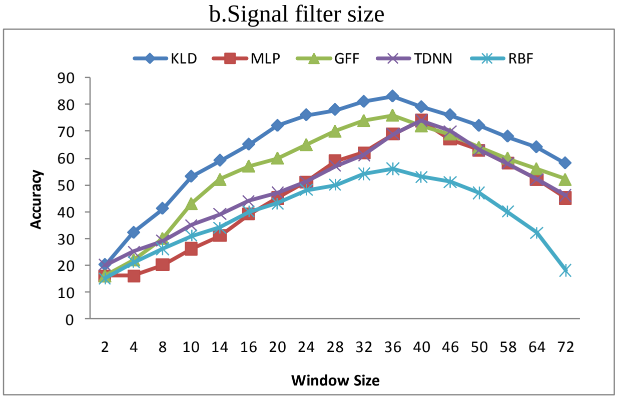
a.ADX filter



b.Signal filter size

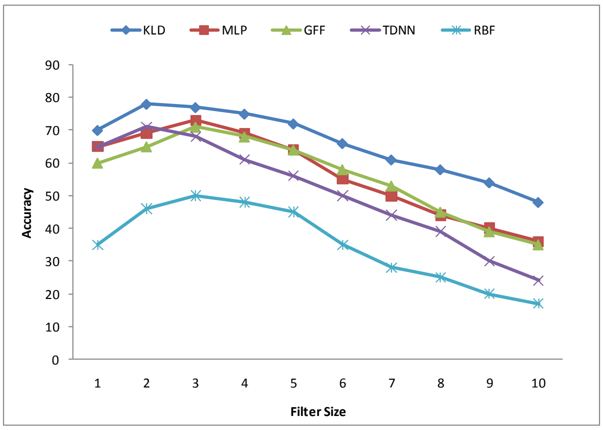


c.SMA window size

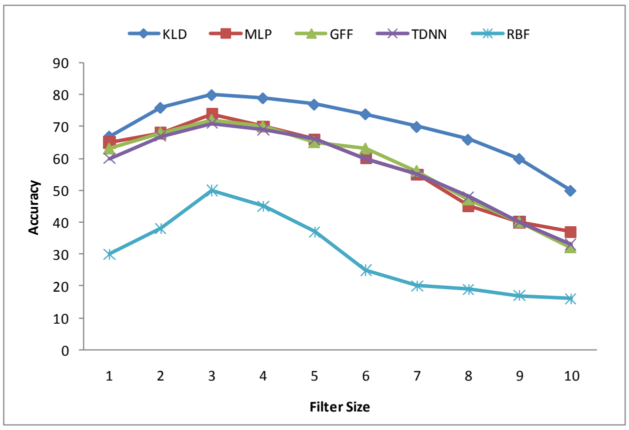


d.EMA window size

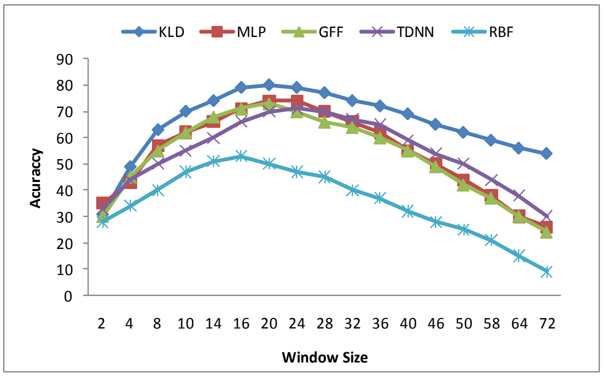
Fig. 6. Sensitivity analysis for data set parameters using KABO data



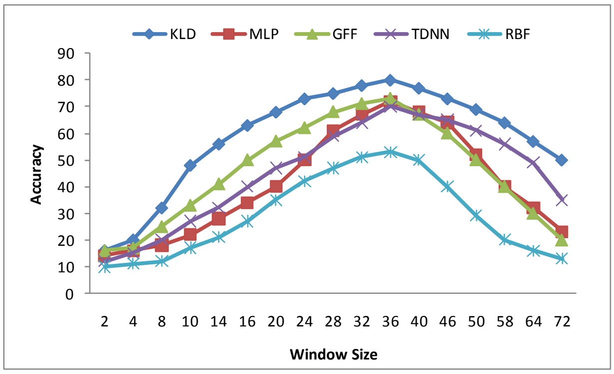
a.ADX filter



b.Signal filter size

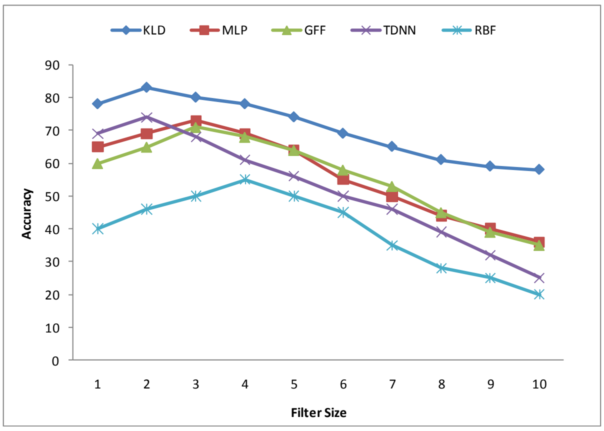


c.SMA window size

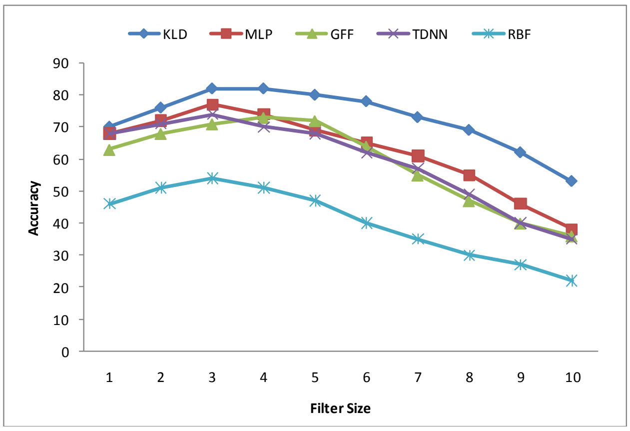


d.EMA window size

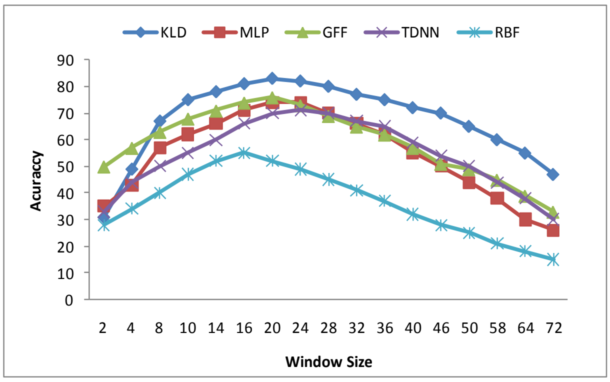
Fig. 7. Sensitivity analysis for data set parameters using ELEC data



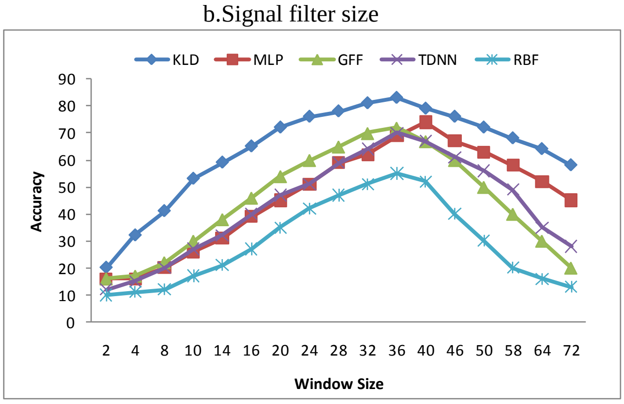
a.ADX filter size



b.Signal filter size



c.SMA window size



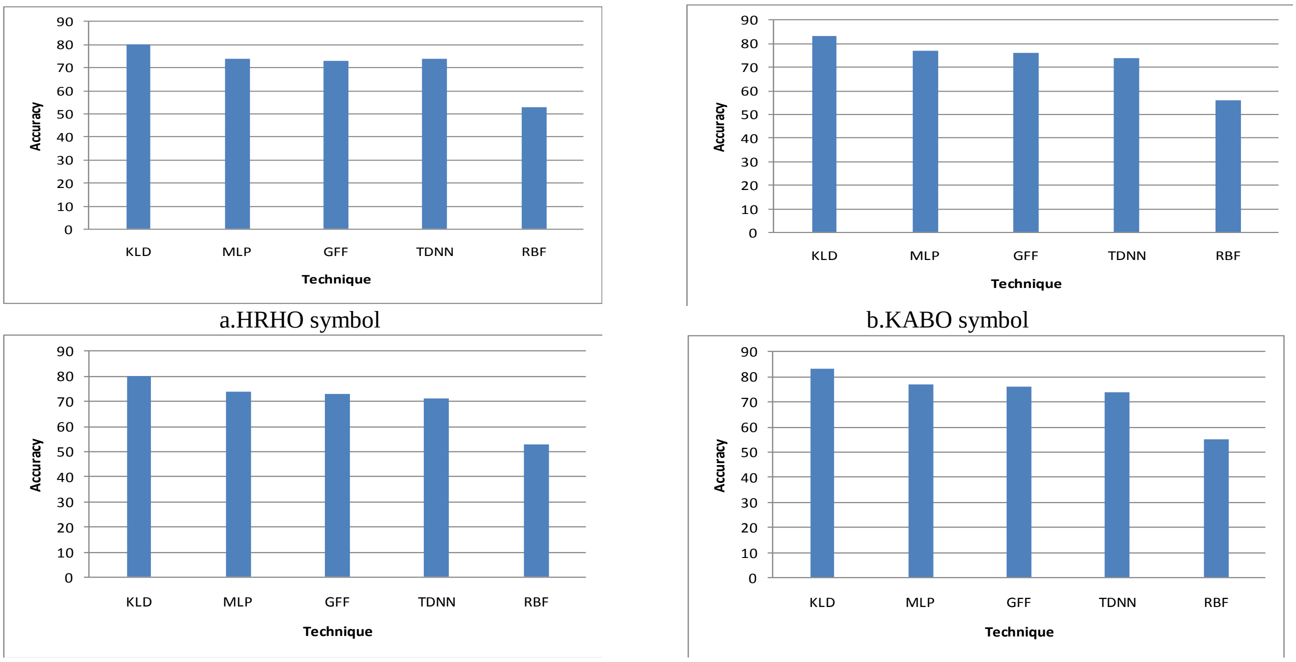
d.EMA window size

Fig. 8. Sensitivity results for data set parameters using MS data

In addition, the RBF-ANN proposed by Quah[15] gave the worst accuracy, with the selected data set. On the other hand, the proposed model maintains the same behavior for different data. As shown in the figures above, the KLD graph pattern is approximately the same for all the selected securities (HRHO, KABO, ELEC and MS), with slight changes in some values. This shows a great progress in the generalization problem for the existing prediction models.

Then, maximum accuracy for most of the given techniques is reached at the previous parameters. This was reached because such parameters realize as much as possible for short term, moderate term and long term moves, accompanied by their signal corrections. On the graphs, when one moves right, the parameter values and the accuracy increase. Then, the accuracy curve begins to drop again as in this stage the selected parameters are more than enough to be affected by short term moves and some of the moderate term moves. So, most of the captured signals are long term signals. Therefore, the model fails to capture a major section of the market signals.

After using the best parameter values from the previous analysis, the results were compared to the previous work of Quah[15], Egeli et al.[16], Grosan et al.[41] and Jang[18] to ensure the accuracy of the proposed technique. The results were computed as the average of 20 separate cycles of training and testing. The maximum accuracy of the proposed technique is 83% as shown in Fig. 9. This adds 5% to 7% to the accuracy of the same neural network without the proposed technique.



a.HRHO symbol

b.KABO symbol

c.ELEC symbol

d.MS symbol

Fig. 9. Accuracy analysis of the proposed technique

ANOVA test[42] was performed on the results to ensure the statistical significance, i.e. the classifications accuracy was not a result of random act. As mentioned earlier, the data sets were run for 20 complete cycles and average accuracy results were used as shown in Table 2. ANOVA test resulted that the F was 9.6 while the critical F was 2.7 as shown in Table 3. Therefore, the means are significantly different and the generalization effect is real.

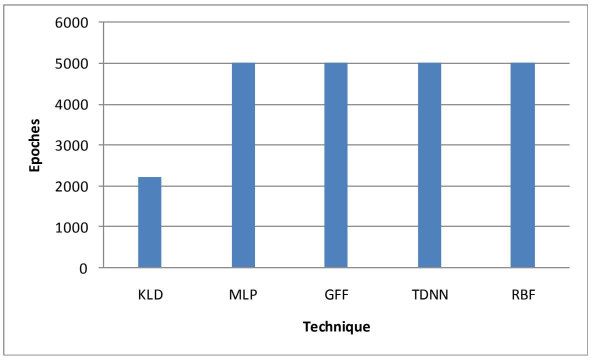
Table 2 Data Sets runs summary

Groups	*Count*	*Sum*	*Average*	*Variance*
MS	20	1660	83	3.684211
HRHO	20	1604	80.2	6.063158
KABO	20	1660	83	4.105263
ELEC	20	1606	80.3	6.747368

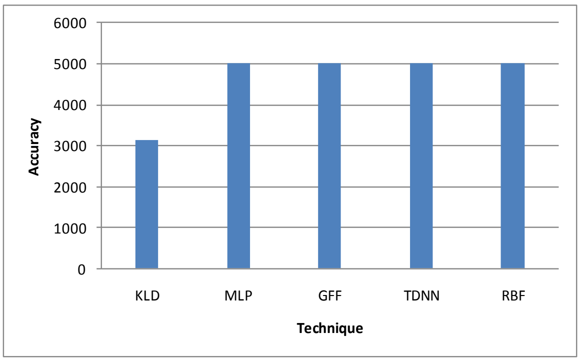
Table 3 ANOVA Test Results

Source of Variation	*SS*	*Df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	151.4	3	50.45	9.8	1.5E-5	2.74
Within Groups	391.4	76	5.15			
Total	542.8	79				

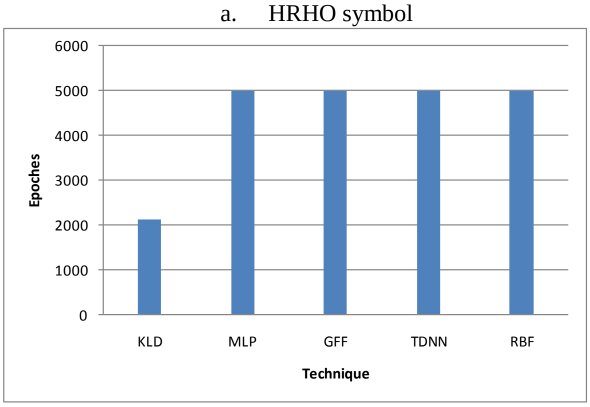
In addition, the performance of the proposed technique is better. The proposed technique converges faster than the other techniques as shown in Fig. 10. The previous techniques always reach the maximum epochs and do not stop at the minimum error criteria. The maximum limit of epochs is set so that the neural network does not fall in the over fitting problem. On the other hand, the proposed technique converges faster, on the average of 3000 epochs, when using the Mean Squared Error as the error measurement function.



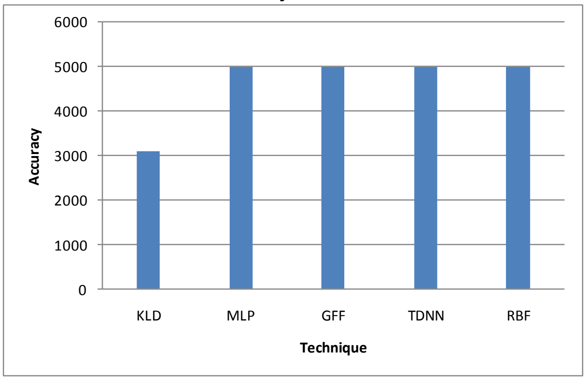
a. HRHO symbol



b. MS symbol



c. KABO symbol



d. ELEC symbol

Fig. 10. Performance analysis of the proposed technique

6. Conclusion

This paper presented a new stock market prediction model based on blind source separation for stock market signal prediction, using Neural Networks. The proposed technique utilizes blind source separation to optimize and speed up the learning process of the Neural Network. KLD was integrated with the neural network as the learning algorithm. This technique was tested with three securities from local Egyptian stock market covering wide sectors; MS security was also included as a test case from a mature global stock market.

The accuracy of the proposed model was confirmed in comparison with the results of the other most accurate previous techniques with average difference of about 5-7%. The results were confirmed to be statistically significant using ANOVA test. Also, the proposed model outperforms the other techniques in the training with an average of 3000 epochs to reach the minimum error, while the other techniques stop at the maximum epochs limit to avoid over fitting. The proposed model covers the generalization problem as mentioned in the sensitivity analysis.

For the future work, there are three issues that can be considered; improving accuracy, optimizing the performance of the real time prediction and testing the scalability. For the first problem, the proposed model can be tested with a number of blind source separation techniques as a learning algorithm. For the second problem, GPU architecture will serve to reach a better accuracy and higher speed up. Finally, the proposed framework obviously will perform well on scaling out. but a further scalability test needs to be performed to give a detailed results and analysis of the framework scalability.

References

1. H. Kwa'snicka and M. Ciosmak, Intelligent techniques in stock analysis, in *Proceedings of Intelligent Information Systems*, 2001.
2. M.R. Hassan and B. Nath, Stock market forecasting using hidden markov model: A new approach, in *ISDA'05. Proceedings. 5th International Conference on Intelligent Systems Design and Applications*, 2005.
3. N. Jegadeesh and S. Titman, Returns to buying winners and selling losers: Implications for stock market efficiency, *The Journal of Finance*, vol. 48, no. 1, pp. 65-91, 1993.
4. P.M. Dechow, A.P. Hutton, L. Meulbroek and R.G. Sloan, Short-sellers, fundamental analysis, and stock returns, *Journal of Financial Economics*, vol. 61, no. 1, pp. 77-106, 2001.
5. J.S. Abarbanell and B.J. Bushee, Fundamental analysis, future earnings and stock prices, *Journal of Accounting Research*, vol. 35, no. 1, pp. 1-24, 1997.
6. D.P. Brown and R.H. Jennings, On technical analysis, *Review of Financial Studies*, vol. 2, no. 4, p. 527, 1989.
7. L. Blume, D. Easley and M. O'hara, Market statistics and technical analysis: The role of volume, *The Journal of Finance*, vol. 49, no. 1, pp. 153-181, 1994.
8. Okaz. [Online.. Available: https://www.okazinvest.com/.
9. H. White, Economic prediction using neural networks: the case of IBM daily stock returns, in *IEEE International Conference on Neural Networks*, 1988.
10. E. Tsang, J. Li and J.M. Butler, EDDIE beats the bookies, *Software - Practice and Experience*, vol. 28, no. 10, pp. 1033-1043, 1998.
11. D. S. Barr and G. Mani, Using neural nets to manage investments., *AI Expert*, vol. 9, no. 2, pp. 16-21, 1994.
12. R. Lawrence, Using neural networks to forecast stock market prices, *Course Project, University of Manitoba*, 1997.
13. T. S. Quah, Using neural network for DJIA stock selection, *Engineering Letters*, vol. 15, no. 1, pp. 126-133, September 2007.
14. S. Kullback and R. A. Leibler, On information and sufficiency, *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79-86, 1951.
15. T. S. Quah, Using neural network for DJIA stock selection, *Engineering Letters*, vol. 15, no. 1, pp. 126-133, September 2007.
16. B. Egeli, M. Ozturan and B. Badur, Stock market prediction using artificial neural networks, in *International Conference on Business*, Hawaii, 2003.
17. P. Tiˇno, C. Schittenkopf and G. Dorffner, Financial volatility trading using recurrent neural networks., *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865-874, 2001.
18. J. S. Jang, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665-685, 1993.
19. O. Castillo and P. Melin, Simulation and forecasting complex financial time series using neural networks and fuzzy logic, in *IEEE International Conference on Systems Management and Cybernetics*, 2001.
20. Y. F. Wang, Predicting stock price using fuzzy gray prediction system, *Expert systems with applications*, vol. 22, no. 1, pp. 33-38, 2002.
21. Y. Yoon and G. Swales, Predicting stock price performance: A neural network approach, Kauai, HI , USA, 1991.
22. T. Kimoto, K. Asakawa, M. Yoda and M. Takeoka, Stock market prediction system with modular neural networks, in *International Joint Conference on Neural Networks*, San Diego, CA, 1990.
23. M. Reaz, S. Z. Islam, M. A. Ali and M. S. Sulaiman, FPGA realization of backpropagation for stock market prediction, in *Proceedings of the 9th International Conference on Neural Information Processing*, Singapore, 2002.
24. Z. Liao and J. Wang, Forecasting model of global stock index by stochastic time effective neural network, *Expert Systems with Applications*, vol. 37, no. 1, pp. 834-841, 2010.
25. X. Wang, P. K. Phua and W. Lin, Stock market prediction using neural networks: Does trading volume help in short-term prediction?, in *Proceedings of the International Joint Conference on Neural Networks*, 2003.
26. M. Sugeno and T. Yasukawa, A fuzzy logic based approach to qualitative modeling, *IEEE Transactions on fuzzy systems*, vol. 1, no. 1, pp. 7-31, 1993.
27. Y. Hiemstra, A stock market forecasting support system based on fuzzy logic, in *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, Wailea, HI, 1994.
28. Y. Wang, Y. Zhang, S. Belkasim and R. Sunderraman, Real time fuzzy personalized web stock information agent, in *Proceedings of the 2nd International Workshop on Intelligent Systems Design and Applications*, 2002.
29. A. Esfahanipour and W. Aghamiri, Adapted neuro-fuzzy inference system on indirect approach TSK fuzzy rule base for stock market analysis, *Expert Systems with Applications*, vol. 37, no. 7, pp. 4741-4748, 2010.
30. P.C. Chang and C.H. Liu, A TSK type fuzzy rule based system for stock price prediction, *Expert Systems with applications*, vol. 34, no. 1, pp. 135-144, 2008.
31. M. E. El-telbany, The Egyptian stock market return prediction: A genetic programming approach, in *International Conference on Electrical, Electronic and Computer Engineering*, Kunming, 2005.
32. D. Zhang and X. Zhang, Study on forecasting the stock market trend based on stochastic analysis method, vol. 4, no. 6, 2009.
33. S.I. Amari and A. Cichocki, Adaptive blind signal processing-neural approaches, *Proceedings of the IEEE,*

vol. 86, no. 10, pp. 2026-2048, 1998.

34. S. Amari, T.P. Chen and A. Cichocki, Stability analysis of adaptive blind source separation, *Neural Networks*, vol. 10, no. 8, pp. 1345-1351, November 1997.
35. S. Kullback and R. A. Leibler, On information and sufficiency, *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79-86, 1951.
36. J. Goldberger, S. Gordon, and H. Greenspan, An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures, vol. 1, 2003.
37. J. Quint, On the equivalence of weighted finite-state transducers, Barcelona, Spain, 2004.
38. J. D. Lafferty and C. Zhai., Document language models, query models, and risk minimization for information retrieval, in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001.
39. I. A. Ismail and T. Nabil, A hybrid approach for time series forecasting using artificial neural network and kullback-leibler divergence, vol. 15, no. 2, 2004.
40. J. W. Wilder, New concepts in technical trading systems, First, Ed., Trend research, 1978.
41. C. Grosan, A. Abraham, V. Ramos and S.Y. Han, Stock market prediction using multi expression programming, in *portuguese conference on Artificial intelligence*, 2005.
42. R.A. Johnson and D.W. Wichern, Applied multivariate statistical analysis, 5 ed., Prentice Hall Upper Saddle River, NJ, 2002.