International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:1, 2010

Fast Forecasting of Stock Market Prices by using New High Speed Time Delay Neural Networks

Hazem M. El-Bakry, Nikos Mastorakis

***Abstract*—**Fast forecasting of stock market prices is very important for strategic planning. In this paper, a new approach for fast forecasting of stock market prices is presented. Such algorithm uses new high speed time delay neural networks (HSTDNNs). The operation of these networks relies on performing cross correlation in the frequency domain between the input data and the input weights of neural networks. It is proved mathematically and practically that the number of computation steps required for the presented HSTDNNs is less than that needed by traditional time delay neural networks (TTDNNs). Simulation results using MATLAB confirm the theoretical computations.

***Keywords*—**Fast Forecasting, Stock Market Prices, Time Delay Neural Networks, Cross Correlation, Frequency Domain.

I. Introduction

FORECASTING of stock market prices is very important for strategic planning in business companies. Because of strong competition between companies, such forecasting process is required to be done as fast as possible. With the advent of the digital computer, forecasting of stock market prices has since moved into the technological realm [36-68]. The most prominent technique involves the use of artificial neural networks (ANNs). Recently, time delay neural networks have shown very good results in different areas such as automatic control, speech recognition, blind equalization of time-varying channel and other communication applications. The main objective of this research is to reduce the response time of time delay neural networks. The purpose is to perform the testing process in the frequency domain instead of the time domain. Our approach was successfully applied for sub-image detection using high speed neural networks (HSNNs) as proposed in [5]. Furthermore, it was used for fast face detection [7,9,26], and fast iris detection [8]. Another idea to further increase the speed of HSNNs through image decomposition was suggested in [7].

HSNNs for detecting a certain code in one dimensional serial stream of sequential data were described in [10,27]. Compared with traditional neural networks, HSNNs based on cross correlation between the tested data and the input weights of neural networks in the frequency domain showed a significant reduction in the number of computation steps required for certain data detection [5-35]. Here, we make use of the theory of HSNNs implemented in the frequency domain to increase the speed of time delay neural networks for market price prediction. The idea of moving the testing process from the time domain to the frequency domain is applied to time delay neural networks. Theoretical and practical results show that the proposed HSTDNNs are faster than TTDNNs.

The paper is organized as follows. Theory of neural networks for pattern detection is described in section II. Section III presents HSTDNNs for forecasting of stock market prices. Experimental results for fast forecasting of stock market prices by using HSTDNNs are given in section IV.

II. Theory of ANNs for Forecasting

Artificial neural network (ANN) is a mathematical model, which can be one or more layered that contain many artificial neural cells. The wide usage of the ANN may be due to the three basic properties:

1. The ability of the ANN as a parallel processing of the problems, for which if any of the neurons violate the constraints would not affect the overall output of the problem.

2. The ability of the ANN to extrapolate from historical data to generate forecasts.

3. The successful application of the ANN to solve non-linear problems. The history and theory of the ANN have been described in a large number of published literatures and will not be covered in this paper except for a very brief overview of how neural networks operate.

The ANN computation can be divided into two phases: learning phase and testing phase. The learning phase forms an iterative updating of the synoptic weights based upon the error back propagation algorithm. Back propagation algorithm is generalized of least mean square learning rule, which is an approximation of steepest descent technique. To find the best approximation, multi-layer feed forward neural network architecture with back propagation learning rule is used. A schematic diagram of typical multi-layer feed-forward traditional time delay neural network architecture is shown in Fig. 1. The number of neurons in the hidden layer is varied to give the network enough power to solve the forecasting problem. Each neuron computes a weighted sum of the individual inputs (*I_1*, *I_2*, …, *Ij*) it receives and adding it with a bias (*b*) to form the net input (*x*). The bias is included in the neurons to allow the activation function to be offset from zero.

$$\mathrm{sum} = w_{1,1}.I_1 + w_{1,2}.I_2 + ... + w_{1,j}.I_j + b \qquad (1)$$

H. M. El-Bakry is assistant professor with Dept. of Information Systems - Faculty of Computer Science and Information Systems – Mansoura University – Egypt. (phone: +2-050-2349340, fax: +2-050-2221442, e-mail: helbakry20@yahoo.com).

N. Mastorakis is full professor with Technical University of Sofia, Bulgaria.

Where, *wji* is the connection weight between neuron *j* and neuron *i*. The net input (*sum*) is then passed to the subsequent layer through a non linear sigmoid function to form its own output (y*j*).

$$y_j = 1/\left(1+e^{-sum}\right) \quad (2)$$

Afterward, the output *yj* was compared with the target output *tj* using an error function of the form:

$$\delta_k = \left(t_j - y_j\right) y_j \left(1 - y_j\right) \quad (3)$$

For the neuron in the hidden layer, the error term is given by the following equation:

$$\delta_j = y_j \left(1 - y_j\right) \sum_k \delta_k w_k \quad (4)$$

where *δk* is the error term of the output layer, and *wk* is the weight between the hidden layer and output layer. The error was then propagated backward from the output layer to the input layer to update the weight of each connection at iteration (*t* + 1) as follows:

$$\begin{aligned} w_{ji}(t+1) &= w_{ji}(t) + \eta \delta_j y_j \\ &+ \alpha \left(w_{ii}(t) - w_{ii}(t-1)\right) \end{aligned} \quad (5)$$

Choosing a small learning rate η leads to slow rate of convergence, and too large η leads to oscillation. The term α is called momentum factor and determines the effect of past weight changes on the current direction of movement. Both of these constant terms are specified at the start of the training cycle and determine the speed and stability of the network. The process was repeated for each input pattern until the error was reduced to a threshold value.

III. Forecasting of Stock Market Prices by using HSTDNNs

Computing the resulted output; for a certain pattern of information; in the incoming serial data, is a prediction problem. First neural networks are trained to predict the estimated variable and this is done in time domain. In pattern detection phase, each position in the incoming matrix is processed to predict the estimated variable by using neural networks. At each position in the input one dimensional matrix, each sub-matrix is multiplied by a window of weights, which has the same size as the sub-matrix. The outputs of neurons in the hidden layer are multiplied by the weights of the output layer. Thus, we may conclude that the whole problem is a cross correlation between the incoming serial data and the weights of neurons in the hidden layer.

The convolution theorem in mathematical analysis says that a convolution of f with h is identical to the result of the following steps: let F and H be the results of the Fourier Transformation of f and h in the frequency domain. Multiply F and H* in the frequency domain point by point and then transform this product into the spatial domain via the inverse Fourier Transform. As a result, these cross correlations can be represented by a product in the frequency domain. Thus, by using cross correlation in the frequency domain, speed up in an order of magnitude can be achieved during the detection process [5-35]. Assume that the size of the attack code is 1xn. In attack detection phase, a sub matrix I of size 1xn (sliding window) is extracted from the tested matrix, which has a size of 1xN. Such sub matrix, which may be an attack code, is fed to the neural network. Let W_i be the matrix of weights between the input sub-matrix and the hidden layer. This vector has a size of 1xn and can be represented as 1xn matrix. The output of hidden neurons h(i) can be calculated as follows [5]:

$$h_i = g\left(\sum_{k=1}^{n} W_i(k) I(k) + b_i\right) \quad (6)$$

where g is the activation function and b(i) is the bias of each hidden neuron (i). Equation 1 represents the output of each hidden neuron for a particular sub-matrix I. It can be obtained to the whole input matrix Z as follows [5]:

$$h_i(u) = g\left(\sum_{k=-n/2}^{n/2} W_i(k)\, Z(u+k) + b_i\right) \quad (7)$$

Eq.6 represents a cross correlation operation. Given any two functions f and d, their cross correlation can be obtained by [4]:

$$d(x) \otimes f(x) = \left(\sum_{n=-\infty}^{\infty} f(x+n) d(n)\right) \quad (8)$$

Therefore, Eq. 7 may be written as follows [5]:

$$h_i = g\left(W_i \otimes Z + b_i\right) \quad (9)$$

where h_i is the output of the hidden neuron (i) and h_i (u) is the activity of the hidden unit (i) when the sliding window is located at position (u) and (u) ∈ [N-n+1].

Now, the above cross correlation can be expressed in terms of one dimensional Fast Fourier Transform as follows [5]:

$$W_i \otimes Z = F^{-1}\left(F(Z) \bullet F^{*}\left(W_i\right)\right) \quad (10)$$

Hence, by evaluating this cross correlation, a speed up ratio can be obtained comparable to traditional neural networks. Also, the final output of the neural network can be evaluated as follows:

$$O(u) = g\left(\sum_{i=1}^{q} W_o(i)\, h_i(u) + b_o\right) \quad (11)$$

where q is the number of neurons in the hidden layer. O(u) is the output of the neural network when the sliding window located at the position (u) in the input matrix Z. W_o is the weight matrix between hidden and output layer.

IV. Complexity Analysis of HSTDNNs for Forecasting Stock Market Prices

The complexity of cross correlation in the frequency domain can be analyzed as follows:

1- For a tested matrix of 1xN elements, the 1D-FFT requires a number equal to $N\log_2 N$ of complex computation steps [13]. Also, the same number of complex computation steps is required for computing the 1D-FFT of the weight matrix at each neuron in the hidden layer.

2- At each neuron in the hidden layer, the inverse 1D-FFT is computed. Therefore, q backward and (1+q) forward transforms have to be computed. Therefore, for a given matrix under test, the total number of operations required to compute the 1D-FFT is $(2q+1)N\log_2 N$.

3- The number of computation steps required by HSTDNNs is complex and must be converted into a real version. It is known that, the one dimensional Fast Fourier Transform requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions [3]. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. Therefore, the total number of computation steps required to obtain the 1D-FFT of a 1xN matrix is:

$$\rho=6((N/2)\log_2 N) + 2(N\log_2 N) \qquad (12)$$

which may be simplified to:

$$\rho=5N\log_2 N \qquad (13)$$

4- Both the input and the weight matrices should be dot multiplied in the frequency domain. Thus, a number of complex computation steps equal to qN should be considered. This means 6qN real operations will be added to the number of computation steps required by HSTDNNs.

5- In order to perform cross correlation in the frequency domain, the weight matrix must be extended to have the same size as the input matrix. So, a number of zeros = (N-n) must be added to the weight matrix. This requires a total real number of computation steps = q(N-n) for all neurons. Moreover, after computing the FFT for the weight matrix, the conjugate of this matrix must be obtained. As a result, a real number of computation steps = qN should be added in order to obtain the conjugate of the weight matrix for all neurons. Also, a number of real computation steps equal to N is required to create butterflies complex numbers ($e^{-jk(2\Pi n/N)}$), where $0<K<L$. These (N/2) complex numbers are multiplied by the elements of the input matrix or by previous complex numbers during the computation of FFT. To create a complex number requires two real floating point operations. Thus, the total number of computation steps required for HSTDNNs becomes:

$$\sigma=(2q+1)(5N\log_2 N)+6qN+q(N-n)+qN+N \qquad (14)$$

which can be reformulated as:

$$\sigma=(2q+1)(5N\log_2 N)+q(8N-n)+N \qquad (15)$$

6- Using sliding window of size 1xn for the same matrix of 1xN pixels, q(2n-1)(N-n+1) computation steps are required when using TTDNNs for certain attack detection or processing (n) input data. The theoretical speed up factor η can be evaluated as follows:

$$\eta=\frac{q(2n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \qquad (16)$$

HSTDNNs are shown in Fig. 2.

Time delay neural networks accept serial input data with fixed size (n). Therefore, the number of input neurons equals to (n). Instead of treating (n) inputs, the proposed new approach is to collect all the incoming data together in a long vector (for example 100xn). Then the input data is tested by time delay neural networks as a single pattern with length L (L=100xn). Such a test is performed in the frequency domain as described before.

The theoretical speed up ratio for searching short successive (n) code in a long input vector (L) using time delay neural networks is listed in tables I, II, and III. Also, the practical speed up ratio for manipulating matrices of different sizes (L) and different sized weight matrices (n) using a 2.7 GHz processor and MATLAB is shown in table IV.

An interesting point is that the memory capacity is reduced when using HSTDNNs. This is because the number of variables is reduced compared with TTDNNs.

V. Conclusion

A New technique for fast forecasting of stock market price has been presented. Such strategy has been realized by using our design for HSTDNNs. Theoretical computations have shown that HSTDNNs require fewer computation steps than traditional ones. This has been achieved by applying cross correlation in the frequency domain between the input data and the weights of neural networks. Simulation results have confirmed this proof by using MATLAB.

References

[1] Justin Wolfers, Eric Zitzewitz, “Interpreting Prediction Market Prices as Probabilities” Institute for the Study of Labor (IZA) in its series IZA Discussion Papers with number 2092, April 2006.

[2] Azoff, E.M. Neural Network Time Series Forecasting of Financial Markets John Wiley and Sons Ltd, 1994.

[3] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series, " Math. Comput. 19, 1965, pp. 297–301.

[4] R. Klette, and Zamperon, "Handbook of image processing operators," John Wiley & Sonsltd, 1996.

[5] Hazem M. El-Bakry, "A Novel High Speed Neural Model for Fast Pattern Recognition," Soft Computing Journal, vol. 14, no. 6, 2010, pp. 647-666.

[6] Hazem M. El-Bakry, "New Faster Normalized Neural Networks for Sub-Matrix Detection using Cross Correlation in the Frequency Domain and Matrix Decomposition, " Applied Soft Computing journal, vol. 8, issue 2, March 2008, pp. 1131-1149.

[7] Hazem M. El-Bakry, "Face detection using fast neural networks and image decomposition," Neurocomputing Journal, vol. 48, 2002, pp. 1039-1046.

[8] Hazem M. El-Bakry, "Human Iris Detection Using Fast Cooperative Modular Neural Nets and Image Decomposition," Machine Graphics & Vision Journal (MG&V), vol. 11, no. 4, 2002, pp. 498-512.

[9] Hazem M. El-Bakry, "New Fast Principal Component Analysis For Real-Time Face Detection," Machine Graphics & Vision Journal (MG&V), vol. 18, no.4, 2009, pp. 405-426.

[10] Hazem M. El-Bakry, "A New Neural Design for Faster Pattern Detection Using Cross Correlation and Matrix Decomposition," Neural World journal, Neural World Journal, 2009, vol. 19, no. 2, pp. 131-164.

[11] Hazem M. El-Bakry, "An Efficient Algorithm for Pattern Detection using Combined Classifiers and Data Fusion," Information Fusion Journal, vol. 11, 2010, pp. 133-148.

[12] Hazem M. El-Bakry, "New Fast Time Delay Neural Networks Using Cross Correlation Performed in the Frequency Domain," Neurocomputing Journal, vol. 69, October 2006, pp. 2360-2363.

[13] Hazem M. El-Bakry, "A New High Speed Neural Model For Character Recognition Using Cross Correlation and Matrix Decomposition," International Journal of Signal Processing, vol.2, no.3, 2005, pp. 183-202.

[14] Hazem M. El-Bakry, "New High Speed Normalized Neural Networks for Fast Pattern Discovery on Web Pages," International Journal of Computer Science and Network Security, vol.6, No. 2A, February 2006, pp.142-152.

[15] Hazem M. El-Bakry "Fast Iris Detection for Personal Verification Using Modular Neural Networks," Lecture Notes in Computer Science, Springer, vol. 2206, October 2001, pp. 269-283.

[16] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Normalized Neural Processors For Pattern Detection Based on Cross Correlation Implemented in the Frequency Domain," Journal of Research and Practice in Information Technology, Vol. 38, No.2, May 2006, pp. 151-170.

[17] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Time Delay Neural Networks," International Journal of Neural Systems, vol. 15, no.6, December 2005, pp.445-455.

[18] Hazem M. El-Bakry, and Qiangfu Zhao, "Speeding-up Normalized Neural Networks For Face/Object Detection," Machine Graphics & Vision Journal (MG&V), vol. 14, No.1, 2005, pp. 29-59.

[19] Hazem M. El-Bakry, and Qiangfu Zhao, "A New Technique for Fast Pattern Recognition Using Normalized Neural Networks," WSEAS Transactions on Information Science and Applications, issue 11, vol. 2, November 2005, pp. 1816-1835.

[20] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Complex Valued Time Delay Neural Networks," International Journal of Computational Intelligence, vol.2, no.1, pp. 16-26, 2005.

[21] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Pattern Detection Using Neural Networks Realized in Frequency Domain," Enformatika Transactions on Engineering, Computing, and Technology, February 25-27, 2005, pp. 89-92.

[22] Hazem M. El-Bakry, and Qiangfu Zhao, "Face Detection Using Fast Neural Processors and Image Decomposition," International Journal of Computational Intelligence, vol.1, no.4, 2004, pp. 313-316.

[23] Hazem M. El-Bakry, and Ahmed Atwan, "Fast Business Networks for Real-Time Applications," International Journal of Universal Computer Sciences, issue 1, vol. 1, 2010, pp. 41-50.

[24] Hazem M. El-bakry, and Nikos Mastorakis, "Prediction of Market Price by using Fast Time Delay Neural Networks," Proc. of 10th WSEAS Int. Conf. on Neural Networks (NN'10), Romania, June 13-15, 2010, pp. 230-237.

[25] Hazem M. El-Bakry, and Nikos Mastorakis, "A New Fast Forecasting Technique using High Speed Neural Networks," WSEAS Transactions on Signal Processing, issue 10, vol. 4, October 2008, pp. 573-595.

[26] Hazem M. El-Bakry, "Automatic Human Face Recognition Using Modular Neural Networks," Machine Graphics & Vision Journal (MG&V), vol. 10, no. 1, 2001, pp. 47-73.

[27] Hazem M. El-Bakry, and H. Stoyan, "FNNs for Code Detection in Sequential Data Using Neural Networks for Communication Applications," Proc. of the First International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2004, 21-25.

[28] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Word Detection in a Speech Using New High Speed Time Delay Neural Networks," WSEAS Transactions on Information Science and Applications, issue 7, vol. 5, July 2009, pp. 261-270.

[29] Hazem M. El-Bakry, "Fast Virus Detection by using High Speed Time Delay Neural Networks," Journal of Computer Virology, vol.6, no.2, 2010, pp.115-122.

[30] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Image Matching on Web Pages," WSEAS Transactions on Signal Processing, issue 4, vol. 5, June 2009, pp. 157-166.

[31] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Information Retrieval from Web Pages," WSEAS Transactions on Information Science and Applications, issue 6, vol. 6, June 2009, pp. 1018-1036.

[32] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Detection of Specific Information in Voice Signal over Internet Protocol," WSEAS Transactions on Communications, issue 5, vol. 8, May 2009, pp. 483-494.

[33] Hazem M. El-Bakry, and Ahmed A. Mohammed, "Optimal Document Archiving and Fast Information Retrieval," The International Journal of Computer science, and Engineering, issue 2, vol. 3, July 2009, pp. 108-121.

[34] Hazem M. El-bakry, and Nikos Mastorakis "Fast Time Delay Neural Networks for Word Detection in a Video Conference," Proc. of European Computing and Computational Intelligence International Conference, Tbilisi, Georgia, June 26-28, 2009, pp. 120-129.

[35] Hazem M. El-bakry, and Nikos Mastorakis, "Fast Packet Detection by using High Speed Time Delay Neural Networks," Proc. of the 10th WSEAS Int. Conference on Multimedia Systems & Signal Processing, Hangzhou University, China, April 11-13, 2010, pp. 222-227.

[36] Christopher Adams, "Does Learning in Prediction Markets Lead to Information Aggregation?", mimeo, Federal Trade Commission, 2006.

[37] Mukhtar Ali, "Probability and Utility Estimators for Racetrack Bettors", Journal of Political Economy, vol. 85, no. 4, 1977, pp. 803-815.

[38] Robert J. Aumann, "Agreeing to Disagree", The Annals of Statistics, vol. 4, no. 6, 1976.

[39] Joyce Berg, Robert Forsythe, Forrest Nelson and Thomas Rietz "Results from a Dozen Years of Election Futures Markets Research," forthcoming in Charles Plott and Vernon Smith (Eds) Handbook of Experimental Economic Results, 2003.

[40] Fisher Black, "Studies of Stock Price Volatility Changes," in Proceedings of the 1976 Meetings of the Business and Economic Statistics Section, American Statistical Association, 1976, pp. 171-181.

[41] Richard Borghesi "Under reaction to New Information: Evidence from an Online Exchange", mimeo, Texas State University, 2006.

[42] Keith Chen, Jonathan E. Ingersoll and Edward H. Kaplan, "Do States Swing Together? Evidence from Political Prediction Markets", mimeo, Yale University, 2006.

[43] Jeff Dominitz, and Charles Manski "How Should We Measure Consumer Confidence," Journal of Economics Perspectives vol. 18, no. 2, 2004, pp. 51-66.

[44] Ray Fair, "Interpreting the Predictive Uncertainty of Presidential Elections", mimeo, Yale University, 2006.

[45] David Fisman, Ray Fisman, Julia Galef and Rakesh Khurana, "Estimating the value of connections to Vice-President Cheney", mimeo, Columbia University, 2006.

[46] Steven Gjerstad, "Risk Aversion, Beliefs, and Prediction Market Equilibrium", mimeo University of Arizona, 2005.

[47] Refet Gürkaynak, and Justin Wolfers, "Macroeconomic Derivatives: An Initial Analysis of Market-Based Macro Forecasts, Uncertainty and Risk", NBER International Seminar on Macroeconomics Annual, 2005.

[48] Milton Harris, and Artur Raviv, "Differences of Opinion Make a Horse Race", The Review of Financial Studies, vol. 6, no. 3, 1993.

[49] Brian Knight, "Are Policy Platforms Capitalized into Equity Prices? Evidence from the Bush/Gore 2000 Presidential Election", Journal of Public Economics, vol. 90, 2006, pp. 4-5.

[50] Andrew Leigh, Justin Wolfers, and Eric Zitzewitz, "What Do Financial Markets Think of War in Iraq?", NBER Working Paper #9587, 2003.

[51] Steven Levitt, "Why Are Gambling Markets Organized So Differently from Financial Markets?", Economic Journal, vol. 114, 2004.

[52] Charles Manski, "Interpreting the Predictions of Prediction Markets", Economic Letters, vol. 91, no. 3, 2006, pp. 425-429.

[53] Paul Milgrom, and Nancy Stokey, "Information, Trade, and Common Knowledge", Journal of Economic Theory, vol. 26, no. 1, 1982.

[54] Stephen Morris, "The Common Prior Assumption in Economic Theory", Economics and Philosophy, vol. 11, 1995.

[55] Stephen Morris, "Speculative Investor Behavior and Learning", Quarterly Journal of Economics, vol. 111, no. 4, 1996.

[56] Marco Ottaviani, and Peter Norman Sørensen, "Aggregation of Information and Beliefs in Prediction Markets", mimeo, London Business School, 2006.

[57] James Sebenius, and John Geanakoplos, "Don't Bet on It: Agreements with Asymmetric Information", Journal of the American Statistical Association, vol. 78, no. 382, 1983.

[58] Servan-Schreiber, Emile, Justin Wolfers, David Pennock and Brian Galebach, “Prediction Markets: Does Money Matter?”, Electronic Markets vol. 14, no. 3, 2004.

[59] Joel Slemrod, and Timothy Greimel, “Did Steve Forbes Scare the Municipal Bond Market?”, Journal of Public Economics, vol. 74, no. 1, 1999, pp. 81-96.

[60] Erik Snowberg, and Justin Wolfers, “Understanding the Favorite-Longshot Bias: Risk Preferences versus Misperceptions”, mimeo, University of Pennsylvania, 2006.

[61] Snowberg, Erik, Justin Wolfers and Eric Zitzewitz, “Partisan Impacts on the Economy: Evidence from Prediction Markets and Close Elections“, Quarterly Journal of Economics, vol. 122, no. 2, 2007.

[62] Paul Tetlock, “How Efficient Are Information Markets? Evidence from an Online Exchange”, mimeo, University of Texas, 2004.

[63] Richard H. Thaler, and William T. Ziemba , “Anomalies: Parimutuel Betting Markets: Racetracks and Lotteries,” Journal of Economic Perspectives, vol. 2, no. 2, 1988.

[64] Justin Wolfers, and Eric Zitzewitz (2004), “Prediction Markets”, Journal of Economic Perspectives, vol. 18, no. 2, 2004, pp. 107-126.

[65] Wolfers, Justin and Eric Zitzewitz, “Five Open Questions About Prediction Markets”, in Robert Hahn and Paul Tetlock (eds) Information Markets: A New Way of Making Decision in the Public and Private Sectors, (Washington DC: AEI-Brookings Press, 2006.

[66] Eric Zitzewitz, “Price Discovery Among the Punters: Using New Financial Betting Markets to Predict Intraday Volatility”, mimeo, Stanford University, 2006.

[67] A. W. Lo, and A. C. Mackinlay, A Non-Random Walk Down Wall Street 5th Ed. Princeton University Press, 2002.

[68] B. Graham, The Intelligent Investor Harper Collins; Rev Ed edition, 2003.

Dr. Hazem M. El-Bakry (Mansoura, Egypt 20-9-1970) received B.Sc. degree in Electronics Engineering, and M.Sc. in Electrical Communication Engineering from the Faculty of Engineering, Mansoura University – Egypt, in 1992 and 1995 respectively. Dr. El-Bakry received Ph. D degree from University of Aizu - Japan in 2007. Currently, he is assistant professor at the Faculty of Computer Science and Information Systems – Mansoura University – Egypt.

His research interests include neural networks, pattern recognition, image processing, biometrics, cooperative intelligent systems and electronic circuits. In these areas, he has published many papers in major international journals and refereed international conferences.

Dr. El-Bakry has the patent No. 2003E 19442 DE HOL / NUR, Magnetic Resonance, SIEMENS Company, Erlangen, Germany, 2003. Furthermore, he is associate editor for journal of computer science and network security (IJCSNS), journal of convergence in information technology (JCIT), International Journal of Digital Content Technology and its Applications (JDCTA), and International Journal of Advancements in Computing Technology (IJACT). Furthermore, he is a referee for IEEE Transactions on Signal Processing, Journal of Applied Soft Computing, the International Journal of Machine Graphics & Vision, Journal of Pattern Recognition, the International Journal of Computer Science and Network Security, and many different international conferences organized by IEEE. In addition, he has been awarded the Japanese Computer & Communication prize in April 2006. He has also been awarded the best paper prize in two international conferences and Mansoura university prize for scientific publication in 2010. Moreover, he has been selected in who is who in Asia 2007 and BIC 100 educators in Africa 2008.

Prof. Nikos Mastorakis is full professor with Technical University of Sofia, Bulgaria.

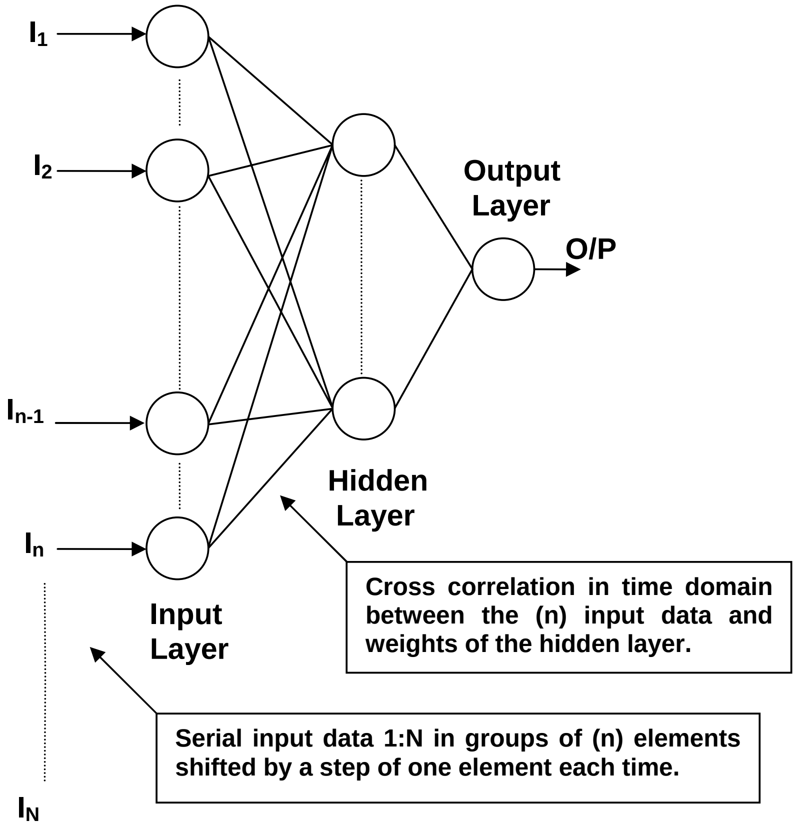
I_1
I_2
I_{n-1}
I_n
I_N
Output Layer
O/P
Hidden Layer
Input Layer
Cross correlation in time domain between the (n) input data and weights of the hidden layer.
Serial input data 1:N in groups of (n) elements shifted by a step of one element each time.

Fig.. 1 TTDNNs for forecasting of stock market prices

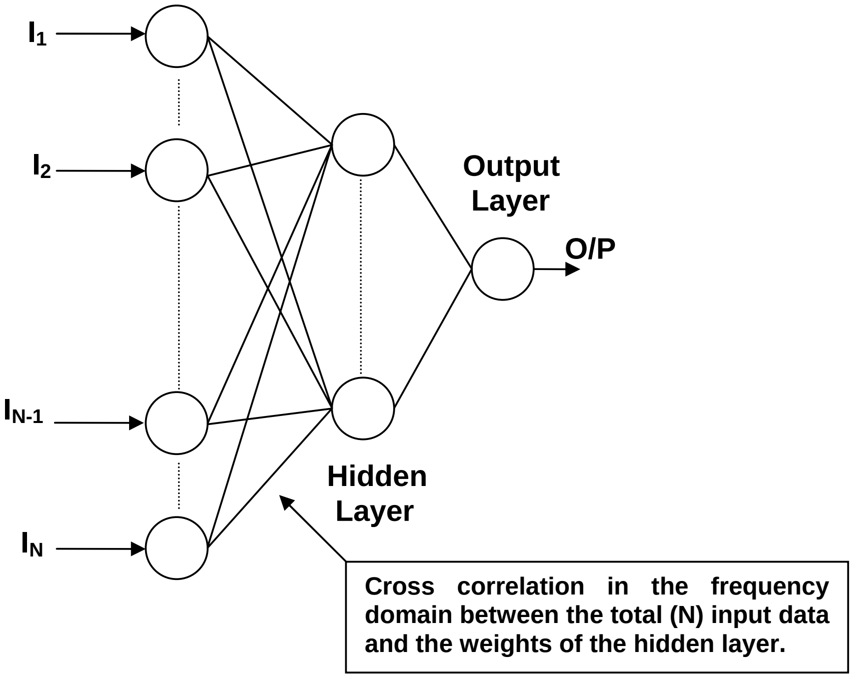
I_1
I_2
I_{N-1}
I_N
Output Layer
O/P
Hidden Layer
Cross correlation in the frequency domain between the total (N) input data and the weights of the hidden layer.

Fig. 2 HSTDNNs for forecasting of stock market prices

TABLE I
THE THEORETICAL SPEED UP RATIO FOR FORECASTING OF STOCK MARKET PRICES (n=400)

Length of serial data	Number of computation steps required for TTDNNs	Number of computation steps required for HSTDNNs	Speed up ratio
10000	2.3014e+008	4.2926e+007	5.3613
40000	0.9493e+009	1.9614e+008	4.8397
90000	2.1478e+009	4.7344e+008	4.5365
160000	3.8257e+009	8.8219e+008	4.3366
250000	5.9830e+009	1.4275e+009	4.1912
360000	8.6195e+009	2.1134e+009	4.0786
490000	1.1735e+010	2.9430e+009	3.9876
640000	1.5331e+010	3.9192e+009	3.9119

TABLE II
THE THEORETICAL SPEED UP RATIO FOR FORECASTING OF STOCK MARKET PRICES (n =625)

Length of serial data	Number of computation steps required for TTDNNs	Number of computation steps required for HSTDNNs	Speed up ratio
10000	3.5132e+008	4.2919e+007	8.1857
40000	1.4754e+009	1.9613e+008	7.5226
90000	3.3489e+009	4.7343e+008	7.0737
160000	0.5972e+010	8.8218e+008	6.7694
250000	0.9344e+010	1.4275e+009	6.5458
360000	1.3466e+010	2.1134e+009	6.3717
490000	1.8337e+010	2.9430e+009	6.2306
640000	2.3958e+010	3.9192e+009	6.1129

TABLE III
THE THEORETICAL SPEED UP RATIO FOR FORECASTING OF STOCK MARKET PRICES (n =900)

Length of serial data	Number of computation steps required for TTDNNs	Number of computation steps required for HSTDNNs	Speed up ratio
10000	4.9115e+008	4.2911e+007	11.4467
40000	2.1103e+009	1.9612e+008	10.7600
90000	4.8088e+009	4.7343e+008	10.1575
160000	0.8587e+010	8.8217e+008	9.7336
250000	1.3444e+010	1.4275e+009	9.4178
360000	1.9381e+010	2.1134e+009	9.1705
490000	2.6397e+010	2.9430e+009	8.9693
640000	3.4493e+010	3.9192e+009	8.8009

TABLE IV
PRACTICAL SPEED UP RATIO FOR FORECASTING OF STOCK MARKET PRICES

Length of serial data	Speed up ratio (n=400)	Speed up ratio (n=625)	Speed up ratio (n=900)
10000	8.94	12.97	17.61
40000	8.60	12.56	17.22
90000	8.33	12.28	16.80
160000	8.07	12.07	16.53
250000	7.95	17.92	16.30
360000	7.79	11.62	16.14
490000	7.64	11.44	16.00
640000	7.04	11.27	15.89