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

Reza Hafezi \textsuperscript{a,c}, Jamal Shahrabi \textsuperscript{b,∗}, Esmaeil Hadavandi \textsuperscript{b}

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

\begin{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.
\end{abstract}

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\cite{1}. It is not a secret that stock market movements are reactive to external and internal factors as political, economic and even social\cite{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\cite{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\cite{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\cite{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\cite{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 date 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 $y(t-1), y(t-2), \ldots, y(t-d)$ 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


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 base system for stock price prediction. Fazel Zarandi et al. [25] presented a type-2 fuzzy rule base 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 indexes 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 determine 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 centred 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 this 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, perceptions, 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, perceptions 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.
In this research we used “min–max” normalization method to reform data set by following equation:

\[
\text{Normalized data} = \frac{\text{x}(i) - \text{Min}(\text{x})}{\text{Max}(\text{x}) - \text{Min}(\text{x})}
\]

where \(x(i)\) is \((i)\)th element in the column and \(\text{Min}(\text{x})\) minimum and \(\text{Max}(\text{x})\) 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 criterion (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.
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:

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

(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]. \( pf(N) \) 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.
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
\[ r = \frac{\sum_i [x(i) - mx] \times (y(i - d) - my)]}{\sqrt{\sum_i (x(i) - mx)^2 \times \sum_j (y(i - d) - my)^2}} \]  

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 preprocessing 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 \(b_j\) 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.
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 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 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 \( v_i \) in a d-dimension search space. The initial pulse frequency \( f_i \) calculated by the following equation:

\[
f = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) \beta
\]

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

The velocity \( v_i \) update at each step:

\[
v_i^{t+1} = v_i^t + \Delta x_i^t + \Delta y_i^t
\]

Here \( x_i \) is the current global best solution.

The new solution \( x_i^{t+1} \) at each time step is given by:

\[
x_i^{t+1} = x_i^{t-1} + v_i^t
\]

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:

\[
x_{\text{new}} = x_{\text{old}} + \varepsilon A_i^t
\]

where \( \varepsilon \in [-1, 1] \) is a random number and \( A_i^t \) 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.

\[
A_i^{t+1} = \alpha A_i
\]

\[
r_i^{t+1} = r_i^0 \left[ 1 - \exp(-y r_i^0) \right]
\]

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

\[
A_i^t \to 0 \quad \text{and} \quad r_i^t \to r_i^0
\]

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

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

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. Gyc [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.

Table 2

<table>
<thead>
<tr>
<th>Sets</th>
<th>From</th>
<th>To</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>1972-04</td>
<td>2004-07</td>
<td>130</td>
</tr>
<tr>
<td>Testing</td>
<td>2004-10</td>
<td>2012-07</td>
<td>32</td>
</tr>
</tbody>
</table>

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-
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:

\[
\text{MAPE} = 100 \times \frac{1}{n} \sum_{i=1}^{n} \left| \frac{Y_i - P_i}{Y_i} \right|
\]

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.
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 5. Conclusion dimension of input increases and accordingly the complexity of the problem space grows.

Table 4

<table>
<thead>
<tr>
<th>Model</th>
<th>Average</th>
<th>Standard deviation</th>
<th>Base model</th>
<th>t test</th>
<th>p-Value</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA-NN(3)</td>
<td>11.05</td>
<td>4.6886</td>
<td>BNNMAS</td>
<td>2.5732</td>
<td>3 × 10⁻²</td>
<td>μ_GA-NN(3) &gt; BNNMAS</td>
</tr>
<tr>
<td>GA-NN(4)</td>
<td>10.3567</td>
<td>3.1547</td>
<td>BNNMAS</td>
<td>5.0141</td>
<td>5 × 10⁻²</td>
<td>μ_GA-NN(4) &gt; BNNMAS</td>
</tr>
<tr>
<td>GA-NN(6)</td>
<td>12.9688</td>
<td>3.9172</td>
<td>BNNMAS</td>
<td>5.5436</td>
<td>2.7 × 10⁻⁴</td>
<td>μ_GA-NN(6) &gt; BNNMAS</td>
</tr>
<tr>
<td>GRNN</td>
<td>10.75</td>
<td>–</td>
<td>BNNMAS</td>
<td>24.1418</td>
<td>8 × 10⁻⁴</td>
<td>μ_GRNN &gt; BNNMAS</td>
</tr>
<tr>
<td>RBE</td>
<td>12.3</td>
<td>–</td>
<td>BNNMAS</td>
<td>29.1298</td>
<td>4 × 10⁻⁶</td>
<td>μ_RBE &gt; BNNMAS</td>
</tr>
<tr>
<td>BNNMAS</td>
<td>3.2480</td>
<td>0.28</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

References


