A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price

Reza Hafezi [a,c], Jamal Shahrabi [b,*], Esmaeil Hadavandi [b]

[a] Technology Foresight Group, Department of Management, Science and Technology, Amirkabir University of Technology, P.O. Box 15875-4413, Tehran, Iran,
[b] Department of Industrial Engineering, Amirkabir University of Technology, P.O. Box 15875-4413, Tehran, Iran
[c] Member of Futures Studies Research Institute

ABSTRACT

Creating an intelligent system that can accurately predict stock price in a robust way has always been a subject of great interest for many investors and financial analysts. Predicting future trends of financial markets is more remarkable these days especially after the recent global financial crisis. So traders who access to a powerful engine for extracting helpful information throw raw data can meet the success. In this paper we propose a new intelligent model in a multi-agent framework called bat-neural network multi-agent system (BNNMAS) to predict stock price. The model performs in a four layer multi-agent framework to predict eight years of DAX stock price in quarterly periods. The capability of BNNMAS is evaluated by applying both on fundamental and technical DAX stock price data and comparing the outcomes with the results of other methods such as genetic algorithm neural network (GANN) and some standard models like generalized regression neural network (GRNN), etc. The model tested for predicting DAX stock price a period of time that global financial crisis was faced to economics. The results show that BNNMAS significantly performs accurate and reliable, so it can be considered as a suitable tool for predicting stock price specially in a long term periods.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Creating an intelligent system that can accurately predict stock price in a robust way has always been a subject of great interest for many investors and financial analysts. However, the stock market domain is dynamic and unpredictable [1]. It is not a secret that stock market movements are reactive to external and internal factors as political, economic and even social [2,3] so stock market prediction can be classified under complex systems.

Complex system is an approach to study how the subsystems of a system interact with each other and how the whole system interacts and manages relationships with its environment. In modeling such complex systems (like ours), difficulties have already faced during designing a robust model and selecting its architecture [4]. To dealing with this problem we chose to design our complex system under multi-agent framework described below.

A multi-agent based system is a combination of autonomous decision makers named agents that communicate with each other under prescribed rules. An agent is a software program which is capable of autonomous action within its environment in order to meet its design objectives [5] and accordingly a multi-agent system is a set of interacted agents suited to solve a problem in a distributed manner. Multi-agent systems have autonomy integration, reactivity and flexibility capabilities [4]. A multi-agent based paradigm is therefore considered to be well suited to explaining and understanding the phenomena associated with the complex phenomena of financial system [5]. Stock market prediction is a distributed problem means that it is a combination of independent subtasks performs specific functions to achieve the global goal which is predicting price accurately.

The most important part of a multi-agent system is the artificial intelligent section. In our research we chose data mining, function approximation and knowledge discovery to handle this role.

A wide range of computer based evolutionary algorithms exist that can be applied for this objectives such as classifier systems, feature selection algorithms (FS), genetic algorithm (GA), genetic programming (GP), artificial neural network (ANN), fuzzy inference systems, etc. In our proposed model we used feature selection and time lag selection for data preprocessing phase and a hybrid bat-neural network (BNN) model for the function approximation phase. Khan and Sahai showed that using bat algorithm (BA) for optimizing neural network weights performs better than some well-known

models such as GA, PSO (particle swarm optimization) and BP (back propagation) and LM (Levenberg–Marquardt algorithm) [6].

For data oriented problems such as stock market prediction, data quality is a key factor. In the literature there are two important indicators for prediction stock price including fundamental analysis and technical analysis. Fundamental analysis uses the information in company financial statements, national financial position and even some international factors such as oil price, gold price, etc. Technical analysts believe that stock price changes generally are driving from prices and volume data [7,8]. Technical analysis gives a framework for studying investors behaviors, however fundamental analysis provides a mechanism to evaluate company's financial health. In addition fundamental analysis can be extended to other factors like geopolitical factors of influence, other economic data that is release by the government and news [2]. To utilizing the benefit of both kind of analysis we used fundamental and technical data sets simultaneously.

Another important factor for this kind of financial problems is time lag selection that known as Takens theorem, states that for a wide class of deterministic systems there exist a one to one mapping between past $y(t-1)$, $y(t-2)$,..., $y(t-d)$ of time series and the state of the system at time $y(t)$ [9].

However in this research we emphasized on promoting the stock price prediction accuracy using the hybrid bat neural network (BNN) intelligence system combining the variables of technical and fundamental analysis considering the data time lags.

The rest of the paper is organized as follows. Section 2 presents a review of related works. Section 3 describes architecture of proposed model. Section 4 demonstrates the experimental results. This paper concluded in Section 5.

2. Related Works

In recent years many researchers showed their interest in the field of combining multi-agent based systems and data mining tasks and algorithm. Ali Albashiri and et al. [10] proposed an extendible multi-agent data miner (EMADS) which is to address data mining requests by interacted agents under decentralized control. Czarnowski and Jedrzejowicz [11] described an agent based distributed learning framework based on data reduction. Yang et al. [12] proposed a multi-agent based data mining system to biological data analysis. Wardeh et al. [13] proposed an approach to multi-agent classification in which individual agents argue for a given example to be classified with a particular label according to their local data, implemented in a PISA (pooling information from several agents) multi-agent framework. Ghedini Ralha and Sarmento Silva [14] described a multi-agent data mining system to extracting information from the Brazilian federal procurement process databases to recognize cartel formation among applicants in the process of corruption detection and prevention. Giannella et al. [15] offered a perspective on distributed data mining algorithms in context of multi agent systems. Klusch et al. [16] focused on the potential added value of using agent technology in the domain of knowledge discovery. Gao et al. [17] presented a model for cooperative agents for mining knowledge from heterogeneous data (CoLe). Shukla et al. [18] conceptualized a bidding-based multi agent system for solving integrated process-planning and scheduling problems. They proposed an architecture consist of autonomous agents capable communicate with each other which can make decisions based on their knowledge. da Silva et al. [19] proposed a view on distributed data mining in multi-agent framework. They discussed the connection between distributed data mining and multi-agent systems. Zghal et al. [20] presented an agent framework for data mining. They used multi-agent systems to improve the execution time at different levels.

Some researches were carried out in using intelligent agent based model in forecasting. Cendon et al. [21] implemented a multi-agent system to control a complex production environment. In their research agents used data mining techniques to simulate actual status of the manufacturing process to support decision making procedure. Lee and Liu [22] proposed an intelligent agent based system for weather forecasting using fuzzy-neuro network to achieve automatic weather information gathering and filtering and for time series weather prediction. Jumadinova and Dasgupta [23] developed a multi-agent based system that incorporates different information related aspects and analyzing the effect of them in prediction market. Chang and Liu [24] developed a Takagi-Sugeno-Kang (TSK) type fuzzy rule based system for stock price prediction. Fazel Zarandi et al. [25] presented a type-2 fuzzy rule based expert systems for stock price analysis. The proposed model applies on both technical and fundamental indexes. Lu [26] proposed an integrated independent component analysis (ICA)-based denoising scheme with artificial neural network for prediction. Boyacioglu and Avci [27] investigated the predictability of stock market return with adaptive network-based fuzzy inference system (ANFIS). The authors used six macro-economic variables and three indices as input variables to model and predict the return on stock price index on Istanbul stock exchange [27]. Kara et al. [28] proposed two models based on two classification techniques, artificial neural network (ANN) and support vector machines (SVM). They showed that the average performance of ANN model was found significantly better than that of SVM model (Ten technical indicators have been used as input variables.) [28]. Feng and Chou [29] presented an artificial neural prediction system developed with combinations of step wise regression analysis (SRI), dynamic learning and recursive-based particle swarm optimization (RPSO) learning algorithms (twenty technical indexes have been investigated as input variables.). Hsu [30] hybridized a self-organized map (SOM) neural network and genetic programming (GP) in order to predict stock price. Chang [31] focused on three different algorithms, artificial neural network, decision trees and the hybrid model of ANN and decision trees, for predicting stock price. The study discovered that compared to other two methods, ANN is more suitable method for predicting stock price in the volatile post-crisis stock market [31]. Chang et al. [32] proposed a novel model by evolving partially connected neural networks (EPCNNs) to predict the stock price trend using technical indicators as inputs.

In financial and economic area Kettera et al. [33] presented that an autonomous agent can use observable market conditions to characterize the micro economic situation of the market and predict market trends. Raudys and Zliobaite [34] proposed a multi-agent system for prediction of financial time series aimed at help to reduce forecasting error using a system of several adaptive forecasting agents. Wei et al. [35] proposed a simple agent-based model of trading incorporating momentum investors. The model is able to reproduce some of the stylized facts observed in real markets [35]. Zhang et al. [5] designed a multi-agent based system which is integrated using an agent-oriented approach and ontology as a common understanding of problem domain, by focusing on valuation effects of bankruptcy filing through inter-firm linkage. Cui et al. [36] described a novel architecture to model the stock market by utilizing stock agent, finance agent and investor agent. Li et al. [37] designed an agent-based continuous double auction stock market, which uses the same trading mechanism as the Chinese stock market. Kluger and McBride [38] developed an autonomous agent-based market micro structure simulation with both informed agents and uninformed agents liquidity-motivated agents. Ponta et al. [39] presented an artificial stock market characterized by heterogeneous and informed agents. Yang et al. [40] constructed an agent-based stock market model which concisely described investors' heterogeneity and adaptability by introducing

price sensitivity and feedback time. This article presents an intelligent financial time series prediction model based on the agent framework to meet the fluctuations made by the financial crisis.

Next part is aimed at describing the architecture and designing of the multi-agent framework. We will explain the goals of the system in each layer and then roles will describe and clarify how the clusters of roles shape agents. Also a detailed explanation about the predictive model will be present.

3. Architecture of the proposed model

The architecture of proposed BNNMAS consists of four layers. The first layer stands for gathering data using external data sources such as daily statistics and even qualitative data like news and also expert knowledge to determining data categories. The results of this layer send to the second layer aimed at data preprocessing and index voting. In this layer we try to choose the most important and robust features among normalized data to reduce the complexity but maintaining validation in the same stage using two parallel processes, lag selection to choose the best time lag of features to improve the accuracy and cross correlation to check one by one correlation (each feature with the target). The results of these two processes send to voting node to choose the most relevant features and applying time lags on them. Then the reformed dataset send to the third layer via selected features protocol. The third layer is aimed at designing an artificial intelligence framework to predict stock price (we used artificial neural network in our case). In this layer 80% of normalized data used to training ANN with BAT algorithm and the results delivered to the forth layer aimed at validation testing and reporting. This layer provides model analysis and knowledge representation, generating prediction of stock price and investigation of the probable scenarios to make the best decision by interacting with the expert (Based on the multi-disciplinary trait of this macro-economical decision it seems to be good if the experts group contain economists, futurist, financial analysers, traders, etc.). The proposed architecture is shown in Fig. 1.

3.1. Goals of the system

The starting point of designing a system is specification of system goals. This procedure often resulting a list or diagram of goals and sub goals. The reasons of building a system should always be central in one's thinking when specifying the system. In addition goals are central to the functioning of the intelligent software agents that are going to realize the system [41]. Our final goal is "forecasting the stock price with the highest accuracy". To achieve this goal we designed a four layer architecture, each one has its own goals and sub goals shown in Fig. 2 (we used the Prometheus designing tools (PDT) version 2.5 to drawing the diagrams). Fig. 2 represents goal overview diagram, directed a cyclic graph of all goals in the system.

As has been shown in Fig. 2, each layer has its own goals and sub-goals coordinated with other layers to facilitate enhancing the overall goal of the whole system (in our case, predicting the stock price accurately). Layer one is aimed at gathering data from even qualitative and quantitative datasets. So data gathering and feature detection agent get quantitative data from daily stock price datasets and get quantitative data from news. To make these two heterogeneous datasets this agent must transforming quantitative data. To achieve this sub-goal quantitative data classified using expert's opinions and then based on the influence rate of each classes the two data sets can be comparable.

Layer two has a main goal to detect most relevant features to decrease the complexity level of computations while considering the whole space of the problem. Three sub goals determined at this

level: lag selection (to define the time lag of each parameter on the goal parameter), feature selection (to select the most relevant features) and data normalization (to prevent the effects of parameters range on the results). All these parts will be described in further section of the article.

Layer three aimed at predicting the stock price. It starts with creating an artificial neural network (ANN) and initializing its architecture. To optimize the ANN's weights this agents create a BAT algorithm to adjust the weights of the ANN. Finally layer four attempt to build scenarios based on prediction results, experts ideas and environmental scanning.

3.2. Roles of the system

After defining goals and sub-goals of the model, it is the time to describe the roles by aggregating goals. At a glance roles define functionality of agents created by grouping of related goals, percepts, and actions for a series of behaviors. Determining the system goals is the key step to defining the agent types. Fig. 3 shows the system roles diagram, which grouped goals, percepts and actions under roles. Roles are generated based on the goals determined before. The star shape stands for introducing inputs while rectangles consist of roles and ovals consist of goals. The outputs are accessible in rectangle plus triangle shape.

As we determined roles of system we can specify agents by grouping roles under agents. The roles that seem related make sense to group together. Fig. 4 shows the agent-role grouping diagram where you can see the agents used in our research.

Fig. 5 describes how different types of agents interact in the system. Each agent is described in the bottom of the figure.

3.3. Description of the agents

As we mentioned in Fig. 3 each layer in our proposed architecture contains number of agents that execute defined tasks to achieve system overall goals. Detailed description of each agent is expressed in the following sections.

3.3.1. Data gathering and feature detection

This agent aimed at creating the most relevant metadata using quantitative and qualitative data sets. But it faces an important problem in its way, doing the job that is these kinds of data are not comparable in most situations and even they are heterogeneous. Of course it is a routine to gathering data in such models but our agent tries to classify qualitative data using categories defined by expert(s) and compute the degree of their influence to create a homogenous dataset.

3.3.2. Data preprocessing agent team

An agent team consists of agents coordinate to execute tasks. Agent teams synchronize the work of the system, execute plans in concurrent mode and strengthen the internal management by local decision making [42].

Data preprocess agent team consists of "data normalization agent", "lag selection agent", "cross correlation feature selection" and "feature selection decision making".

3.3.2.1. Data normalization agent.
The data normalization agent utilizes a normalization method to adjust values measured on different financial indexes with various domains and scales. Simply data normalization defines as adjusting values measured on different scales to a notionally common scale. Our data set consist of different financial indexes with different domains and scales and data normalization helps us to access to a dataset which contains homogenous data.
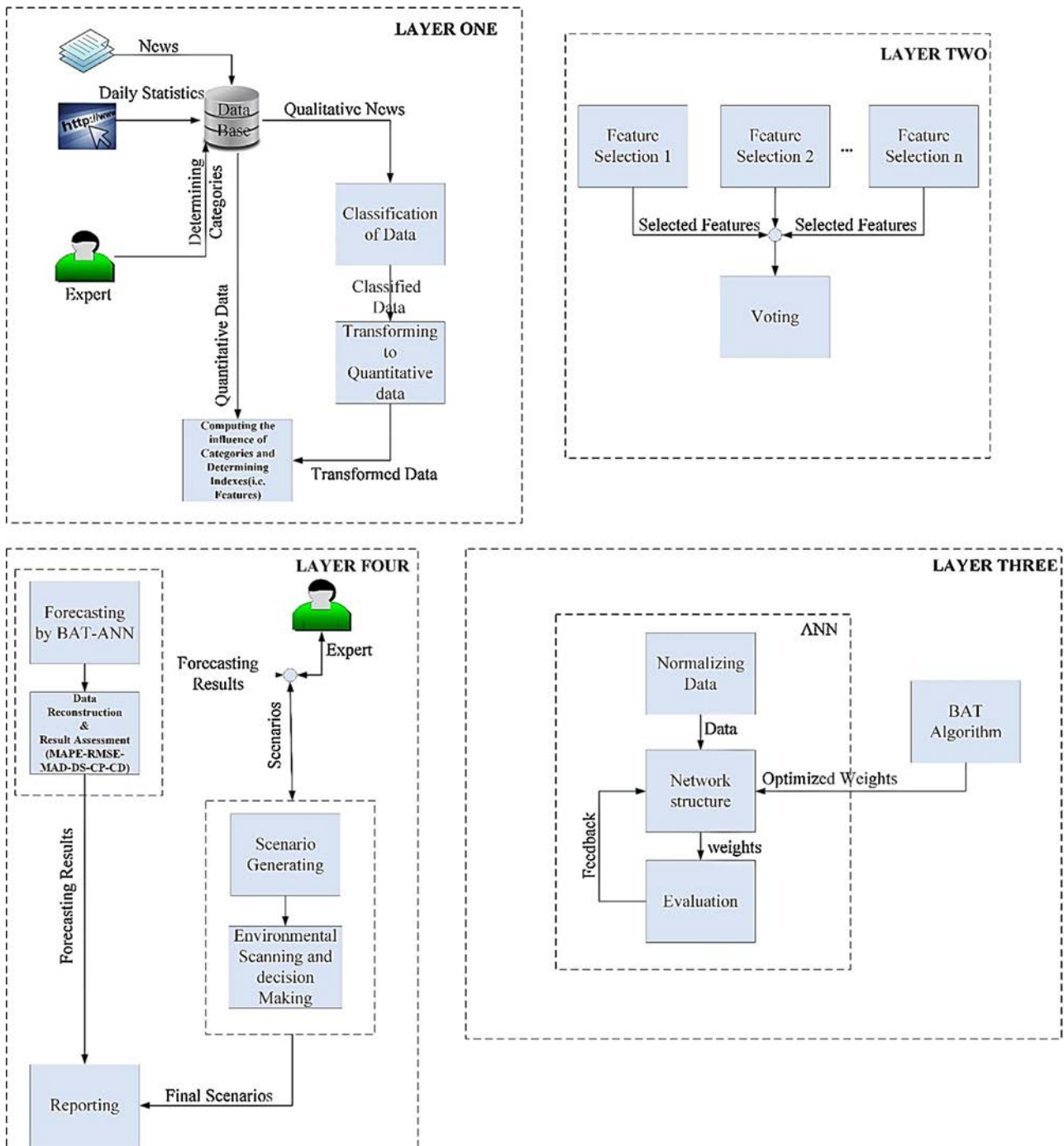
Fig. 1. Architecture of proposed BNNMAS.

In this research we used "min–max" normalization method to reform data set by following equation:

$$\text{Normalized data} = \frac{(x(i) - \text{Min}\{x\})}{(\text{Max}\{x\} - \text{Min}\{x\})} \tag{1}$$

where $x(i)$ is (i)th element in the column and $\text{Min}\{x\}$ minimum and $\text{Max}\{x\}$ is the maximum of related column's elements.

3.3.2.2. Lag selection agent. Sometimes financial features show their effects with lags of time. Imagine a new tax policy applied by the government. In macroeconomics no economist would say that the policy will work after the notification of implication. There

would be lags of time for a decision in this kind of environment to affect the market, etc. Detecting these lags help the model to accurately follow the fluctuations. We have several features may each one has different scale time lag. At this stage we attempt to determine these lags and apply on the dataset.

Several approaches have been proposed for lag selection. These approaches consider lag selection as a pre- or post-processing or as a part of the learning process [43]. For example statistical tests based on information criteria (IC) [44] such as Akaike information criteria (AIC), Bayesian information criteria (BIC) and Schwarz Bayesian information criteria (SBIC) are among the popular pre-processing methods to select an appropriate number of lags in time
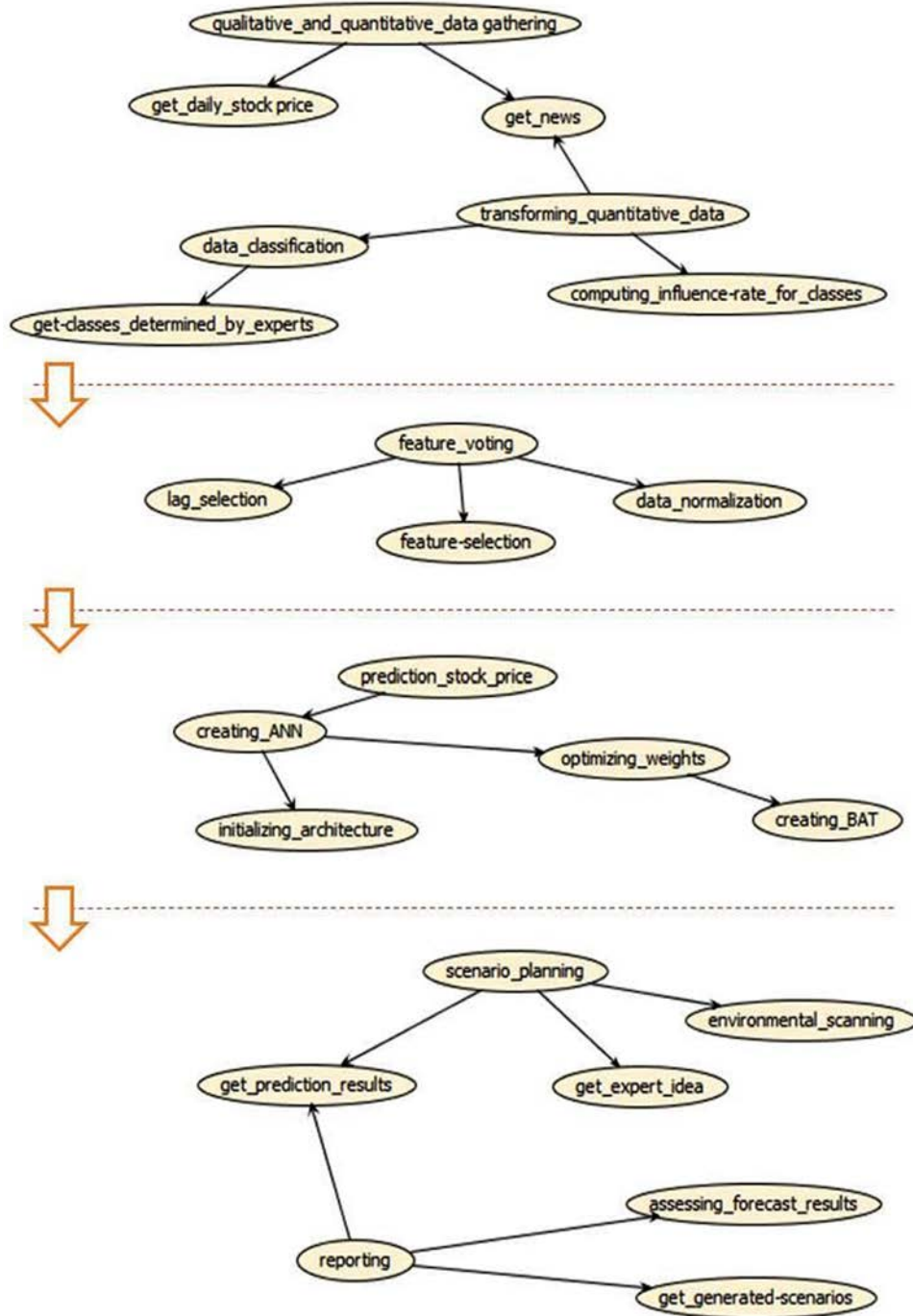
Fig. 2. Goals assigned to each (of four) layer.

series forecasting [45]. These methods consider models with a minimum number of 1 lag up to a maximum of p (including all the intermediate lags). The idea is to choose the lag order p to minimize a function of the form presented by following equation:

$$IC(p) = N \ln \hat{\sigma}^2(p) + p[f(N)] \tag{2}$$

where $\hat{\sigma}^2(p)$ is the estimated regression variance, which depends upon the sample size and order p of lag structure of the model, and N is the number of usable observations [46]. $p[f(N)]$ can be interpreted as the penalty function for increasing the order of the model. Different choices of $f(N)$ give different information criteria. The AIC results from setting $f(N) = 2$, and SBIC results from assuming $f(N) = \ln(N)$

and finally the HQIC is equal to $f(N) = \ln(\ln(N))$. Ideally AIC, SBIC and HQIC will be as small as possible. Therefore, the model to be chosen should be the one with the lowest value of the information criteria test.

As we discussed above to achieve better results we implemented lag selection on our normalized data by using time series tool provided by MATLAB software.

Samples of executed lag selection are shown in Fig. 6. We used −10 to +10 time lags for each attribute versus the target. For each lag the horizontal line shows the level of correlation among related attribute and target. So the time lag which was in higher correlation level is the effective one.
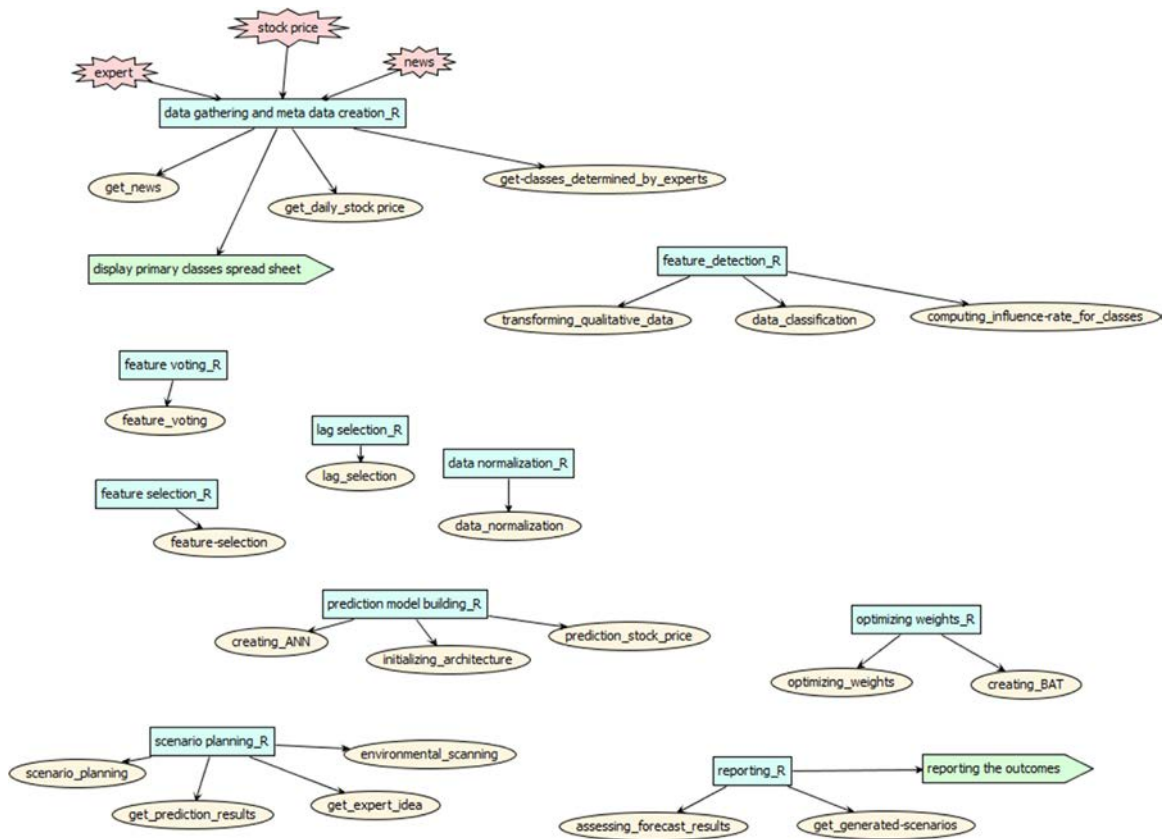
Fig. 3. System roles diagram.

3.3.2.3. Cross correlation feature selection. Feature selection can be simply defined as choosing relevant features and denying irrelevant ones. For models including learning algorithm like as ours feature selection help to improve outcomes and also reducing computational costs like running time.

In this paper we implemented a method to select the most relevant attributes based on correlation statistic. We did it in parallel with the lag selection task throw previous step.

Cross correlation is a well-known method of computing the degree of relationship exists between two series $x(i)$ and $y(i)$ where
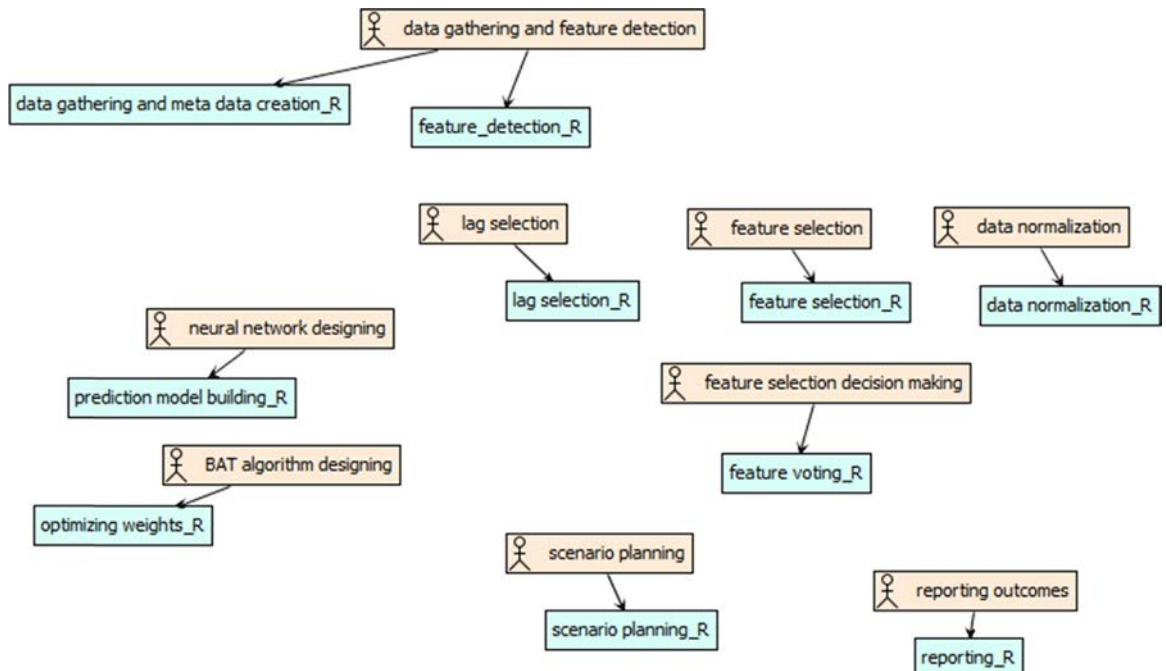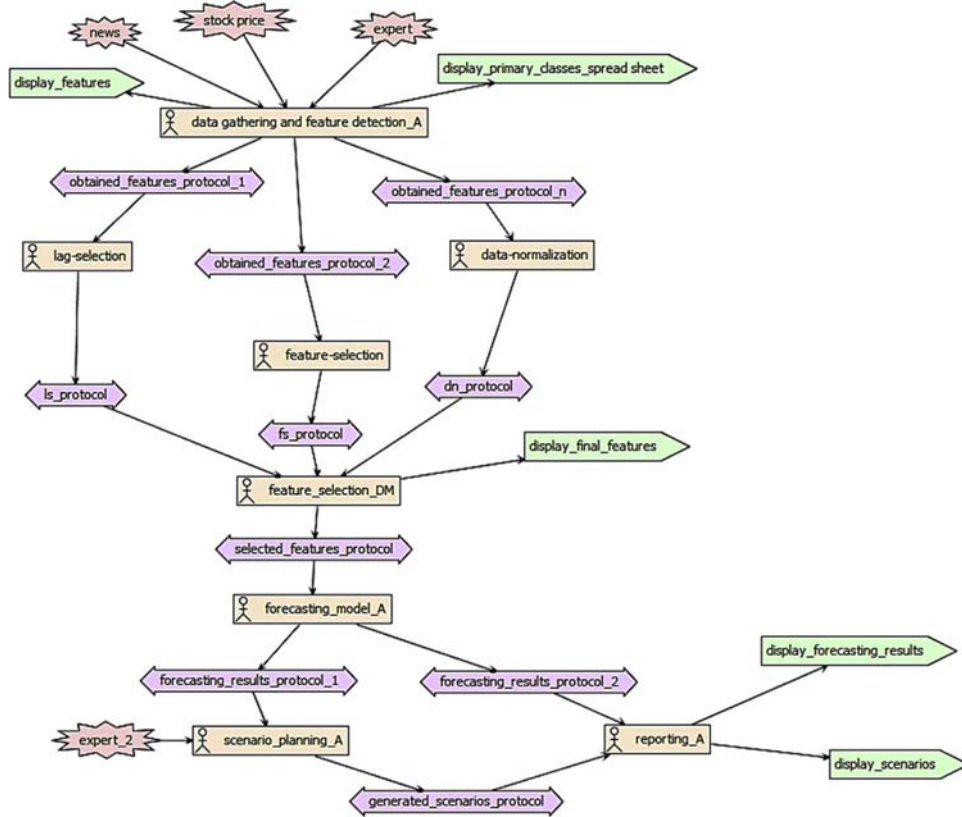

Fig. 4. Agents of system.

Fig. 5. System overview diagram.

$i = 0, 1, \ldots, N-1$. The cross correlation r at delay d is defined as:

$$r = \frac{\sum_i[(x(i) - mx) \times (y(i-d) - my)]}{\sqrt{\sum_i(x(i) - mx)^2} \times \sqrt{\sum_i(y(i-d) - my)^2}} \quad (3)$$

where mx and my are the means of the corresponding series.

The above equation returns a value in -1 to $+1$ range where -1 shows a powerful inverse relation and $+1$ shows a powerful direct relation and 0 presents that there is no relation between two compared variables. So we checked the cross correlation in the best time lag, selected previously.

3.3.2.4. Feature decider agent. This agent aims at preparing the data for modeling phase by delivering the outcomes of other three agents join in the data preprocess agent team. It chooses the most relevant and meaningful attributes to improve the prediction accuracy.

3.3.3. Modeling agent team

This agent team aimed at creating an artificial neural network trained by a metaheuristic algorithm called BATS so this team consist of two agents: "neural network designing" and "bat algorithm designing". These agents are described in the following sections.

3.3.3.1. Neural network designing agent. A biological neural network is composed of number of process units called neurons. A single neuron can be connected to many neurons to create a network aimed at a predefined goal. Artificial neural network is a form of computing inspired by biological models for studying, learning and intelligence.

Architecture of an artificial neural network consists of parameters some of them listed below:

1. The input layer.
2. The hidden layer(s).
3. The interconnection pattern between different layers.
4. The learning process for updating the weights of the interconnections.
5. The transformation function that converts a weighted input to output.
6. The number of each layer neurons.
7. The output layer.

A simple form of an artificial neural network structure is showed in Fig. 7.

The neurons of an ANN are stood in layers (fig. 7). To interact and send information the nodes of one layer are connected to the nodes of the next layer. The connection between node i and j of previous and following layers are associated with the weight W_{ij} and also each neuron has a bias b_i associated with it [47]. A neural network uses an error function and also an algorithm to minimize the computed error at each stage (epoch).

An ANN has the capability to be trained for adjusting matrix of weights W_{ij} to structure a more accurate network and minimizing the performance function.

We aimed at delivering an accurate stock price prediction as an output layers outcome, so we input the data provided by the data preprocessing agent team at the first layer (input layer). We chose to have one hidden layer to perform the prediction so we have a three layer neural network. Also we used six neurons for the hidden layer because it returned the best results in practical testing instead of other number of neurons. Results for three, four, five, six
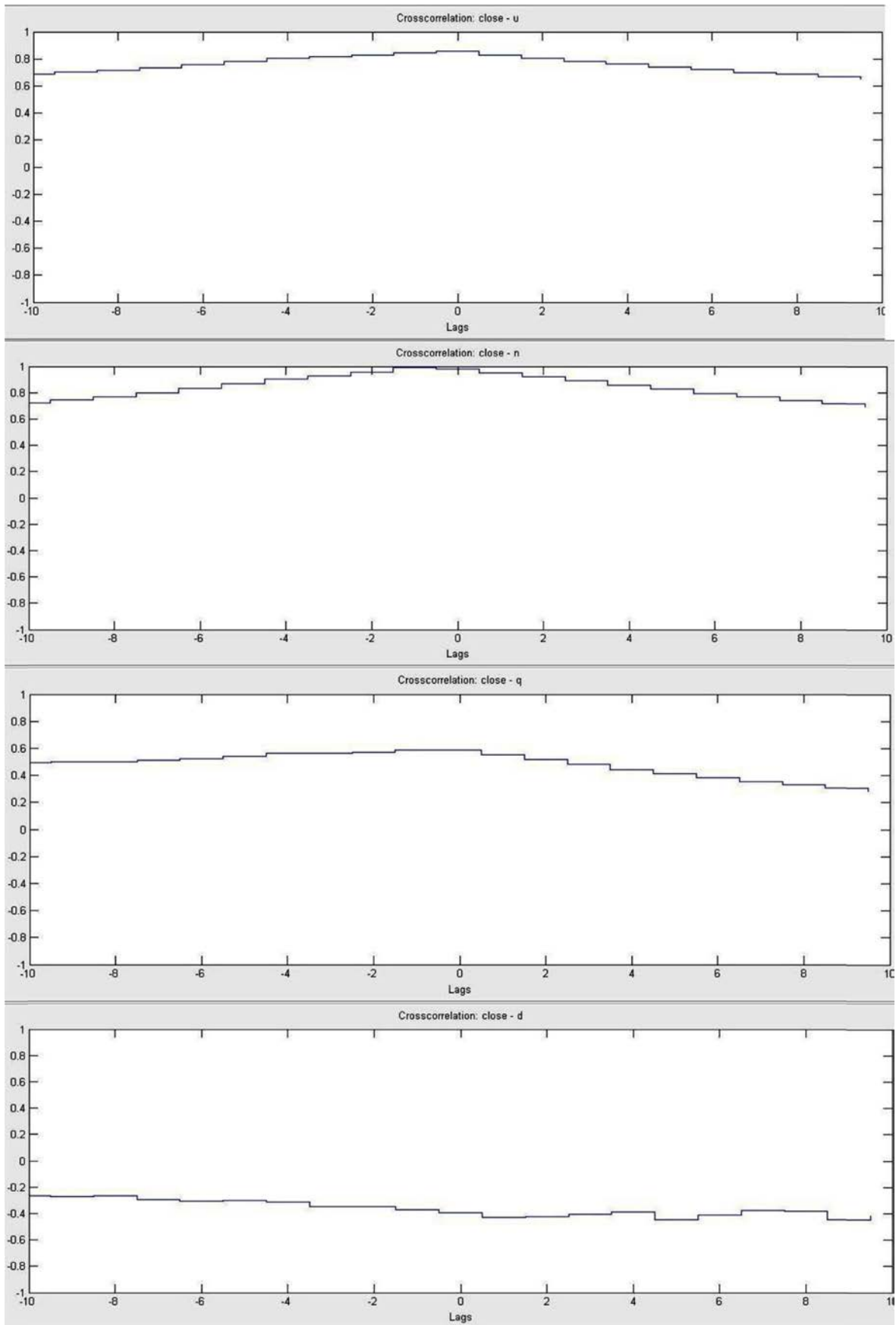
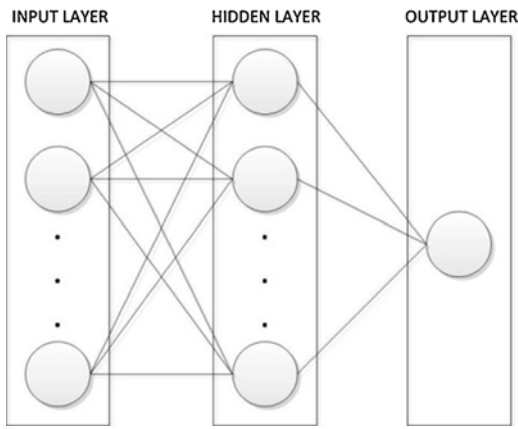**Fig. 6.** Samples of lag selection diagrams.

Fig. 7. A simple form of artificial neural network (ANN).

and seven neurons are showed in Fig. 8. Based on these results we decided to choose a six neuron hidden layer.

We used radial basis function (RBF) as transfer function for the hidden layer. Park and Sandberg [48] showed that radial a basis network with a hidden layer has the ability to do the universal approximation. In RBF each unit in the hidden layer of the network has its own centroid and the distance between the centroid and x is computed for each input $X = (x_1, x_2, \ldots, x_n)$. The output signal at one of the kernel nodes is some nonlinear function of that distance. Thus, each kernel node in the RBF network computes an output that depends on a radially symmetric function, and usually the strongest output is obtained when the input is near the centroid of the node [48].

Training a neural network is a complex task that has a direct effect on the output results. Refers to Atsalakis and Valavanis [49], many successful researches are done that used metaheuristic and intelligent algorithms such as genetic algorithms. Yang [50] proposed a new metaheuristic bat-inspired algorithm and showed that it works significantly better such powerful algorithms like particle swarm optimization (PSO) and genetic algorithm (GA) in their standard version.

We proposed an artificial neural network that uses bat algorithm as a learning algorithm. Flowchart of proposed bat-neural network (BNN) is showed below.

Principals of the bat algorithm are described in the following section.

3.3.3.2. Design of the BAT algorithm. Bat algorithm (BA) is a metaheuristic method based on the echolocation behavior of bats, proposed by Yang. Yang [50] showed that BA is potentially more powerful than PSO and GA as well as harmony search. Bat algorithm tries to combine the advantages of existing algorithms and also covers their disadvantages. Fig. 9 shows the flowchart of BA.

Bats are the only mammals with wings. Bats use a type of sonar, called, echolocation, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. A very loud sound emitted by bats and
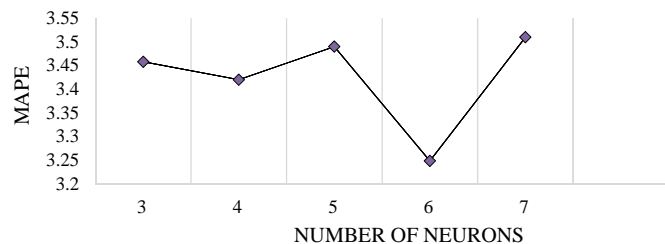

Fig. 8. Performance of BNN for different number of neurons.

surrounding objects detected by listening to the echo that bounces back [50]. The procedure of applying BAT algorithm conceptualized in Fig. 10 based on the echolocation capability of bats.

As showed in Fig. 9 at the start point we have to initialize the position x_i and velocity y_i in a d-dimension search space. The initial pulse frequency f_i calculated by the following equation:

$$f = f_{\min} + (f_{\max} - f_{\min})\beta \tag{4}$$

where $\beta \in [0, 1]$ is a random vector drawn from uniform distribution and $f_{\min}$, $f_{\max}$ are predefined values that related to the problem space (For our problem: $f_{\min} = 0$ and $f_{\max} = 1$).

The velocity v_i update at each step:

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i \tag{5}$$

Here x_* is the current global best solution.

The new solution x_i^t at each time step is given by:

$$x_i^t = x_i^{t-1} + v_i^t \tag{6}$$

The loudness A_i and the pulse rate r_i are initialized randomly in range of predefined minimum and maximum value for each variable.

For the local search part a new solution is generated locally using random walk:

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t \tag{7}$$

where $\varepsilon \in [-1, 1]$ is a random number and A^t is the average of loudness value of all bats at the current time step.

Refers to bats behavior the loudness in the nature A_i usually decreases ones bat reaches its target while pulse rate r_i increases.

$$A_i^{t+1} = \propto A_i \tag{8}$$

$$r_i^{t+1} = r_i^0[1 - \exp(-yt)] \tag{9}$$

where $\alpha \in (0, 1)$ and $y > 0$ are constants the user defines according to the problem. Also when $t \to \infty$ we have:

$$A_i^t \to 0 \quad \text{and} \, r_i^t \to r_i^0 \tag{10}$$

As we used Eq. (9) r_i^t can be any value $r_i^0 \in [0, 1]$.

3.3.4. Knowledge representation agent team

This agent team delivers outcome of the modeling agent team and tries to discover the hidden knowledge by interaction with the expert. This team consists of two agents "scenario planning" and "reporting" agents.

3.3.4.1. Scenario planning agent. This agent is aimed at scenario planning based on the models outcome and experts knowledge. Then it does environmental search and the results return to the expert for planning an alternative scenario or redesigning the existing ones if it is necessary.

3.3.4.2. Reporting agent. This agent is used to publish reports in type of textual, graphical or other type of documents required.

For this study we focused on data preprocessing agent team and modeling agent team and the other parts left for further researches.

4. Experimental results

4.1. Data

The dataset we used in our research gathered from the duetsche bundes bank (updated data can be find in the duetsche bundes bank website: www.bundesbank.de). Our dataset consists of quarterly data of 17 national indexes and 3 international indexes including oil price, gold price and exchange rate of German mark by U.S. dollar.
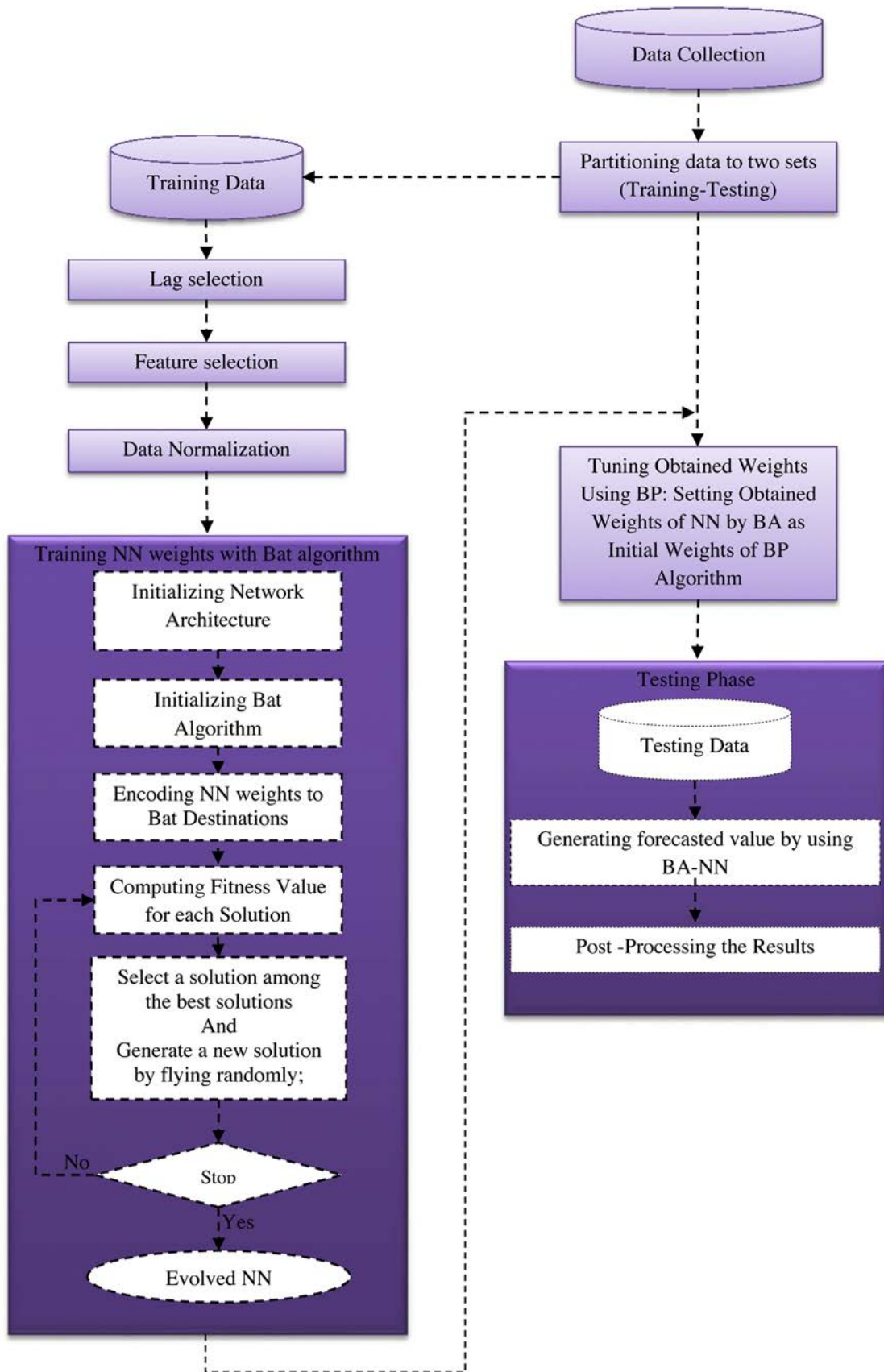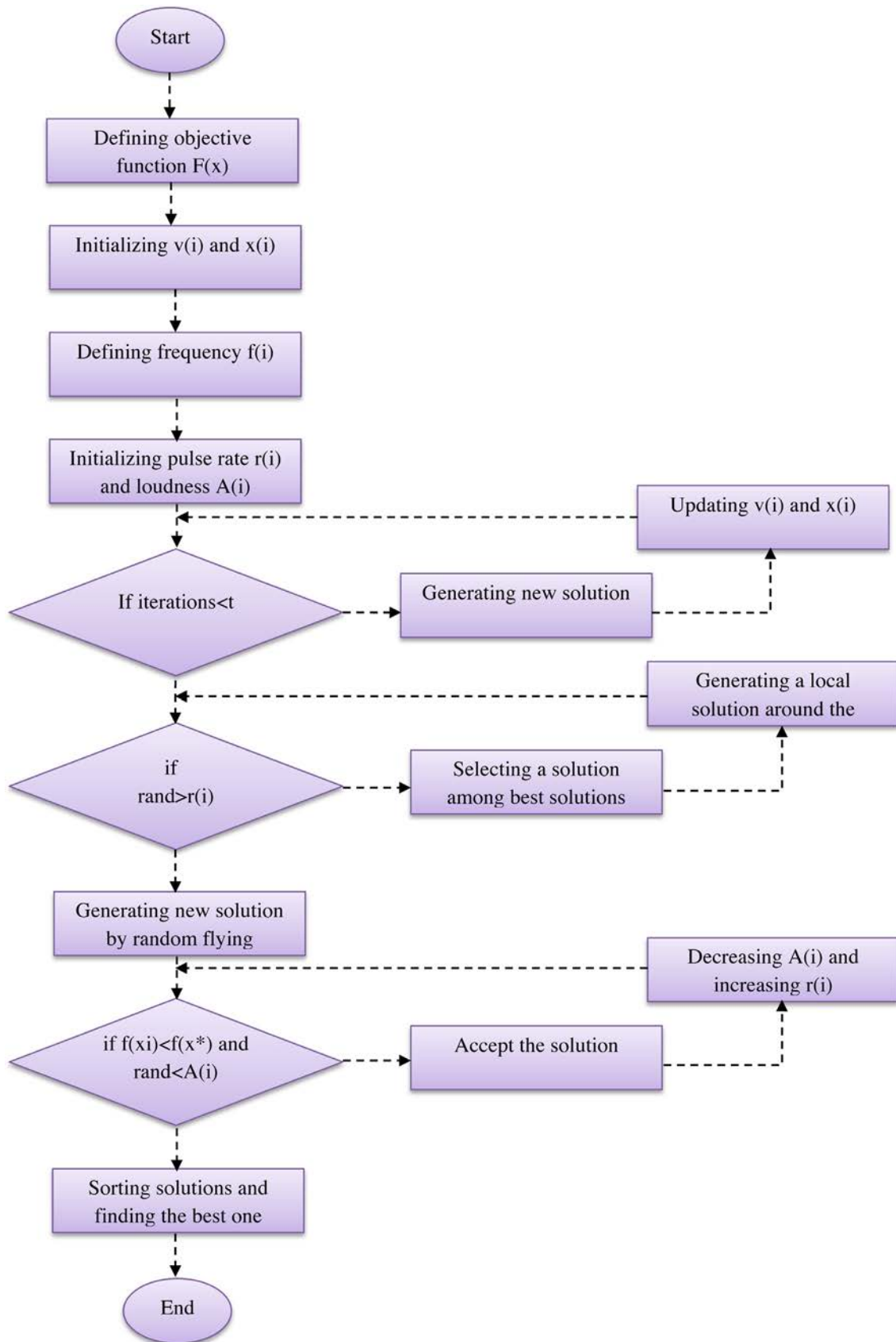
Fig. 9. Flowchart of proposed BNN.

Fig. 10. Flowchart of bat algorithm.

Table 1
Features used for prediction DAX stock price.

Raw features	Unit	Selected as input feature
Balance on financial account	DM/Euro	
Foreign investment in Germany – portfolio investment – total – balance	DM/Euro	
German investment abroad – direct investment – total – balance	DM/Euro	
Goods exports (fob)	DM/Euro	*
Goods – imports (cif)	DM/Euro	*
Income – total – receipts	DM/Euro	*
Indicator of the German economy's price competitiveness against 24 selected industrial countries, based on the deflators of total sales	1Q99 = 100	
Domestic demand (price adjusted)	2005 = 100	*
Exports (price adjusted)	2005 = 100	*
Government consumption (price adjusted)	2005 = 100	*
Gross domestic product (price adjusted)	2005 = 100	*
Imports (price adjusted)	2005 = 100	*
Open	-	*
High	-	*
Low	-	*
Gold price	1 ounce = …USD	
Oil price (real)	Dollars per Barrel	
Mark/USD	DM/USD	
Production of total industry	2005 = 100	*
Consumer price index energy	2005 = 100	*

Table 2
Training and testing datasets.

Sets	From	To	Count
Training	1972-04	2004-07	130
Testing	2004-10	2012-07	32

As mentioned above we used quarterly data of the DAX price from 1972-04 to 2012-07 that plotted in Fig. 11.

4.2. Implementing of BNNMAS model for stock price prediction

After preprocess applied on raw data, the outcome send to the modeling layer. As we used features from different units and ranges, the data normalized at the start point.

We used a six neuron neural network and trained the normalized data with the bat algorithm described in Section 3.3.3.2. Results of implementing the model show that BNNMAS is able to cope with the fluctuation of stock market prices and it also achieves good prediction accuracy in comparing with other models. The results of prediction for training and testing data are shown in Fig. 12. As it is clear the upper figure points to the training phase (to train the designed intelligent model to follow the trends) where the blue line shows real value in historical data and the red line is the trained outcome of our BA-NN model. Next figure represents the testing phase where BA-NN predictions showed by the red line and to comparison with the actual data the real value of DAX stock price during the test period (8 years) plotted by blue line. In a glance view it can be realized and claimed that the BA-NN works properly.

4.3. Performance analysis of proposed BNNMAS model

To demonstrate the feasibility and effectiveness of the proposed model, experiment on predicting DAX stock price is implemented.

In the actual of prediction of stock price experimental results observed through two major points of view: (1) relative prediction performance and (2) absolute prediction performance [31].

Relative prediction performance refers to the stability of the prediction while absolute prediction performance focuses on the algorithm to distinguish whose predicted outcomes are the most accurate.

Observation dates are from April, 1972 to July, 2012. The last 32 data are used to verify the prediction capability of the prediction procedure.

For calculating the absolute prediction performance we need to represent prediction error to sorting models based on their degree of accuracy. The model with the closest prediction error ratio to zero obtains the greatest degree of accuracy. For the purpose of evaluat-

After implementing the layer two sixteen of them remained for predicting DAX stock price. As Arnott et al. [51] mentioned in the book named "Fundamental Indexation" weighting by fundamental factors avoids the pitfalls of equal weighting while still removing the claimed systematic inefficiency of capitalization weighting. Gryc [52] showed that for forecasting long term periods fundamental analysis give a better prediction comparing to technical analysis. Consequently we used fundamental data. Attributes which passed the layer two filtering are named and described in Table 1 and details about training and testing datasets are given in Table 2.

4.1.1. The DAX

The DAX is a (German) stock market index consisting of 30 major German companies trading on the Frankfurt stock exchange. DAX uses capitalization-weighted weighting method. More information about DAX can be finding at: duetsch-boerse.com.
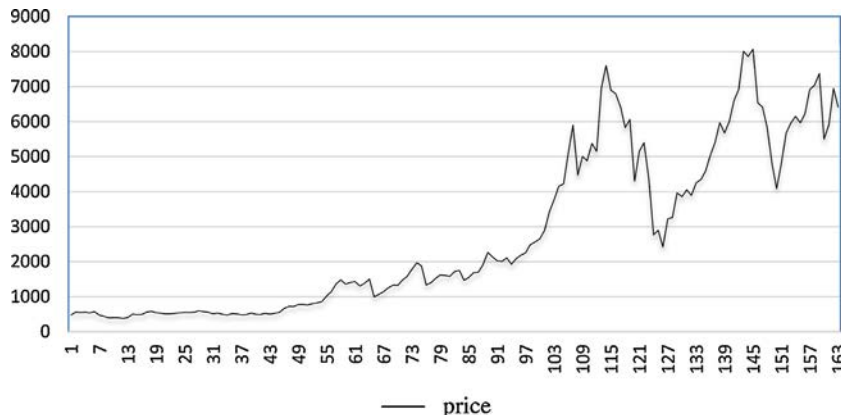

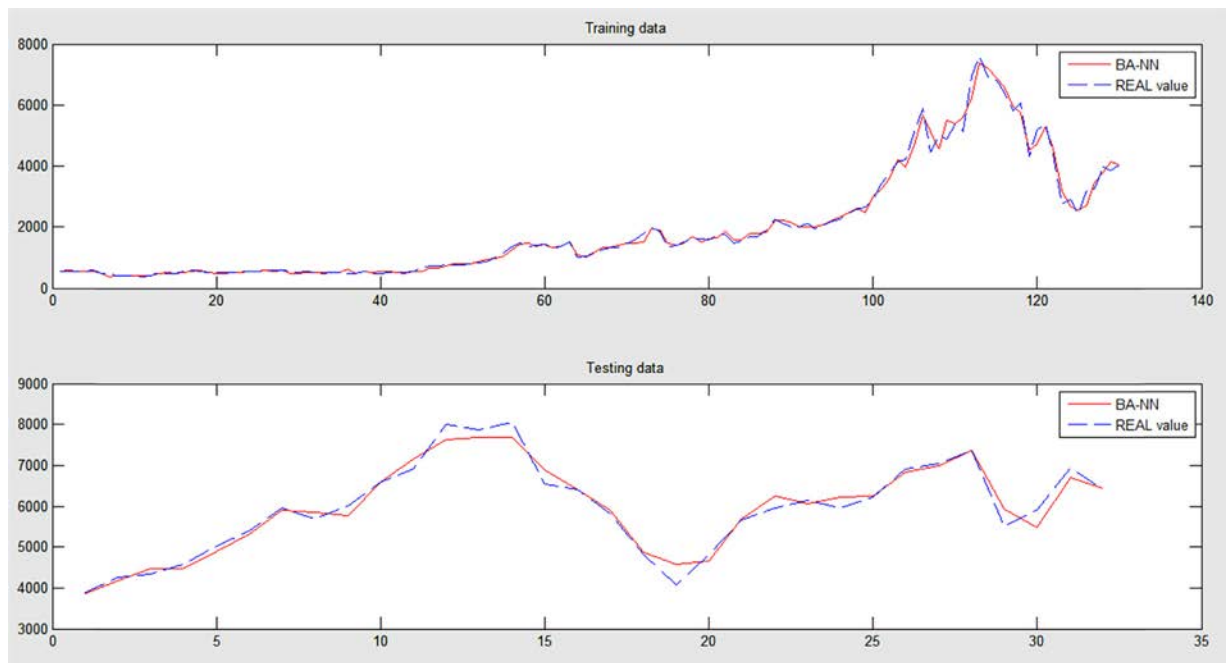**Fig. 11.** The DAX stock price historical data from 1972-04 to 2012-07.

Fig. 12. Experimental results of BNNMAS.

ing BNNMAS prediction accuracy comprising rise and fall, outputs of this model compared with other models such as GANN (genetic algorithm neural network), GRNN (generalized regression neural network) and exact radial basis network. We used mean absolute percentage error (MAPE) statistic to do this job. MAPE defines by the following equation:

$$\text{MAPE} = 100 \times \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i P_i|}{Y_i} \qquad (11)$$

where P_i stands for the predicted value and Y_i is the actual value and n is the number of observations.

The results of MAPE for all cases are shown in Table 3 (The parameters of each model obtained by testing a wide range of valid values throw trial and error process). As Hassan [53] mentioned, models with the lowest MAPE statistic gain much more profit. We chose to compare our proposed BNNMAS model with some latest intelligent models (genetic neural network (GA-NN)) and standard models (generalized regression neural network and exact radial basis network). So at first we attempt to adjust the parameters of the GA-NN. Fig. 13 shows the lowest MAPE statistic for different

neurons of GA-NN. Other architectural parameters obtain running the GA-NN many times to optimize the parameters (experimentally).

Based on the above chart we decide to use the top three GA-NN models for compare with the proposed model so the GA-NN(3), GA-NN(4) and GA-NN(6) has been selected.

According to Table 3 BNNMAS outperforms all cases: GA-NN(3) (2.84 < 6.76), GA-NN(4) (2.84 < 4.77), GA-NN(6) (2.84 < 8.1512), generalized regression neural network (2.84 < 10.75) and exact radial basis network (2.84 < 12.3) in terms of overall prediction results. The results show that the five models indicate clearly different prediction results.

To evaluate the prediction accuracy a paired t test is carried out between the models mentioned in Table 3. The following hypotheses are proposed:

H0. No meaningful difference exists between BNN and the compared model.

H1. Exist a meaningful difference among BNN and the compared model.

Results of paired t test are shown in Table 4.

Table 3
MAPE statistics for all cases.

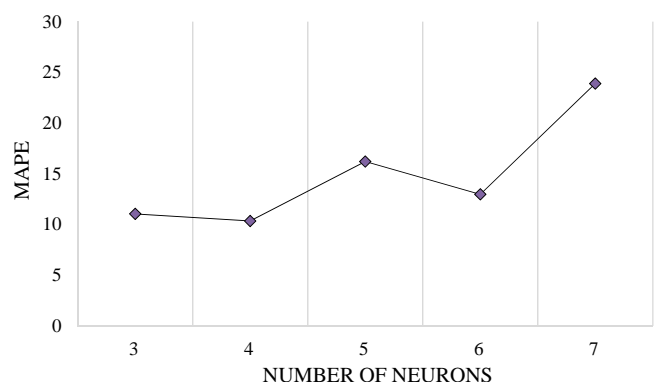| MAPE	Parameters	Model
GA-NN(3)		3.49
GA-NN(4)		4.77
GA-NN(6)		8.1512
Generalized regression neural network (GRNN)		10.75
Exact radial basis network (RBE)		12.3
BNNMAS	Architecture of neural network: (13-6-1) Population = 15 Number of generation = 20 $F_{\min} = 0$ $F_{\max} = 1$ Lamda = 1.2 Alpha = 0.5	2.84

Fig. 13. Performance of GA-NN for different number of neurons.

Table 4
Results of paired *t* test carried out between BNN and the compared model.

Model	Average	Standard deviation	Base model	*t* test	*p*-Value	Conclusion
GA-NN(3)	11.05	4.6886	BNNMAS	2.5732	3×10^{-2}	$\mu_{\text{GA-NN(3)}} > \mu_{\text{BNNMAS}}$
GA-NN(4)	10.3567	3.1547	BNNMAS	5.0141	5×10^{-4}	$\mu_{\text{GA-NN(4)}} > \mu_{\text{BNNMAS}}$
GA-NN(6)	12.9848	3.9172	BNNMAS	5.5436	2.7×10^{-4}	$\mu_{\text{GA-NN(6)}} > \mu_{\text{BNNMAS}}$
GRNN	10.75	–	BNNMAS	24.1418	8×10^{-6}	$\mu_{\text{GRNN}} > \mu_{\text{BNNMAS}}$
RBE	12.3	–	BNNMAS	29.1298	4×10^{-6}	$\mu_{\text{RBE}} > \mu_{\text{BNNMAS}}$
BNNMAS	3.2480	0.28	–	–	–	–

As data used for prediction in all models were same and according to Table 4 which includes paired *t* test on MAPE statistic as a representative for prediction accuracy, H1 hypotheses is proved and we can therefore deduce that using BNNMAS to predict the closing price of DAX stock is more in line with actual stock price. In addition as we used different kinds of indicators, containing technical and fundamental indexes, we can claim that our proposed BNNMAS is more effective than other four cases when the dimension of input increases and accordingly the complexity of the problem space grows.

5. Conclusion

In this research we proposed a four layer BNNMAS architecture for dealing with the distributed nature of stock prediction problem. We used multi-agent approach to create autonomous and independent subtasks to design an accurate prediction model. Our BNNMAS uses preprocess methods in a parallel way such as data normalization, time lag selection and feature selection. We applied a recently developed metaheuristic method named bat to train an artificial neural network. To evaluate the prediction accuracy we tested our model using 1972–2012 quarterly data to predict DAX stock price. Calculating MAPE statistic showed that our proposed BNNMAS model significantly works better than some other models such as GANN, etc. Moreover our test data containing eight years end to 2012, and because of the financial crisis happened in this period of time the good accuracy reached by BNNMAS showed the robust capability of the model. In addition we focused on fundamental data to improve long term prediction accuracy refers to researches did in the literature. So investors are able to deal with the uncertainty rise from the complexity exist in dataset inherently.

Future research effort will be devoted to (1) applying proposed model on other cases to evaluate the robustness of the model, (2) implementing layer one including gathering quantitative and qualitative data to create a comprehensive dataset, (3) executing scenario planning phase to complete the whole proposed model and offering stocks based on developed scenarios.

References

[1] A. Ayodele, A. Charles, A. Marion, O. Sunday, Stock price prediction using neural network with hybridized market indicators, J. Emerg. Trends Comput. Inf. Sci. 3 (2012) 1–9.

[2] A. Khadjeh Nassirtoussi, T.Y. Wah, D.N.C. Ling, A novel FOREX prediction methodology based on fundamental data, Afr. J. Bus. Manage. 5 (2011) 8322–8330.

[3] H. Haleh, B. Akbari Moghaddam, S. Ebrahimijam, A new approach to forecasting stock price with EKF data fusion, Int. J. Trade Econ. Finance 2 (2011) 109–114.

[4] M.H. Fazel Zarandi, E. Hadavandi, I.B. Turksen, A hybrid fuzzy intelligent agent-based system for stock price prediction, Int. J. Intell. Syst. 27 (2012) 1–23.

[5] J. Zhang, L. Cheng, H. Wang, A multi-agent-based decision support system for bankruptcy contagion effects, Expert Syst. Appl. 39 (2012) 5920–5934.

[6] K. Khan, A. Sahai, A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context, Intell. Syst. Appl. 7 (2012) 23–29.

[7] B.J. Vanstone, G. Finnie, C. Tan, Applying fundamental analysis and neural networks in the Australian stockmarket, in: Information Technology Papers, 2004.

[8] C.F. Tsai, S.P. Wang, Stock price forecasting by hybrid machine learning techniques, in: Proceedings of the International Multiconference of Engineers and Computer Scientists, Hong Kong, 2009.

[9] Y.Q. Zhang, S. Akkaladevi, G. Vachtsevanos, T.Y. Lin, Granular neural web agents for stock prediction, Soft Comput. 6 (2002) 406–413.

[10] K. Ali Albashiri, F. Coenen, P. Leng, Emads: an extendible multi-agent data miner, Knowl. Based Syst. 22 (2009) 523–528.

[11] I. Czarnowski, P. Jedrzejowicz, An agent-based framework for distributed learning, Eng. Appl. Artif. Intell. 24 (2011) 93–102.

[12] P. Yang, L. Tao, L. Xu, Z. ZiliZhang, Multi agent framework for bio-data mining, in: P. Wen (Ed.), Lecture Notes in Computer Sciences, Springer-Verlag, 2009, pp. 200–207.

[13] M. Wardeh, F. Coenen, T.B. Capon, PISA: a framework for multiagent classification using argumentation, Data Knowl. Eng. 75 (2012) 34–57.

[14] C.G. Ralha, C.V.S. Silva, A multi-agent data mining system for cartel detection in Brazilian government procurement, Expert Syst. Appl. 39 (2012) 11642–11656.

[15] C. Giannella, R. Bhargava, H. Kargupta, Multi-agent systems and distributed data mining, in: Cooperative Information Agents VIII, Springer, Berlin/Heidelberg, 2004, pp. 1–15.

[16] M. Klusch, S. Lodi, G. Moro, Agent-based distributed data mining: the KDEC scheme, in: Intelligent Information Agents, Springer, Berlin/Heidelberg, 2003, pp. 104–122.

[17] J. Gao, J. Denzinger, R.C. James, A cooperative multi-agent data mining model and its application to medical data on diabetes, in: Autonomous Intelligent Systems: Agents and Data Mining, Springer, Berlin/Heidelberg, 2005, pp. 93–107.

[18] S.K. Shukla, M.K. Tiwari, Y.J. Son, Bidding-based multi-agent system for integrated process planning and scheduling: a data-mining and hybrid tabu-SA algorithm-oriented approach, Int. J. Adv. Manuf. Technol. 38 (2008) 163–175.

[19] J.C.d. Silva, C. Giannella, R. Bhargava, H. Karguptab, M. Klusch, Distributed data mining and agents, Eng. Appl. Artif. Intell. 18 (2005) 791–807.

[20] H.B. Zghal, S. Faiz, H.B. Ghezala, A Framework for Data Mining Based Multi-Agent: An Application to Spatial Data, vol. 5, World Academy of Science, Engineering and Technology, 2007.

[21] J.A. Cendón, A.G. Marcos, M.C. Limas, J.O. Meré, A multi-agent data mining system for defect forecasting in a decentralized manufacturing environment, Comput. Intell. Secur. Inf. Syst. 85 (2010) 43–50.

[22] R.S.T. Lee, J.N.K. Liu, iJADE Weather MAN: a weather forecasting system using intelligent multiagent-based fuzzy neuro network, in: IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 34, 2004, pp. 369–377.

[23] J. Jumadinova, P. Dasgupta, A multi-agent system for analyzing the effect of information on prediction markets, Int. J. Intell. Syst. 26 (2011) 383–409.

[24] P.C. Chang, C.H. Liu, A TSK type fuzzy rule based system for stock price prediction, Expert Syst. Appl. 34 (2008) 135–144.

[25] M.H.F. Zarandi, B. Rezaee, I.B. Turksen, E. Neshat, A type-2 fuzzy rule-based expert system model for stock price analysis, Expert Syst. Appl. 36 (2009) 139–154.

[26] C.J. Lu, Integrating independent component analysis-based denoising scheme with neural network for stock price prediction, Expert Syst. Appl. 37 (2010) 7056–7064.

[27] M.A. Boyacioglu, D. Avci, An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange, Expert Syst. Appl. 37 (2010) 7908–7912.

[28] Y. Kara, M.A. Boyacioglu, Ö.K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul stock exchange, Expert Syst. Appl. 38 (2011) 5311–5319.

[29] H.M. Feng, H.C. Chou, Evolutional RBFNs prediction systems generation in the applications of financial time series data, Expert Syst. Appl. 38 (2011) 8285–8292.

[30] C.M. Hsu, A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming, Expert Syst. Appl. 38 (2011) 14026–14036.

[31] T.S. Chang, A comparative study of artificial neural networks, and decision trees for digital game content stocks price prediction, Expert Syst. Appl. 38 (2011) 14846–14851.

[32] P.C. Chang, D.d. Wang, C.l. Zhou, A novel model by evolving partially connected neural network for stock price trend forecasting, Expert Syst. Appl. 39 (2012) 611–620.

[33] W. Kettera, J. Collinsb, M. Ginib, A. Guptac, P. Schraterb, Detecting and forecasting economic regimes in multi-agent automated exchanges, Decis. Support Syst. 47 (2009) 307–318.

[34] Š. Raudys, I. Zliobaite, The multi-agent system for prediction of financial time series, in: Artificial Intelligence and Soft Computing – ICAISC 2006, Springer, Berlin/Heidelberg, 2006, pp. 653–662.

[35] J.R. Wei, J.P. Huang, P.M. Hui, An agent-based model of stock markets incorporating momentum investors, Physica A: Stat. Mech. Appl. 392 (2013) 2728–2735.

[36] B. Cui, H. Wang, K. Ye, J. Yan, Intelligent agent-assisted adaptive order simulation system in the artificial stock market, Expert Syst. Appl. 39 (2012) 8890–8898.

[37] Y. Li, W. Zhang, Y. Zhang, X. Zhang, X. Xiong, Calibration of the agent-based continuous double auction stock market by scaling analysis, Inf. Sci. 256 (2012) 46–56.

[38] B.D. Kluger, M.E. McBride, Intraday trading patterns in an intelligent autonomous agent-based stock market, J. Econ. Behav. Organ. 79 (2011) 226–245.

[39] L. Ponta, S. Pastore, S. Cincotti, Information-based multi-assets artificial stock market with heterogeneous agents, Nonlinear Anal.: Real World Appl. 12 (2011) 1235–1242.

[40] C.X. Yang, R. Wang, S. Hu, Modeling and analysis of an agent-based model for Chinese stock market, Phys. Lett. A 377 (2013) 2041–2046.

[41] L. Padgham, M. Winikoff, Developing Intelligent Agent Systems: A Practical Guide, John Wiley & Sons Ltd., England, 2004.

[42] M.V. Sokolova, A. Fernández-Caballero, Evaluation of environmental impact upon human health with DeciMaS framework, Expert Syst. Appl. 39 (2012) 3469–3483.

[43] E. Parras-Gutierrez, V.M. Rivas, M. Garcia-Arenas, M.J.d. Jesus, Short, medium and long term forecasting of time series using the L-Co-R algorithm, Neurocomputing 128 (2014) 433–446.

[44] P. Winker, Optimized multivariate lag structure selection, Comput. Econ. 16 (2000) 87–103.

[45] J. Shahrabi, E. Hadavandi, S. Asadi, Developing a hybrid intelligent model for forecasting problems: case study of tourism demand time series, Knowl.-Based Syst. 43 (2013) 112–122.

[46] K.P. Burnham, D.R. Anderson, Multimodel inference: understanding AIC and BIC in model selection, Sociol. Methods Res. 33 (2004) 261–304.

[47] J.P. Teixeira, P.O. Fernandesa, Tourism time series forecast – different ANN architectures with time index input, Procedia Technol. 5 (2012) 445–454.

[48] J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (1991) 246–257.

[49] G. Atsalakis, K. Valavanis, Surveying stock market forecasting techniques – Part II: Soft computing methods, Expert Syst. Appl. 36 (2009) 5932–5941.

[50] X.S. Yang, A new metaheuristic bat-inspired algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Springer, Berlin/Heidelberg, 2010, pp. 65–74.

[51] R.D. Arnott, J. Hsu, P. Moore, Fundamental indexation, Financ. Anal. J. 61 (2005) 83–99.

[52] W. Gryc, Neural Network predictions of Stock Price Fluctuations, Technical Report, 2010 http://i2r.org/nnstocks.pdf

[53] R. Hassan, A combination of hidden Markov model and fuzzy model for stock market forecasting, Neurocomputing 72 (2009) 3439–3446.