Expert Systems with Applications 38 (2011) 10389–10397

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Using artificial neural network models in stock market index prediction

Erkam Guresen [a], Gulgun Kayakutlu [a,*], Tugrul U. Daim [b]

[a] Istanbul Technical University, Istanbul, Turkey
[b] Portland State University, Portland OR, USA

ARTICLE INFO

Keywords:
Financial time series (FTS) prediction
Recurrent neural networks (RNN)
Dynamic artificial neural networks (DAN2)
Hybrid forecasting models

ABSTRACT

Forecasting stock exchange rates is an important financial problem that is receiving increasing attention. During the last few years, a number of neural network models and hybrid models have been proposed for obtaining accurate prediction results, in an attempt to outperform the traditional linear and nonlinear approaches. This paper evaluates the effectiveness of neural network models which are known to be dynamic and effective in stock-market predictions. The models analysed are multi-layer perceptron (MLP), dynamic artificial neural network (DAN2) and the hybrid neural networks which use generalized autoregressive conditional heteroscedasticity (GARCH) to extract new input variables. The comparison for each model is done in two view points: Mean Square Error (MSE) and Mean Absolute Deviate (MAD) using real exchange daily rate values of NASDAQ Stock Exchange index.

1. Introduction

Forecasting simply means understanding which variables lead to predict other variables (Mcnelis, 2005). This means a clear understanding of the timing of lead-lag relations among many variables, understanding the statistical significance of these lead-lag relations and learning which variables are the more important ones to watch as signals for predicting the market moves. Better forecasting is the key element for better financial decision making, in the increasing financial market volatility and internationalized capital flows.

Accurate forecasting methods are crucial for portfolio management by commercial and investment banks. Assessing expected returns relative to risk presumes that portfolio strategists understand the distribution of returns. Financial expert can easily model the influence of tangible assets to the market value, but not intangible asset like know-how and trademark. The financial time series models expressed by financial theories have been the basis for forecasting a series of data in the twentieth century.

Studies focusing on forecasting the stock markets have been mostly preoccupied with forecasting volatilites. There has been few studies bringing models from other forecasting areas such as technology forecasting.

To model the market value, one of the best ways is the use of expert systems with artificial neural networks (ANN), which do not contain standard formulas and can easily adapt the changes of the market. In literature many artificial neural network models are evaluated against statistical models for forecasting the market value. It is observed that in most of the cases ANN models give better result than other methods. However, there are very few studies comparing the ANN models do among themselves, where this study is filling a gap.

Objective of this study is to compare performance of most recent ANN models in forecasting time series used in market values. Autoregressive Conditional Heteroscedasticity (ARCH) model (Engle, 1982), generalized version of ARCH model Generalized ARCH (GARCH) model (Bollerslev, 1986), Exponential GARCH (EGARCH) model (Nelson, 1991) and Dynamic Architecture for Artificial Neural Networks (DAN2).

Ghiassi and Saidane (2005) will be analyzed in comparison to classical Multi-Layer Perceptron (MLP) model. Despite the popularity and implementation of the ANN models in many complex financial markets directly, shortcomings are observed. The noise that caused by changes in market conditions, it is hard to reflect the market variables directly into the models without any assumptions (Roh, 2007). That is why the new models will also be executed in hybrid combination with MLP. The analysed models will be tested on NASDAQ index data for nine months and the methods will be compared by using Mean Square Error (MSE) and Mean Absolute Deviation (MAD).

The remaining sections of this paper are organized as follows: Section 2 gives the background of the related studies; Section 3 introduces the models used in this study and Section 4 provides results of each model using daily exchange rates of NASDAQ index. Final section gives the conclusion and recommendations for future researches.

This study will not only make contribution to the ANN research but also to the business implementations of market value calculation.

* Corresponding author.
E-mail address: gkayakutlu@gmail.com (G. Kayakutlu).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.02.068

2. Background

2.1. Time series forecasting and ANN

The financial time series models expressed by financial theories have been the basis for forecasting a series of data in the twentieth century. Yet, these theories are not directly applicable to predict the market values which have external impact. The development of multi layer concept allowed ANN (Artificial Neural Networks) to be chosen as a prediction tool besides other methods. Various models have been used by researchers to forecast market value series by using ANN. A brief literature survey is given in Table 1.

Gooijer and Hyndman (2006) reviewed the papers about time series forecasting from 1982 to 2005. It has been prepared for the silver jubilee volume of international journal of forecasting, for the 25th birthday of International Institute of Forecasters (IIF). In this review statistical and simulation methods are analyzed to include exponential smoothing, ARIMA, seasonality, state space and structural models, nonlinear models, long memory models, ARCH-GARCH. Gooijer and Hyndman (2006) compiled the reported advantages and disadvantages of each methodology and pointed out the potential future research fields. They also denoted existence of many outstanding issues associated with ANN utilisation and implementation stating when they are likely to outperform other methods. Last few years researches are focused on improving the ANN's prediction performance and developing new artificial neural network architecture.

Engle (1982) suggested the Autoregressive Conditional Heteroscedasticity (ARCH) model, Bollerslev (1986) generalized the ARCH model and proposed the Generalized ARCH (GARCH) model for time series forecasting. By considering the leverage effect limitation of the GARCH model, the Exponential GARCH (EGARCH) model was proposed by Nelson (1991). Despite the popularity and implementation of the ANN models in many complex financial markets directly, shortcomings are observed. The noise that caused by changes in market conditions, it is hard to reflect the market variables directly into the models without any assumptions (Roh, 2007).

Preminger and Franck (2007) used a robust linear autoregressive and a robust neural network model to forecast exchange rates. Their robust models were better than classical models but still are not better than Random Walk (RW). Roh (2007) used classical ANN and EWMA (Exponentially Weighted Moving Average), GARCH and EGARCH models with ANN. NN-EGARCH model outperforms the other models with a 100% hit ratio for smaller forecasting period than 10 days.

Kumar and Ravi (2007) reviews 128 papers about bankruptcy prediction of banks and firms. This review shows that ANN methods outperforms many methods and hybrid systems can combine the advantages of different methods. Ghiassi, Saidane, and Zimbra (2005) evaluated ANN, ARIMA and DAN2 (Dynamic Architecture for Artificial Neural Networks) using popular time series in literature. DAN2, is a new NN architecture first developed by Ghiassi and Saidane (2005), clearly outperforms the other methods. DAN2 is pure feed forward NN architecture and detailed information about this architecture will be given in Section 5.

Menezes and Nikolaev (2006) used a new NN architecture and named it Polynomial Genetic Programming. It is based on Polynomial Neural Network first developed by Ivakhnenko (Menezes & Nikolaev, 2006). This architecture uses polynomials to build an ANN. Menezes and Nikolaev (2006) uses genetic algorithm to estimate ANN parameters such as starting polynomials, weight estimation etc. This study gives better result for some problems. It is a new promising architecture but it needs improvement (Menezes & Nikolaev, 2006).

Zhang and Wan (2007) developed a new ANN architecture Statistical Fuzzy Interval Neural Network based on Fuzzy Interval Neural Network. JPY/USD and GBP/USD exchanges rates are predicted using these methods. These methods are developed to predict only an interval not a point in time. Hassan, Nath, and Kirley (2007) used a hybrid model including Hidden Markov Model, ANN and Genetic Algorithm. They test hybrid model on stock exchange rates. Hybrid model is proven to be better than simulation models.

Yu and Huarng (2008) used bivariate neural networks, bivariate neural network-based fuzzy time series, and bivariate neural network-based fuzzy time series model with substitutes to apply neural networks to fuzzy time series forecasting. Bivariate neural network-based fuzzy time series model with substitutes performs the best. Zhu, Wang, Xu, and Li (2008) used basic and augmented neural network models to show trading volume can improve the prediction performance of neural networks. Leu, Lee, and Jou (2009) compared radial basis-function neural network (RBFNN), random walk, and distance-based fuzzy time series models with daily closing values of TAIEX, and exchange rates NTD/USD, KRW/USD, CNY/USD, JPY/USD. Results show that RBFNN outperformed the random walk model and the artificial neural network model in terms of mean square error. Cheng, Chen, and Lin (2010) used PNN (Prbobabilistic NN), rough sets, and hybrid model (PNN, Rough Set, C 4.5 Decision Tree) to integrate fundamental analysis and technical analysis to build up a trading model of stock market timing. They report that hybrid model is helpful to construct a better predictive power trading system for stock market timing analysis. Chang, Liu, Lin, Fan, and Ng (2009) used an integrated system (CBDWNN) which combines dynamic time windows, case based reasoning (CBR), and neural network (NN). Their CBDWNN model outperformed other compaired methods, and very informative and robust for average investors.

Egrioglu, Aladag, Yolcu, Uslu, and Basaran (2009) introduced a new method which is based on feed forward artificial neural networks to analyze multivariate high order fuzzy time series forecasting models. Khashei and Bijari (2010) compaired autoregressive integrated moving average (ARIMA), artificial neural networks (ANNs), and Zhang's hybrid model. And Hybrid model outperforms the other models. Hamzacebi, Akay, and Kutay (2009) compaired ARIMA and ANN and conclude that direct forecast with ANN is better and noted that before generalizing the conclusion other researchs should be done. Majhi, Panda, and Sahoo (2009) compaired functional link artificial neural network (FLANN), cascaded functional link artificial neural network (CFLANN),and LMS model and observed that the CFLANN model performs the best followed by the FLANN and the LMS models.

Liao and Wang (2010) used stochastic time effective neural network model to shows some predictive results on the global stock indices and their model is showed predictive results. Atsalakis and Valavanis (2009a) used Adaptive Neuro Fuzzy Inference System (ANFIS) to determine the best stock trend prediction model and results show that ANFIS clearly demonstrates the potential of neurofuzzy based modeling for financial market prediction. Chen, Ying, and Pan (2010) also used ANFIS to predict monthly tourist arrivals. And conclude that ANFIS performs better than markov and fuzzy models. Bildirici and Ersin (2009) combined ANNs with ARCH/GARCH, EGARCH, TGARCH, PGARCH, APGARCH. This combined models better perfomred than ANNs or GARCH based models. Guresen and Kayakutlu (2008) used hybrid models like GARCH-DAN2 and EGARCH-DAN2 to forecast Istanbul Stock Exchange Index (ISE XU100). Yudong and Lenan (2009) used bacterial chemotaxis optimization (BCO), and back propagation neural network (BPNN) on S&P 500 index and conclude that their hybrid model (IBCO–BP) model offers less computational complexity, bet-

Table 1
Financial time series researches (ANN and hybrid models).

Date	Researchers	Used method	Data years	Data type	Goal	Predicted period	Results
2005	Ghiassi, Saidane & Zimbra	ANN, ARIMADAN2		Time series from literature	To compare the methods		DAN2, is an alternative of ANN & gives better result & needs to only choose the inputs
2005	Yümlü, Gürgen & Okay	Mixture of Experts (MoE) MLP, RNN, EGARCH	1990–2002	ISE XU100 daily values	Exchange prediction & To compare the methods	4 years	MoE outperforms the other modelsEGARCH is outperformed by all other methods
2006	Menezes & Nikolaev	Genetic Programming (GP) Polynomial Genetic Programming		Time series from literature	To compare the methods		Find the polynomials in time series & promising for future researches
2007	Preminger & Franck	Robust Liner AutoregressiveRobust Neural Network	1971–2004	GBP/USDJPY/USD	Better forecasting	1–3–6 months	Robust models are better than start models but still are not better than RW (Random Walk)
2007	Hamzaçebi & Bayramoğlu	ARIMA ANN	2202–2006	ISE-XU100	To compare ARIMA & ANN	Daily	ANN has better results
2007	Pekkaya and Hamzaçebi	LR (Linear regression) ANN	1999–2006	YTL/USD	To compare the forecasts using macro variables	Monthly	ANN gives better results & predicts two important breaking point with 6.611 % error
2007	Roh	ANN, EWMA, GARCH, EGARCH	930 tradedays	KOSPI 200	To compare ANN with hybrid models	Daily	NN-EGARCH & NN-GARCH; for periods shorter than a month 100 % direction prediction and periods shorter than 160 days min 50 % direction prediction
2007	Kumar & Ravi	ANN, Fuzzy Logic, Cased-Based Reasoning, Decision Trees, Rough Sets (RS)			Review-Bankruptcy prediction (128 paper)		RS based models outperform logistic regression & decision tree. Logistic regression, LDA, QDA, FA clearly outperformed by ANN. Hybrid methods can combine the advantages of methods
2007	Celik and Karatepe	ANN	1989–2004	Banking sector data series	Crises prediction		Financial ratios successfully predicted for 4 months
2007	Zhang & Wan	Fuzzy Interval NN (FINN)	1998–2001	JPY/USDGBP/USD	Exchange prediction	6 weeks	Promising for future researches
2007	Hassan, Nath & Kirley	Hidden Markov Model (HMM),ANN, Genetic Algorithm (GA)	2003–2004	Stocks; Apple, IBM, Dell	Exchange prediction	5 weeks	Hybrid model is better than HMM & ARIMA
2008	Yu & Huarng	Bivariate NN, Bivariate NN-based fuzzy time series, Bivariate NN-based fuzzy time series model with substitutes	1999	Daily closing values of TAIEX & TAIFEX	Appling neural networks to fuzzy time series forecasting	Daily	Bivariate neural network-based fuzzy time series model with substitutes performs the best, Bivariate neural network-based fuzzy time series performs the worst
2008	Zhu, Wang, Xu, & Li	Basic & augmented neural network models	1989–2005	NASDAQ, DJIA & STI indices	To investigate effect of trading volume on prediction with ANN	Daily, weekly & monthly	It is possible to modestly or significantly improve the network performance by adding trading volume
2009	Leu, Lee & Jou	Radial basis-function neural network (RBFNN), Random walk, Distance-based fuzzy time series	2006–2007	TAIEX, NTD/USD, KRW/USD, CNY/USD, JPY/USD	Index prediction & To compare the methods	1,3,5 & 7 days	RBFNN outperformed the random walk model & the artificial neural network model in terms of mean square error
2010	Cheng, Chen,& Lin	PNN (Probabilistic NN), Rough Sets, Hybrid (PNN, Rough Set, C 4.5 Decision Tree)	1988–2005	Monthly Taiwan weighted stock index	To integrate fundamental analysis & technical analysis on a trading model	1, 3, 6 & 12 months	PNN, rough sets & C4.5 classifiers to generate trading rule sets, which is helpful to construct a better predictive power trading system for stock market timing analysis
2009	Chang, Liu, Lin, Fan, & Ng	An integrated system (CBDWNN); combining dynamic time windows, case based reasoning (CBR), & neural network (NN)	2004–2006	Daily values of nine different stocks	Efficient forecasting model for making the buy/selldecisions	Daily	The CBDWNN in this study is outperforming other two methods, and very informative and robust for average investors
2009	Egrioglu, Aladag, Yolcu, Uslu, & Basaran	A new method which is based on feed forward artificial neural networks to analyze multivariate high order fuzzy time series forecasting models	1974–2004	Annual car road accidents casualties in Belgium	A new method that does not require use of fuzzy logic relation tables to determine fuzzy relationships	Annually	The proposed method provides forecasts with a smaller AFER value than ones obtained from the methods in literature
2010	Khashei & Bijari	Auto-regressive integrated moving average (ARIMA), Artificial neural networks (ANNs), Zhang's hybrid		Wolf's sunspot, Canadian lynx, GBP/USD	To demonstrate the appropriateness & effectiveness of the proposed models.		Zhang's hybrid model outperforms ARIMA, & ANNs

(continued on next page)

Table 1 (*continued*)

Date	Researchers	Used method	Data years	Data type	Goal	Predicted period	Results
		model					
2009	Hamzacebi, Akay, & Kutay	ARIMA ANN (both for direct & iterative forecast)		Time series used in literature	To find the method which gives a better result		They claim the superiority of the direct method, however other researches' are necessary to generalize the conclusion
2009	P& a & Sahoo	Functional link artificial neural network (FLANN), Cascaded functional link artificial neural network (CFLANN), LMS model.		USD to GBP, Indian Rupees & Japanese Yen	To evaluate the performance of the proposed models		It is observed that the CFLANN model performs the best followed by the FLANN & the LMS models.
2010	Liao & Wang	Stochastic time effective neural network model	1990–2008	SAI, SBI, HSI, DJI, IXIC & SP500	To shows some predictive results on the global stock indices	Daily	Stochastic time effective neural network model shows some predictive results on the global stock indices.
2009	Atsalakis, & Valavanis	Adaptive Neuro Fuzzy Inference System (ANFIS)		Ten stocks from Athens & NYSE	To determine the best stock trend prediction model	Daily	Proposed system clearly demonstrates the potential of neurofuzzy based modeling for financial market prediction
2010	Chen, Ying, & Pan	Adaptive network-based fuzzy inference system (ANFIS)	1989–2000	Tourist arrivals to Taiwan,Hong Kong, USA & Germany	To demonstrate the forecasting performance of ANFIS	Monthly	The ANFIS model yield more accurate tourist arrivals forecasting than that of the Markov, GM & Fuzzy
2009	Bildirici & Ersin	ARCH/GARCH, EGARCH, TGARCH, PGARCH, APGARCH, ANN	1987–2008	ISE-XU100	To improve forecasts with ANNs	Daily	ANN models provide significant improvement in forecasts.
2009	Yudong & Lenan	Bacterial chemotaxis optimization (BCO), Back propagation neural network (BPNN)	1998–2008	Standard's & Poor's 500 (S& P 500)	Eficient forecasting model for prediction of stock indices	Daily	The IBCO–BP model offers less computational complexity, better prediction accuracy, & less training time

ter prediction accuracy, and less training time. Atsalakis and Valavanis (2009b) surveyed more than 100 related published articles which focused on neural networks and neuro-fuzzy techniques derived and applied to forecast stock markets.

This literature survey shows that ANN models generally outperform other methods when applied on time series Further, new architectures and Hybrid models are promising but only DAN2 clearly outperforms all compared models (Ghiassi & Saidane, 2005; Ghiassi et al., 2005; Ghiassi, Zimbra, & Saidane, 2006).

2.2. Forecasting market values

Currently many forecasts are established by the analysts working for different financial institutions in the US (Ramnath, Rock, & Shane, 2008). Academic forecasts have also been published.

Prior research attempted forecasting US stock market indices or US financial metrics in the past. A group literature focused on understanding volatilities. Maltritz and Eichler (2010) focus on the American depository receipts in order to forecast crises and changes in the exchange rates. Ando (2009) through use of Bayesian theory developed a new portfolio selection method. Scharth and Medeiros (2009) attempted to forecast volatilities in the Dow Jones index by using a combination of regression trees and smooth transition. Another related study by Chen and So (2006) used a threshold heteroscedastic model to forecast similar volatilities. Wang, Keswani, and Taylor (2006) explored the role of sentiment in forecasting volatilities.

Another group of studies used different group of methods for forecasting the stock market. They brought tools used for technology forecasting into financial forecasting with some good success. Lee, Lee, and Oh (2005) used Lotka-Volterra model which is really based on prey predator relationship and used for forecasting the diffusion of competing technologies. They applied it to the Korean Stock Market.

3. ANN models used in time series forecasting

3.1. Multilayer perceptron (MLP)

The multilayer perceptron is one of the most widely implemented neural network topologies. In terms of mapping abilities, the MLP is believed to be capable of approximating arbitrary functions (Principe, Euliano, & Lefebvre, 1999). This has been important in the study of nonlinear dynamics, and other function mapping problems.

Two important characteristics of the multilayer perceptron are: its nonlinear processing elements (PEs) which have a nonlinearity that must be smooth (the logistic function and the hyperbolic tangent are the most widely used); and their massive interconnectivity, i.e. any element of a given layer feeds all the elements of the next layer (Principe et al., 1999).

MLPs are normally trained with the backpropagation algorithm (Principe et al., 1999). The backpropagation rule propagates the errors through the network and allows adaptation of the hidden PEs. The multilayer perceptron is trained with error correction learning, which means that the desired response for the system must be known.

Error correction learning works in the following way: From the system response at PE i at iteration n, y_i(n), and the desired response d_i(n) for a given input pattern an instantaneous error ε_i(n) is defined by

$$\varepsilon_i(n) = d_i(n) - y_i(n). \tag{1}$$

Using the theory of *gradient descent learning*, each weight in the network can be adapted by correcting the present value of the weight with a term that is proportional to the present input and error at the weight, i.e.

$$w_{ij}(n+1) = w_{ij}(n) + \eta\delta_i(n)x_j(n). \tag{2}$$

The local error $\delta_i(n)$ can be directly computed from $\varepsilon_i(n)$ at the output PE or can be computed as a weighted sum of errors at the internal PEs. The constant η is the step size and called the learning rate. This procedure is called the backpropagation algorithm.

Backpropagation computes the sensitivity of a cost functional with respect to each weight in the network, and updates each weight proportional to the sensitivity. The beauty of the procedure is that it can be implemented with local information and requires just a few multiplications per weight, which is very efficient. Because this is a gradient descent procedure, it only uses the local information so can be caught in local minima. Moreover, the procedure is inherently noisy since we are using a poor estimate of the gradient, causing slow convergence (Principe et al., 1999).

Momentum learning is an improvement to the straight gradient descent in the sense that a memory term (the past increment to the weight) is used to speed up and stabilize convergence. In momentum learning the equation to update the weights becomes

$$w_{ij}(n+1) = w_{ij}(n) + \eta\delta_i(n)x_j(n) + \alpha(w_{ij}(n) - w_{ij}(n-1)), \tag{3}$$

where α is the momentum. Normally α should be set between 0.1 and 0.9.

Training can be implemented in two ways: Either we present a pattern and adapt the weights (on-line training), or we present all the patterns in the input file (an epoch), accumulate the weight updates, and then update the weights with the average weight update. This is called batch learning. To start backpropagation, loading an initial value for each weight (normally a small random value) is needed, and proceeding until some stopping criterion is met. The three most common are: to cap the number of iterations, to threshold the output mean square error, or to use cross validation. Cross validation is the most powerful of the three since it stops the training at the point of best generalization (i.e. the performance in the test set) is obtained (Principe et al., 1999). To implement cross validation one must put aside a small part of the training data and use it to see how the trained network is doing (e.g. every 100 training epochs, test the net with a validation set). When the performance starts to degrade in the validation set, training should be stopped (Alpaydın, 2004; Haykin, 1999; Principe et al., 1999).

Measuring the progress of learning is fundamental in any iterative training procedure. The learning curve (how the mean square error evolves with the training iteration) is such a quantity. The difficulty of the task and how to control the learning parameters can be judged from the learning curve. When the learning curve is flat, the learning rate should be increased to speed up learning. On the other hand, when the learning curve oscillates up and down, the step size should be decreased. In the extreme, the error can go steadily up, showing that learning is unstable. At this point the network should be reset. When the learning curve stabilizes after many iterations at an error level that is not acceptable, it is time to rethink the network topology (more hidden PEs or more hidden layers, or a different topology altogether) or the training procedure (other more sophisticated gradient search techniques).

Principe et al. (1999) present below a set of heuristics that will help decrease the training times and, in general, produce better performance;

- Normalizing training data,
- Using the tanh nonlinearity instead of the logistic function.
- Normalizing the desired signal to be just below the output nonlinearity rail voltages (i.e. when using the tanh, the desired signals of +/− 0.9 instead of +/− 1).
- Setting the step size higher towards the input (i.e. for a one hidden layer MLP, set the step size at 0.05 in the synapse between the input and hidden layer, and 0.01 in the synapse between the hidden and output layer).
- Initializing the net's weights in the linear region of the nonlinearity (dividing the standard deviation of the random noise source by the fan-in of each PE).
- Using more sophisticated learning methods (quick prop or delta bar delta).
- Always having more training patterns than weights. It can be expected that the performance of the MLP in the test set to be limited by the relation N > W/ε, where N is the number of training epochs, W the number of weights and e the performance error. The MLP should be trained until the mean square error is less than $\varepsilon/2$.

3.2. Dynamic architecture for artificial neural networks (DAN2)

This model is developed by Ghiassi and Saidane (2005) and compared with the classical ANN models using a known time series (Ghiassi et al., 2005). Fig. 1 shows the structure of DAN2.

The algorithm steps given by Ghiassi and Saidane (2005) are as follows:

For input matrix $X = \{X_i; i = 1,2\ldots,n\}$ as n independent records of m attributes let $X_i = \{x_{ij}; j = 1,2,\ldots,m\}$, and the reference vector $R = \{rj; j = 1,2,\ldots,m\}$

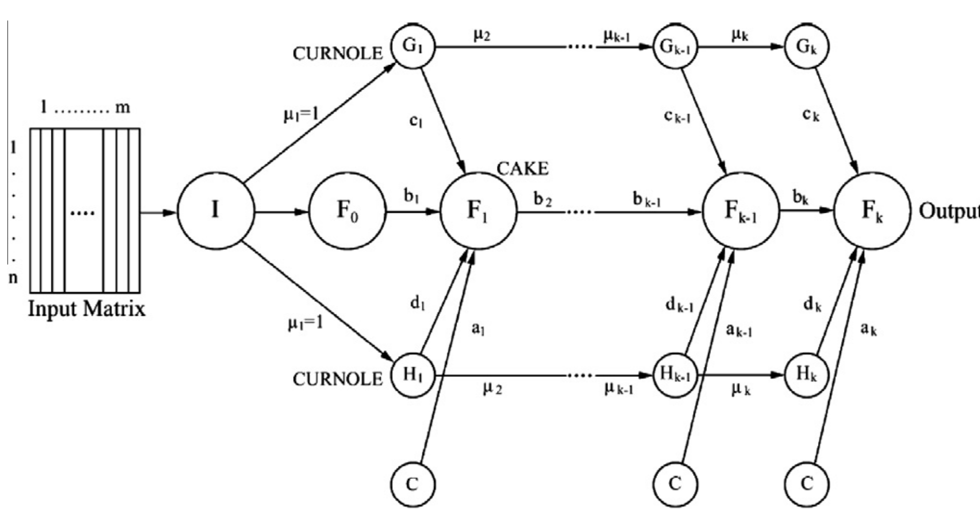

Fig. 1. The DAN2 network architecture (Ghiassi & Saidane, 2005).

1. The initial linear layer:

$$F_0(X) = a_0 + \sum_j b_{0j}x_{ij}. \tag{3}$$

2. Subsequent hidden layers' CAKE node at iteration k:

$$F_k(X_i) = a_k + b_k F_{k-1}(X_i) + c_k G_k(Xi) + d_k H_k(Xi). \tag{4}$$

3. The CURNOLE node's input and transfer function at iteration k ($k = 1,2,\ldots,K$; where K is the maximum sequential iterations or number of hidden layers) is defined as:
   (a) Specify a random set of m constant representing the "reference" vector R (default $r_j = 1$ for all $j = 1,2,\ldots,k,m$).
   (b) For each input record X_i, compute the scalar product:

$$R \times X = \sum_j r_j x_{ij}. \tag{5}$$

   (c) Compute the length (norm) of the vector R and a record vector

$$X_i : \|R\| = \sqrt{\sum_j r_j^2}; \quad \|X_i\| = \sqrt{\sum_j x_{ij}^2}; \tag{6}$$

   (d) Normalize $R \times X$ to compute

$$(R \times X)_N = (R \times X_i) = (R \times X_i)/(\|R\| \times \|X_i\|). \tag{7}$$

   Recall that:

$$(R \times X_i)_N = (\|R\| \times \|X_i\|) \times \cos(angle(R, X_i)), \tag{8}$$

   thus,

$$\cos ine(angle(R, X_i) = (R \times X_i)/(\|R\| \times \|X\|_i) = (R \times X_i)_N. \tag{9}$$

   (e) For i = 1,2,…,n; compute

$$angle(R, X_i) = \arccos(R \times X_i)_N = \alpha_i. \tag{10}$$

   (f) Compute the transferred nonlinear component of the signal as: $G_k(X_i) = \cos(\mu_k \times \alpha_i), H_k(X_i) = \sin(\mu_k \times \alpha_i)$, μ_k is a constant multiplier for iteration k.
   (g) Replacing $G_k(X_i)$ and $H_k(X_i)$ in Eq. 5.9 will result

$$F_k(X_i) = a_k + b_k F_{k-1}(X_i) + c_k \cos(\mu_k \times \alpha_i) + d_k \sin(\mu_k \times \alpha_i). \tag{11}$$

Data normalization in DAN2 can be represented by the trigonometric function cos ($\mu_k \times \alpha_i + \theta$). At each layer vector R is rotated and shifted to minimize the resulting total error.

If the model training stops too early, the network is said to be *under-trained* or *under-fitted*. An under-trained model often has high SSE values for either or both the training and validation data sets. Under-training often occurs when there are insufficient data for model fitting. DAN2 uses $\varepsilon_1 = (SSE_k - SSE_{k-1})/SSE_k \leqslant \varepsilon_1^*$ to assess existence or absence of under-training in the models (Ghiassi & Saidane, 2005). Over-training or over-fitting is a more common problem in neural net modeling. A neural network modeler considered over-fitted (over-trained) when the network fits the in sample data well but produces poor out-of-sample results. To avoid overfitting, (Ghiassi & Saidane, 2005) divide the available in-sample data into the training and validation data sets. At each iteration k, ($k > 1$), they compute MSE values for both the training (MSE_T) and validation (MSE_V) sets and they use $\varepsilon_2 = |MSE_T - MSE_V|/MSE_T \leqslant \varepsilon_2^*$ to guard against over-fitting. The modeler should consider fully trained when the user specified accuracy criteria and the over fitting constraint are both satisfied. The accuracy levels ε_1^* and ε_2^* are problem dependent and should determined experimentally (Ghiassi & Saidane, 2005).

3.3. GARCH-MLP models

Autoregressive conditional heteroscedasticity (ARCH) model considers the variance of the current error term to be a function of the variances of the previous time period's error terms. ARCH relates the error variance to the square of a previous period's error. If an autoregressive moving average model (ARMA model) is assumed for the error variance, the model is a *generalized autoregressive conditional heteroskedasticity* (GARCH). In that case, the GARCH (p,q) model (where p is the order of the GARCH terms σ^2 and q is the order of the ARCH terms ε^2) is given by

$$\begin{aligned}\sigma_t^2 &= \alpha_0 + \alpha_1\varepsilon_{t-1}^2 + \ldots + \alpha_q\varepsilon_{t-q}^2 + \beta_1\sigma_{t-1}^2 + \ldots + \beta_p\sigma_{t-p}^2 \\ &= \alpha_0 + \sum_{i=1}^{q}\alpha_i\varepsilon_{t-i}^2 + \sum_{i=1}^{p}\beta_i\sigma_{t-i}^2.\end{aligned} \tag{12}$$

Most of the financial series models are known to be easily modelled by GARCH (1,1), so this research uses the extracted variables from GARCH (1,1) as Roh suggests (Roh, 2007). The GARCH (1,1) has the following formula:

$$\sigma_t^2 = \alpha_0 + \alpha_1\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2, \tag{13}$$

where σ_t is volatility at t, α_0 is the nonconditional volatility coefficient, ε_{t-1}^2 residual at $t-1$, σ_{t-1}^2 is the variance at $t-1$.

The newly extracted variables are as follows (Roh, 2007):

$$\sigma_t^{2'} = \beta_1\sigma_{t-1}^2, \tag{14}$$

$$\varepsilon_{t-1}^2 = \alpha_1\varepsilon_{t-1}^2. \tag{15}$$

We use these new variables as additional inputs for every type of ANN given above.

3.4. EGARCH–MLP Models

EGARCH has the leverage effect with the following formula:

$$\ln\sigma_t^2 = \alpha + \beta\ln\sigma_{t-1}^2 + \gamma\left(\left|\frac{\varepsilon_{t-1}}{\sigma_{t-1}} - \sqrt{\frac{2}{\pi}}\right|\right) + \omega\left(\frac{\varepsilon_{t-1}}{\sigma_{t-1}}\right), \tag{16}$$

where α is the nonconditional variance coefficient, $\ln\sigma_t^2$ is the log value of variance at $t-1$, $(|\varepsilon_{t-1}/\sigma_{t-1} - \sqrt{(2/\pi)}|)$ is the asymmetric shock by leverage effect, and $(\varepsilon_{t-1}/\sigma_{t-1})$ is the leverage effect. The newly extracted variables are as follows (Roh, 2007):

$$\ln\sigma_{t-1}^{2'} = \beta\ln\sigma_{t-1}^2, \tag{17}$$

$$\text{LE (leverage effect)} = \gamma\left(\left|\frac{\varepsilon_{t-1}}{\sigma_{t-1}} - \sqrt{\frac{2}{\pi}}\right|\right), \tag{18}$$

$$\text{L (leverage)} = \omega\left(\frac{\varepsilon_{t-1}}{\sigma_{t-1}}\right). \tag{19}$$

3.5. Model performance measures: MSE and MAD

In literature, mean square error (MSE) and mean absolute deviate (MAD) are generally used for evaluating performances of ANN's. MSE and MAD are obtained by the following formulas:

$$\text{MSE} = \frac{1}{n}\sum_k (r_k - y_k)^2, \tag{20}$$

$$\text{MAD} = \frac{1}{n}\sum_k |r_k - y_k|, \tag{21}$$

where y_k is the actual output value of kth observation, r_k is the output value of the ANN's obtained from kth observation and n is the number of observations.

In this study another form of MAD is used. MAD % is mean absolute deviate in percentage. When the descriptive statistics is analyzed it is easy to see that training data is between 1268.64 and 1844.25 (range is 43194.05). But the test data is between 1664.19 and 1867.32 (range is 16889.10). Thus to clarify the errors between training and testing, MAD % values are given. MAD % is obtained by the following formula:

$$\mathrm{MAD} = \frac{1}{n}\sum_{k}\frac{|r_k - y_k|}{y_k} \times 100. \tag{22}$$

In order to facilitate the comparison of training and testing data performance of the models, MAD % values are used.

4. Case: forecasting NASDAQ index

In this research daily stock exchange rates of NASDAQ from October 7, 2008 to June 26, 2009 are used. First 146 days are used for training and cross validation and last 36 used for testing. For hybrid models also new variables extracted from GARCH and EGARCH are calculated using MS Excel. Since EGARCH showed no asymmetric shocks (gives two inputs values as 0 for EGARCH-ANN), the model is eliminated from the comparisons. For MLP and GARCH-MLP NeuroSolutions 5.06 software is used. For calculating DAN2 and GARCH-DAN2 MS Excel is used.

MLP and DAN2 uses last four days to forecast the fifth date while GARCH-MLP and GARCH-DAN2 uses last four days and additional two inputs calculated from GARCH model. While MLP models uses data one at a time, DAN2 uses all data at once to calculate model parameters.

Training data is given in Fig. 2 and the forecast period with tests is shown in Fig. 3.

As it is observed in Table 2, the analysis shows that DAN2 gives the best MSE and MAD in test data MLP gives the best results in the training data.

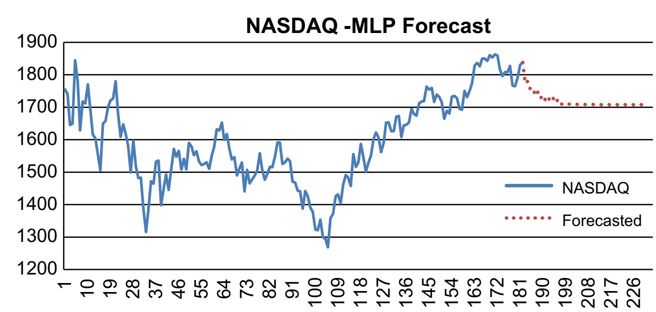

Fig. 2. Training data of NASDAQ index and MLP forecast.

5. Results & discussion

The fact that the GARCH-DAN2 gave unexpectedly high results that is far from all the rest of the methods both in training and test data forced us to look into the regression model of DAN2. Unfortunately, DAN2 has inconsistencies in the regression model as shown in Table 3.

The inconsistencies in DAN2 can be summarized as follows:

1. If the starting process (regression) of DAN2 forecast well, the remaining structure of DAN2 can take the forecasts one step further.
2. There is multiple input but single output in the model since it is based on multi-variable regression. So DAN2 cannot be used for multiple output problems.
3. DAN2 does not carry adaptivity feature of ANN models, which lasts in remodelling when the training data changes. As a result when time passes and new data become available for time series forecasting and a new forecast needed, the entire DAN2 model should be thrown away and a new DAN2 model must be established.
4. Reliability tests on input data, which is a necessity due to the regression starter, does not exist.
5. Each layer has to be controlled by reliability tests since it is the input of the following layers regression model. DAN2 can have a dynamic architecture by removing insignificant layer connections.
6. Since this study and other studies shows that some significance test should be done for validity of the model, it contradicts with input-out mapping feature of ANN models. Input–output mapping feature is described by Haykin (1999) as learning from available data by input–output mapping without making prior assumptions on the model or inputs.
7. Since DAN2 is a new architecture and studied by its developers some parts of the architecture are not clear enough. For example updating method of α_i values is never mentioned by Ghiassi & Saidane (2005). Another example is calculating μ_k, which is a constant multiplier of CURNOLE node for iteration k. In this study the suggested bisection method (Ghiassi & Saidane, 2005) gave very close results to starting μ_k with converging to starting value.

The problems mention about DAN2 structure clearly shows DAN2 architecture behaves like a statistical method rather than an artificial neural network.

The overall results show that classical ANN model MLP gives the most reliable best results in forecasting time series. Hybrid methods failed to improve the forecast results.

When MLP results are observed NASDAQ index seems to show inconsistencies for a considerable time before it can stabilize. Only

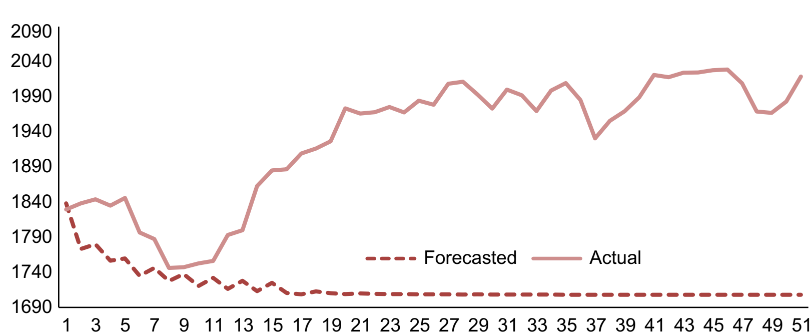

Fig. 3. NASDAQ index forecasted period.

Table 2
Results of ANN and hybrid models.

Method	Training			Test		
	MSE	MAD	MAD %	MSE	MAD	MAD%
MLP	2227.416	36.909	2.324	2478.1468	41.153	2.516
GARCH-MLP	2695.324	38.446	2.465	3665.8387	42.739	2.775
DAN2	2349.259	37.290	2.409	1472.278	32.875	2.768
GARCH-DAN2	19383.400	119.081	7.361	20901.198	109.626	6.487

Table 3
Coefficients of DAN2 regression model[a].

	Model	Unstandardized coefficients		Standardized coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1460.510	196.043		7.450	.000
	LAG1	.277	.244	.224	1.138	.257
	LAG2	.017	.310	.014	.056	.955
	LAG3	.052	.308	.044	.168	.867
	LAG4	−.243	.247	−.219	−.985	.326
	$\sigma^2\ (t-1)$	−119517.701	190445.589	−.071	−.628	.531
	$\varepsilon^2(\mathrm{t}-1)$	−12831.899	37110.957	−.030	−.346	.730

[a] Dependent Variable: NASDAQ.

17 days can be predicted based on cycles completed in time series history. It can also be said that longer periods cannot be forecasted by simply technical analysis (by only using previous index values or variables derived from previous index variables) but those analysis are to be completed by using the fundamental analysis (evaluating index with global and national economic analysis, market analysis, ratio analysis etc.) or case based scenario analysis. It should be noted that when the forecasted and realized Index data evaluated, the MLP model clearly showed that first movement of NASDAQ index is down and this forecast realized about 9th day. But as mentioned before we do not have a crystal ball so forecasting the following movements are very difficult.

6. Conclusion & future research

This study is in search for reducing the shortcomings of using ANN in predicting the market values. With this aim this study motivated from a new ANN model; DAN2 developed by Ghiassi & Saidane (2005) and the hybrid models (GARCH-ANN, EGARCH-ANN) developed by Roh (2007). In order to present the differences in accuracy of prediction, all the models are applied on the same set of data retrieved from NASDAQ Stock exchange.

The results show that classical ANN model MLP outperforms DAN2 and GARCH-MLP with a little difference. GARCH inputs had a noise effect on DAN2 because of the inconsistencies explained in the previous section and GARCH-DAN2 clearly had the worst results. Thus further researches' should focus on improving DAN2 architecture. At least for now, simple MLP seems to be the best and practical ANN architecture.

When the MLP model used to forecast the future movements of the NASDAQ index, MLP model correctly forecasted the first movement as down. The realized value (1747.17) had very small difference (0.54%) with the forecasted value (1737.70). Thus MLP is a powerful and practical tool for forecasting stock movements.

Since hybrid models (GARCH-ANN) do not give satisfying results, despite Roh's (2007) research, a lot of time series should be used to understand the inner dynamics of hybrid models, before making a conclusion about hybrid models performance. Roh reported that 20–25 % of the learning of each ANN came from GARCH or E-GARCH input variables, which are inputs of technical analysis, but 75–80 % in that research many other correlated variables, which are inputs of fundamental analysis, such as bond yields, bond prices, contract volume etc. Further researches should be focus to discover whether GARCH, E-GARCH has a correcting effect on forecasts or other correlated variables has a corrective effect on forecasts. The results of these further researches will lead us to many powerful financial time series forecasting models.

References

Alpaydın, E. (2004). *Introduction to machine learning*. Londan, England: The MIT Press.

Ando, T. (2009). Bayesian portfolio selection using a multifactor model. *International Journal of Forecasting, Special Section: Time Series Monitoring, 25*(3), 550–566.

Atsalakis, G. S., & Valavanis, K. P. (2009b). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications, 36*, 5932–5941.

Atsalakis, G. S., & Valavanis, K. P. (2009a). Forecasting stock market short-term trends using a neuro-fuzzy based methodology. *Expert Systems with Applications, 36*, 10696–10707.

Bildirici, M., & Ersin, Ö. Ö. (2009). Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul stock exchange. *Expert Systems with Applications, 36*, 7355–7362.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroscedasticity. *Journal of Econometrics, 31*, 307–327.

Celik, A. E., & Karatepe, Y. (2007). Evaluating and forecasting banking crises through neural network models: An application for Turkish banking sector. *Expert systems with Applications, 33*, 809–815.

Chang, P.-C., Liu, C.-H., Lin, J.-L., Fan, C.-Y., & Ng, C. S. P. (2009). A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications, 36*, 6889–6898.

Chen, C. W. S., & So, M. K. P. (2006). On a threshold heteroscedastic model. *International Journal of Forecasting, 22*(1), 73–89.

Chen, M.-S., Ying, L.-C., & Pan, M.-C. (2010). Forecasting tourist arrivals by using the adaptive network-based fuzzy inference system. *Expert Systems with Applications, 37*(2), 1185–1191.

Cheng, J., Chen, H., & Lin, Y. (2010). A hybrid forecast marketing timing model based on probabilistic neural network, rough set and C4. *5. Expert Systems with Applications, 37*(3), 1814–1820.

Egrioglu, E., Aladag, C. H., Yolcu, U., Uslu, V. R., & Basaran, M. A. (2009). A new approach based on artificial neural networks for high order multivariate fuzzy time series. *Expert Systems with Applications, 36*, 10589–10594.

Engle, R. F. (1982). conditional heteroscedasticity with estimator of variance of United Kingdom inflation. *Econometrica, 50*(4), 987–1008.

Ghiassi, M., & Saidane, H. (2005). A dynamic architecture for artificial neural networks. *Neurocomputing, 63*, 397–413.

Ghiassi, M., Saidane, H., & Zimbra, D. K. (2005). A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting, 21*, 341–362.

Ghiassi, M., Zimbra, D. K., & Saidane, H. (2006). Medium term system load forecasting with a dynamic artificial neural network model. *Electric Power Systems Research, 76*, 302–316.

Gooijer, J. G. D., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting, 22*, 443–473.
Guresen, E., & Kayakutlu, G. (2008). Forecasting stock exchange movements using artificial neural network models and hybrid models. In Zhongzhi Shi, E. Mercier-Laurent, & D. Leake (Eds.), *Proceedings of 5thIFIP international conference on intelligent information processing. Intelligent Information Processing IV* (Vol. 288, pp. 129–137). Boston: Springer.
Hamzacebi, C., Akay, D., & Kutay, F. (2009). Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert Systems with Applications, 36*, 3839–3844.
Hassan, M. R., Nath, B., & Kirley, M. (2007). A fusion model of HMM, ANN and GA for stock market forecasting. *Expert Systems with Applications, 33*, 171–180.
Haykin, S. (1999). *Neural networks: A comprehensive foundation*. New Jersey, USA: Prentice Hall.
Khashei, M., & Bijari, M. (2010). An artificial neural network (p,d,q) model for timeseries forecasting. *Expert Systems with Applications, 37*(1), 479–489.
Kumar, P. R., & Ravi, V. (2007). prediction in banks and firms via statistical and intelligent techniques – A rewiev. *European Journal of Operational Research, 180*, 1–28.
Lee, S., Lee, D., & Oh, H. (2005). Technological forecasting at the Korean stock market: A dynamic competition analysis using Lotka–Volterra model. *Technological Forecasting and Social Change, 72*(8), 1044–1057.
Leu, Y., Lee, C., & Jou, Y. (2009). A distance-based fuzzy time series model for exchange rates forecasting. *Expert Systems with Applications, 36*, 8107–8114.
Liao, Z., & Wang, J. (2010). Forecasting model of global stock index by stochastic time effective neural network. *Expert Systems with Applications, 37*(1), 834–841.
Majhi, R., Panda, G., & Sahoo, G. (2009). Efficient prediction of exchange rates with low complexity artificial neural network models. *Expert Systems with Applications, 36*, 181–189.
Maltritz, D., & Eichler, S. (2010). Currency crisis prediction using ADR market data: An options-based approach. *International Journal of Forecasting, 26*(4), 858–884.
Mcnelis, P. D. (2005). *Neural networks in finance: Gaining predictive edge in the market*. USA: Elseivier Academic Press.
Menezes, L. M., & Nikolaev, N. Y. (2006). Forecasting with genetically programmed polynomial neural networks. *International Journal of Forecasting, 22*, 249–265.
Nelson, D. B. (1991). Conditional heterosdasticity in asset returns: A new approach. *Econometrica, 59*(2), 347–370.
Pekkaya, M., & Hamzaçebi, C. (2007). Yapay sinir ağları ile döviz kuru tahmini üzerine bir uygulama. In *Proceedings of the 27th YA/EM National Congress* (pp. 973–978). Izmir, Turkey.
Preminger, A., & Franck, R. (2007). Forecasting exchange rates: A robust regression approach. *International Journal of Forecasting, 23*, 71–84.
Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (1999). *Neural and adaptive systems: Fundamentals through simulations*. New York, USA: John Wiley & Sons.
Ramnath, S., Rock, S., & Shane, P. (2008). The financial analyst forecasting literature: A taxonomy with suggestions for further research. *International Journal of Forecasting, 24*(1), 34–75.
Roh, T. H. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications, 33*, 916–922.
Scharth, M., & Medeiros, M. C. (2009). Asymmetric effects and long memory in the volatility of Dow Jones stocks, international journal of forecasting. *Forecasting Returns and Risk in Financial Markets using Linear and Nonlinear Models, 25*(2), 304–327.
Wang, Y., Keswani, A., & Taylor, S. J. (2006). The relationships between sentiment, returns and volatility. *International Journal of Forecasting, 22*(1), 109–123.
Yu, T. H., & Huarng, K. (2008). A bivariate fuzzy time series model to forecast the TAIEX. *Expert Systems with Applications, 34*, 2945–2952.
Yudong, Z., & Lenan, W. (2009). Stock market prediction of *S*& *P* 500 via combination of improved BCO approach and BP neural network. *Expert Systems with Applications, 36*, 8849–8854.
Zhang, Y., & Wan, X. (2007). Statistical fuzzy interval neural networks for currency exchange rate time series prediction. *Applied Soft Computing, 7*, 1149–1156.
Zhu, X., Wang, H., Xu, L., & Li, H. (2008). Predicting stock index increments by neural networks: The role of trading volume under different horizons. *Expert Systems with Applications, 34*, 3043–3054.