BULGARIAN ACADEMY OF SCIENCES

Institute of Information and Communication Technologies

RESEARCH ON NEURAL NETWORKS BASED CAPITAL MARKETS FORECASTING MODELS

ABSTRACT

Thesis for awarding educational and scientific degree PhD

Scientific Field: **4. Natural sciences, mathematics and informatics**

Professional Area: **4.6. Computer sciences**

Scientific Specialty: **01.01.12 Informatics**

Veselin Lazarov Shahpazov

Scientific Advisor: **Acad. Ivan Popchev**

Sofia

2019

Thesis Data:

- Number of Pages: 148
- Number of Figures: 21
- Number of Tables: 12
- Number of References: 139
- Number of Author’s publications on the subject of the Thesis: 6

INTRODUCTION

Various methods have been proposed and used to forecast the capital market over the years based on technical analysis, fundamental analysis and statistical indicators with varying degrees of success. However, no technique or combination of techniques has been successful enough to "conquer the market" permanently, guessing its direction and the magnitude of its movements.

With computing technology advancements and developments in the field of artificial neural networks, researchers and investors are hoping to successfully acquire the ability to untangle complex "market nodes" that will allow them to build a successful trading strategy. This PhD Thesis will look at some traditional stock market and other financial instruments traded on the capital market forecasting techniques, focusing on why they are not sufficient for constant correct forecasting and how neural networks are used and could be used to improve the forecasting process.

Among the many techniques in the field of Artificial Intelligence, the ones that best deal with uncertainty are neural networks. Dealing with financial uncertainty involves a basic recognition of trends in databases and the use of these trends to predict future events. Neural networks handle issues such as insecurity and risk, better than other Artificial intelligence techniques because they work well with large and imprecise databases. Unlike expert systems and rule-based systems, neural networks are not so clear and the results obtained through their use cannot be so easily explained, which makes them difficult to interpret.

Neural networks are used to predict stock market prices as they are able to learn nonlinear relationships between the input of the system and its output. Contrary to the "Effective Market Hypothesis" thesis, several researchers argue that the stock market and other financial markets are chaotic systems. Chaos is a non-linear deterministic process that has the characteristics of being random because it cannot be easily expressed. Using the ability of neural networks to study chaotic, nonlinear systems, it may be possible to achieve better results than traditional analysis and other computer-based forecasting methods. Because the use of neural networks in the financial sphere is so extensive, this work will focus on using them to predict stock market equity indices, especifically, the onese on the Bulgarian market.

The main tasks of the PhD Thesis as defined in its title, is to conduct a study on models for forecasting the capital market with neural networks. Formulated are the following six tasks, the solution of which will lead to achieving of this goal:

- Research of the formation, development and ability of neural networks to forecast the capital market.
- Analysis of the methods for forecasting capital markets with neural networks.
- Researching the Bulgarian stock market as a subject of forecasting.
- Formulating a hybrid model for capital market forecasting.
- Carrying out experiments with the hybrid model in the real life conditions on the Bulgarian capital market.
- Summary of the achieved results and future development of the model.

CHAPTER I: CAPITAL MARKETS FORECASTING MODELS

In this chapter, the traditional analytical methods used to forecast capital markets are discussed: Fundamental Analysis, Technical Analysis, Dynamic Row Forecasting, Effective Market Hypothesis, Chaos Theory, and Other Capital Market Forecasting Techniques.

Stock exchanges, as they are most commonly called, are official markets for financial instruments, which are governed by rules. Another commonly used term is a Multilateral Trading Facilities - organized by a market operator, through which the trading of financial instruments is carried out in manner organized by the market operator and in accordance with the national and/or international legislation. Until recently, official markets were the only trading venues, but this has changed after the introduction of the Markets in Financial Instruments Directive, known as MiFID. This directive decentralized this industry and allowed the organization of Multilateral Trading Facilities, which significantly changed the marketplace by improving liquidity, thus complicating access to such venues and causing the mass introduction of algorithmic trading. There is only one such multilateral stock trading system in Bulgaria, organized by the investment intermediary Capman AD called MTF Sofia.

Artificial neurons have been developed to model biological neurons in the human body. The artificial neural network is made up of numerous artificial neurons that appear as computational elements, connected together by weighted links and organized in layers. To model the natural process in the human body with respect to the electrical activity of a neuron like this: a neuron receives electrical signals from other neurons, collects all these signals to calculate all the energy received, meanwhile part of that energy is lost, to overcome the natural resistance level of the neuron and the rest of the energy is relayed to the next neurons from the network

Perhaps the first work in the vast field of stock market price forecasting using artificial neural networks, was the work of Hubert White in 1988, the author of which seeks to refute the Effective Market Hypothesis. In its simplest form, the hypothesis states that asset price movements are random, which means that price movements are absolutely impossible to guess in advance, based on publicly available information such as market price and market volume and historical prices. The purpose of the paper was to illustrate how the search for any patterns that may indicate the future movement in the price of IBM common stock using artificial neural networks and the change in stock price, can be successful. The topology of the network used is the standard feed forward network with one hidden layer with full interconnectivity, with the activation of the hidden layer transmitted to the output. The author uses a network with three layers, five inputs and five neurons in the hidden layer. The choice of five hidden units is a compromise in the need to include enough hidden units so that the simplest nonlinear regularities can be found by the network. The training was done using the backpropagation algorithm, one of the most popular to this day.

Research in this field has grown exponentially over time. The models are enriched by artificial neural networks of various types and structures. Research objects vary, with the most common and influential indices of traded stocks being the most common target Indices such as the Dow Jones Industrial Average, S&P 500, Nasdaq, Euro Stoxx 50, FTSE 100, DAX 30, Nikkei 225, CSI 300 and many others. The data used as inputs in most cases is price information, data for traded volumes, technical indicators etc. In some situations, authors resort to more non-standard data, such as analyzing minutes of central bank meetings and even sentiment in specialized social networks.

In addition, Chapter One gives an overview of the stages in the development of neural networks and the use of other techniques in artificial intelligence tools for forecasting financial markets and capital markets in particular.

CAPTER II: ANALYSIS OF THE NEURAL NETWORKS BASED CAPITAL MARKETS FORECASTING METHODS

The neurons in the neural networks are organized in layers. They can be of three types: input, output and hidden. The input layer receives information from the external environment only with each neuron corresponding to an input variable. No calculations are made in the input layer, it transmits the information to the next layer. The output layer produces the final results that are fed from the network out of the system. Each neuron in the output layer corresponds to the values of the predicted variable. The layers located between the input and output layers are called hidden because they do not come into direct contact with the external environment. The hidden layers are used entirely for analytical purposes. Their function is to detect existing relationships between input and output variables.

The architecture, also known as the topology of an artificial neural network, represents its organization, namely: the number of layers, the number of elements (neurons) in each of the layers, and the way the neurons are connected to each other.

Other intrinsic features are the direction of movement of the information (the direction in which the calculations are made, and the type of relationships between the individual elements.

A neural network with more than one layer of weighted neurons containing one or more hidden layers is called a multilayer neural network, the most commonly used being the multilayer perceptron.

Different information flows result in different types of neural networks. For feedforward neural networks, information moves only in one direction, from the input layer through the hidden layers to the output layer, and there are no return cycles. Feedback neural networks move information in both directions, from the input layer to the output layer and vice versa.

If each element of one layer is connected to all elements of the next layer, the artificial neural network is called completely interconnected; if each element is connected to each element of each layer, the artificial neural network is called fully connected

In this chapter a detailed historical overview of the application of neural networks for forecasting financial markets and stock prices. As can be seen from the graph below that in 20.75% (eleven of the articles) the authors found the performance of artificial neural networks unsatisfactory or less successful than the methods by which they were compared, while in other studies (79.25%) the conclusion is positive. Figure 1 shows schematically this observation.

Figure 1: Positive/Negative results from reviewed papers, according to their authors

Among the architectures used, the most prominent is the multilayer perceptron (MPL), used 37 times. Among the less popular varieties are radial basis function neural networks (RBFNN) used 3 times, probabilistic neural networks (PNN) used 4 times and general regression neural networks (GRNN) only once. In the category of other architectures, where the number is 19, it should be noted that there are some new or at least more popular, in more recent studies structures, such as short and long memory networks and deep learning neural network, along with considered more traditional recurrent neural networks (RANN). Figure 2 shows the distribution of neural network types in the studies considered.

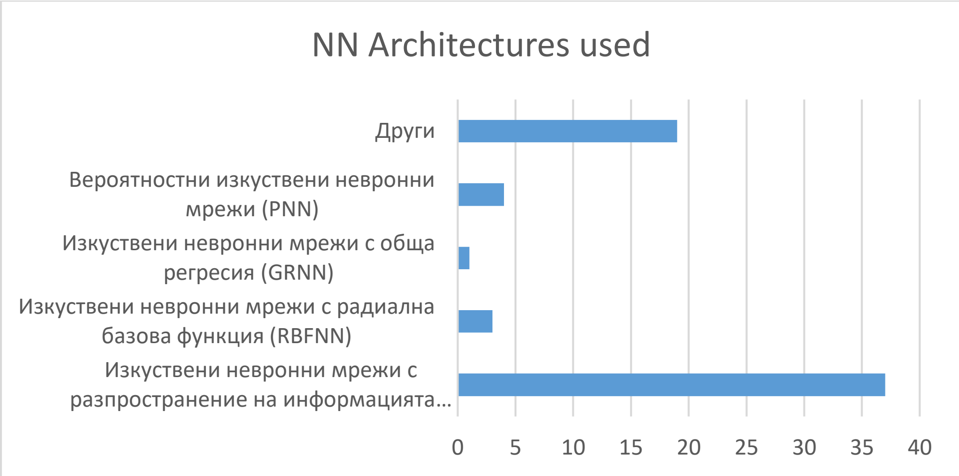

Figure 2: Used neural network architecture

The forecasted types of markets are another reviewed criterion as developed markets have been the subject 32 times followed by emerging markets with 14 forecasts. It is no coincidence that the author's idea is to focus on the niche of unexplored and little-known markets like the Bulgarian, presented by a study. Figure 3 shows the breakdown by type of labor market.

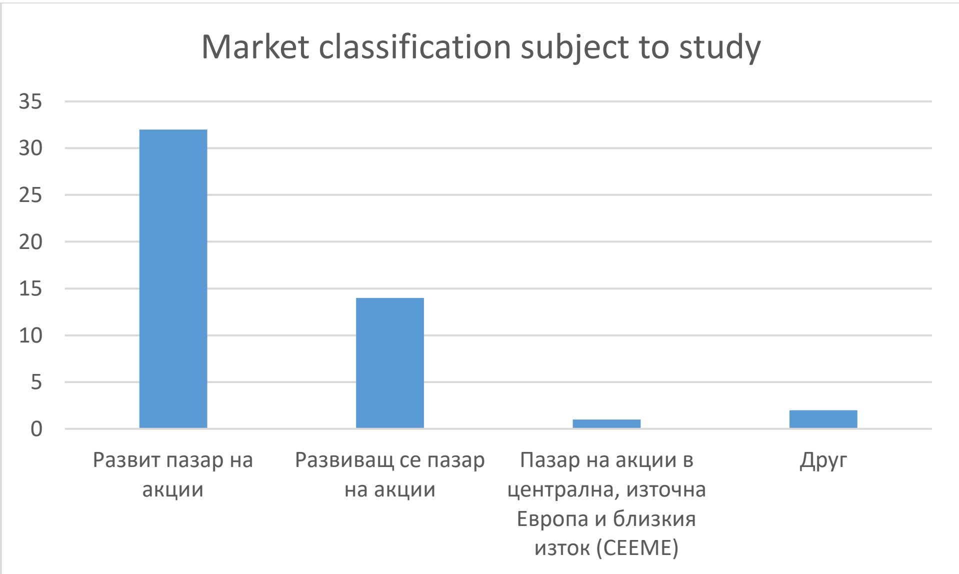

Figure 3: Forecasted type of market

The scientific papers reviewed tend to be the subject of research into major global markets in developed countries and stocks and indices of stocks of multinationals traded in some cases in several different markets, as well as broader industry indices .

Although low liquid and underdeveloped markets, also known as frontier markets present an environment of low liquidity and the dependence on events on the global economic scene is much less pronounced, there is an opportunity to test the ability of artificial neural networks to predict such markets as well.

CHAPTER III: RESULTS FROM RESEARCH ON NEURAL NETWORKS BASED CAPITAL MARKETS FORECASTING MODELS AND PROPOSED NEURAL NETWORKS AND RULE-BASED SYSTEMS HYBRID MODEL FOR FORECASTING THE CAPITAL MARKET

This chapter begins with a brief history of the Bulgarian stock market, describing its structure and classifying it as an illiquid and undeveloped stock market.

The data used for the experiments conducted in this chapter are from the official site of the Bulgarian Stock Exchange and consist of the following information about the SOFIX index: latest value, opening value, highest value of the day, and lowest value of the day, as well as the volume traded. In addition to the price information, five of the most common technical indicators are used in the experiment: 30-day moving average, 60-day moving average, 200-day moving average, 14-day relative strength index and 30-day relative strength index. All data are for the period between 04/01/2010 and 02/28/2013.

After the initial selection of the data, the selected data is divided into three parts, half of which is used for actual training and the rest is divided between test data and validation data or: training data - 50%, test data - 25% and validation data - 25%.

Three different networks are tested starting with most widely used multi-layer perceptron, a feed-forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. A multi-layer perceptron consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one the network utilizes a supervised learning technique called back-propagation for training the network it is the most popular algorithm and is extremely simple to program but tends to converge slowly. It calculates the local gradient of each weight with respect to each case during training. Weights are updated once per training case.

The formula is:

$$\Delta\omega_{ij}(t) = \eta\delta_j o_i + \alpha\Delta\omega_{ij}(t-1) \qquad (1)$$

Where:

η – the learning rate;

δ - the local error gradient;

α - the momentum coefficient;

o_i – the output of the i'th unit.

Thresholds are treated as weights with o_i = -1. The local error gradient calculation depends on whether the unit into which the weights feed is in the output layer or the hidden layers. Local gradients in output layers are the product of the derivatives of the network's error function and the units' activation functions. Local gradients in hidden layers are the weighted sum of the unit's outgoing weights and the local gradients of the units to which these weights connect.

The Conjugate gradient descent is an advanced method of training multilayer perceptron's. It usually performs significantly better than back propagation, and can be used wherever back propagation can be. It is the recommended technique for any network with a large number of weights (more than a few hundred) and/or multiple output units. Conjugate gradient descent is a batch update algorithm: whereas back propagation adjusts the network weights after each case, conjugate gradient descent works out the average gradient of the error surface across all cases before updating the weights once at the end of the epoch.

In the process of back propagation, the gradient vector of the error surface is calculated. This vector points in the direction of the steepest descent from the current point, so if we move in this direction even for a short distance, we will reduce the error. The sequence of such movements will eventually find some minimum. The difficult part is deciding how big the steps should be. Big steps can get you closer to the minimum value of the error faster, but they can also skip it, or (if the surface of the error is very eccentric) go in the wrong direction. Usually, the algorithm is modified by including an inertia moment in the training. This encourages movement in a certain direction, so that if several steps are taken in the same direction, the algorithm increases its speed, which sometimes enables it to escape from local minima as well as to move quickly over flat spots and fabrics. The initial network configuration is random, and training stops when a number of epochs expire, when the error reaches an acceptable level, or when the error stops improving.

One major problem with this type of training is that it does not actually reduce the error that is really of interest - such as the expected mistake the network will make when new cases other than the one it is trained on are presented. In other words, the most desirable ability of a network is its ability to generalize or generalize to new situations.

The most important manifestation of this distinction is the problem of overfitting. An artificial neural network with more weights models a more complex function and is therefore prone to over-training. A smaller weight network may not be powerful enough to model the underlying function. For example, a network without hidden layers actually models a simple linear function. When learning the network excessively during the training process, it is usually recommended to reduce the number of hidden units and / or hidden layers as the network is too powerful for the specific problem. In contrast, if the network is not powerful enough to model (has less than the required optimal number of units and hidden layers) the core function will not result in over-training, but neither training errors nor error test will fall to a satisfactory level.

The final model is tested with validation data to ensure that its learning and testing results are real. Of course, to fulfill this role successfully, the validation set must be used only once. If it, in turn, is used to adjust the network parameters and repeat the learning process, then the process becomes a data selection.

The process of data separation is highly subjective, and in theory there are various different theories as to how it should be conducted. This problem can be overcome by reclassifying individual databases. Experiments can be performed using different selections of available data in training, testing and validation groups. There are a number of approaches to this transformation of subsets, including random calculations (Monte Carlo), cross validation, and bootstrap. Another approach is to retain only the best networks found during the test and validation process, based on different samples, but to average their results, which will reduce the subjective factor in the allocation of data sets.

The most widely used activation function for the output layer are the sigmoid and hyperbolic functions. In this paper, the sigmoid transfer function is employed and is given by

$$E(t) = \frac{1}{1+e^{-t}} \qquad (2)$$

The error of a particular network configuration can be determined when all network training cycles are completed by comparing the actual generated predicted value with the desired or target results. The differences are combined by an error function to calculate the network error. The most common error measure (used for regression problems) is the root mean square error where the individual errors of the output units are in each case squared and summed together.

The criteria which will evaluate the Neural Networks performance will be the error of the network on the subsets used during training (Root Mean Square-RMS).

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} (\hat{\delta}_i - \delta_i)^2}{n}} \qquad (3)$$

As many authors have recommended that in order not to slow down the constructed model and to delay its calculation time further, the data submitted to be of smaller order was considered as having to reorganize the data from values in daily change of the original ones. values. This is determined by the number of input variables, which are ten over a period of two years and two months.

Initial study results are not particularly good. They show an interesting trend, namely that networks that use all ten input parameters are constantly performing much worse than those that isolate some of the input parameters. Models that base their forecasts solely on the last price show the best results in terms of learning error. Moreover, in no case did a network with more than one hidden layer perform better than one with a single hidden layer.

Several types of architecture were tested, with the best performing artificial neural network having three layers, with one element in the input layer responsible for the last-change parameter, which was the ultimate goal of forecasting and using 43 lagged steps. This model has 7 elements in the hidden layer, and one element in the output layer corresponding to the predicted input variable. The structure was selected using the model structure optimization feature available in the software product provided, which proved to be more efficient and much faster than manual adjustments made by the author. Training on the best performing network at this stage involves 100 cycles (epochs) of back propagation and 23 cycles of conjugate gradient descent. This led to a test error of 0.119279, which at first glance may lead to two conclusions. The first is that the network has not been over-trained and the second is that the result certainly does not look impressive.

After taking into account the achieved initial results, it is evident that some adjustments should be made to the model, which is why the author turned to the input data. Taking into account the peculiarities of the Bulgarian stock market, and in particular its illiquidity, which allows individual transactions not executed on a purely market principle to affect the overall value and an index change, it was considered appropriate to undertake additional preliminary data processing in the form of averaging of the input values, respectively, of the index changes starting from 3 days averaging. The results of such experiments are presented in the following table:

	Network structure Input-hidden-output	Learning samples BP/CGD	Test error RMS
3 days smoothing	1(12)-5-1	BP-100; CGD-115	0.076298
4 days smoothing	1(12)-11-1	BP-100; CGD-58	0.089827
5 days smoothing	1(25)-15-1	BP-100; CGD-48	0.067025
6 days smoothing	1(10)-5-1	BP-100; CGD-76	0.075116
7 days smoothing	1(20)-13-1	BP-100; CGD-34	0.057099
8 days smoothing	1(15)-14-1	BP-100; CGD-68	0.057903
9 days smoothing	1(15)-5-1	BP-100; CGD-67	0.064887

Table 1: Results from smoothing input data

It is evident from the results in the table that the averaging performed results and manages to reduce the error in the validating data set significantly. There is a tendency for the error to continue to decrease as the averaging period is extended to 7 days. Then the error starts to grow again and loses, and the continuation of the averaging loses its meaning. The neural network that performed best in the form of the smallest validation error value of 0.057099 was built from a single input layer (and here again the tendency to better represent networks using only the predicted parameter as input), 13 neurons in the hidden layer and one output element. The training was performed on the basis of 34 epochs of conjugated gradient descent and 100 epochs of back propagation and data 20 steps back in time. It can be seen from the table that the epochs of learning by back propagation are limited to 100, due to the time consuming nature of this type of training and the lack of better results when conducting experiments without setting a maximum number of cycles. A figurative representation of the best performing neural network is shown in Figure 4

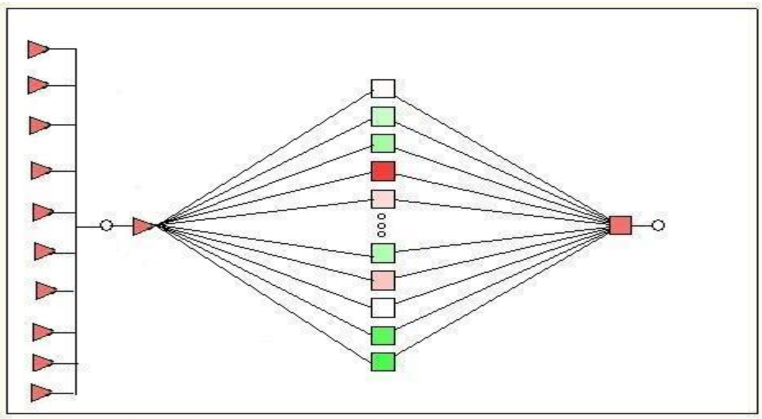

Figure 4: Multilayer perceptron with 10 inputs, 13 hidden neurons and one output

Figure 5 shows the predicted results from actual observations of a multilayer perceptron with 10 inputs, 13 neurons in the hidden layer, and one output element

Figure 5 shows the predicted results from actual observations of a multilayer perceptron with 10 inputs, 13 neurons in the hidden layer, and one output element

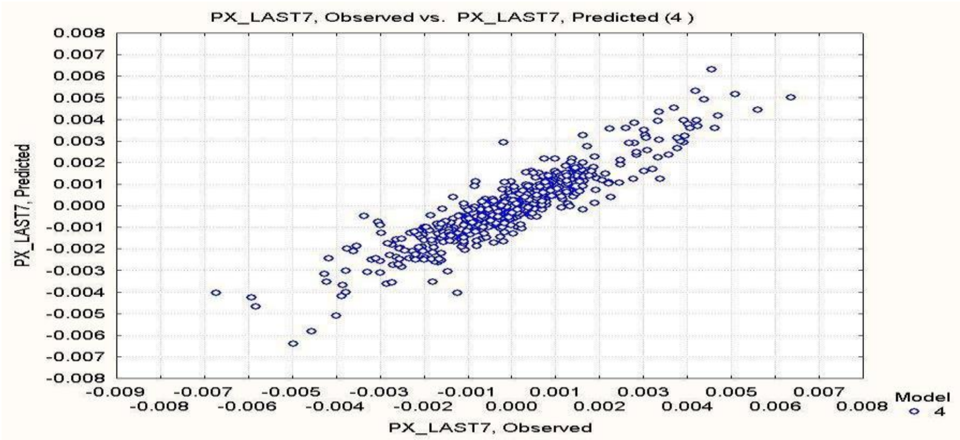

Figure 5: Predicted values versus observed plot

Predicted change (dotted line) versus observed changeis presented on Figure 6.

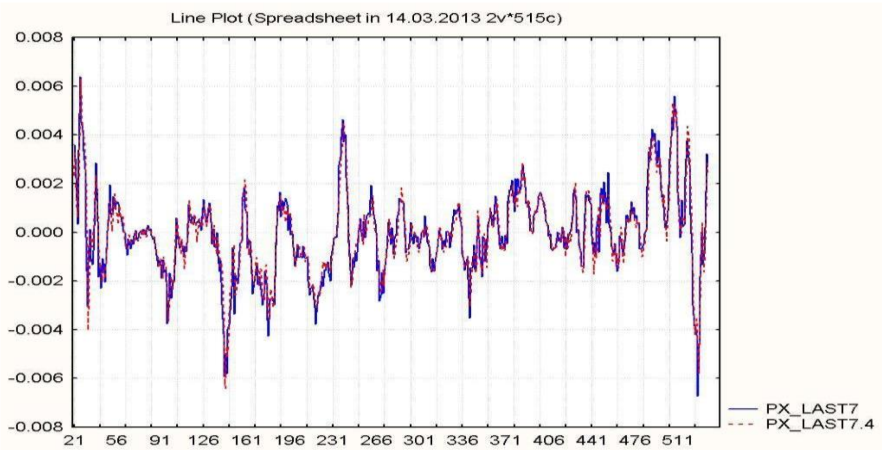

Figure 6: Predicted change (dotted line) versus observed change

Residual error diagram respectively of the multilayer perceptron with 10 inputs, 13 hidden neurons and one output presented on Figure 7.

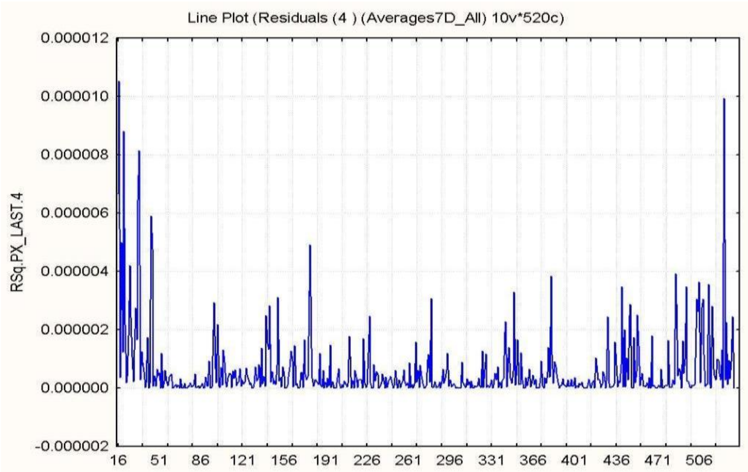

Figure 7: The residual error diagram

An interesting trend can be seen from the residual error graph shown in Figure 7. An increase in the magnitude of the error is observed at the beginning of the period and its end, which can be explained by the composition of the input data presented on the artificial neural network. In the first part of the observed period, the Bulgarian stock market is in an upward trend, with the index rising by more than 23% in about a month. At the end of the observed period, the index first rose with an increase of more than 26% for about a month and a half, and then there was a strong correction (a sharp withdrawal from the achieved high values) in its value of 11% within a few days, which According to the author, the chart also leads to a jump in the deviation of the network's predictions from the actual values of Sofix.

Another observation of the averaged data experiment is the continued poor performance of neural networks using the full set of 10 parameters for training, testing and validation. The best result was obtained from a model using 9 of the variables, with averages over 7 days. The network was made up of 9 input elements responsible for the nine variables used, 21 neurons in the hidden layer and one element in the output layer, trained based on 100 cycles of error back propagation and 81 cycles of conjugated gradient descent. The result for the root mean square error is 0.068935 and is still considered not good enough, especially considering the more complex structure of the neural network and most of the time it takes to train it.

An experiment on two BSE Sofia indexes with different network architectures is presented below.

Radial basis function networks have a number of advantages over MLPs. First, as previously stated, they can model any nonlinear function using a single hidden layer, which removes some design-decisions about numbers of layers. Second, the simple linear transformation in the output layer can be optimized fully using traditional linear modeling techniques, which are fast and do not suffer from problems such as local minima which plague multilayer perceptrons training techniques. Radial basis functions networks can therefore be trained extremely quickly. Figure 8 illustrates the way in which neural networks with radial basis function are represented schematically.

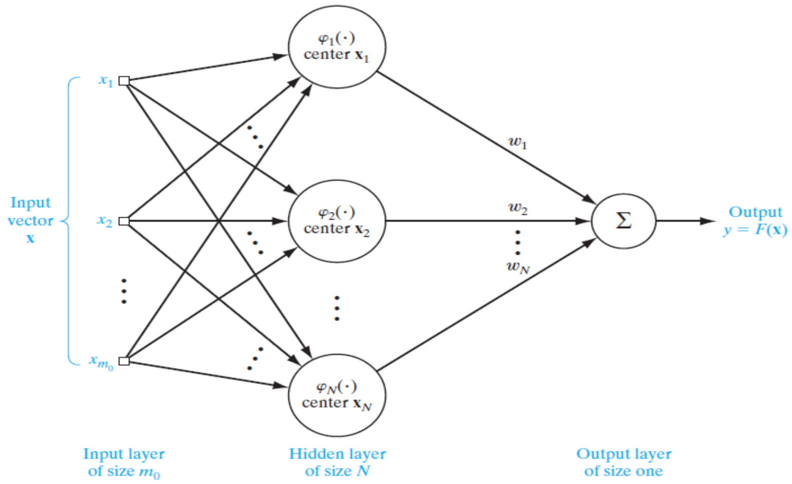

Figure 8: Radial basis function network

Unlike radial basis networks, multilayer perceptron becomes more secure in long-range results. Whether this can be considered an advantage or a disadvantage depends largely on the application, but in general, extrapolation of the multilayer perceptron is considered a weak point (extrapolation away from training data is usually considered dangerous and unjustified).

Networks with a radial basis function are also more sensitive to the scale of the input and have greater difficulty if the number of input units is large. One aspect of the so-called "curse of dimension" is that as the number of predictors increases (due to the increase in the number of inputs), the variance also increases. The higher the value of the variance, the more difficult it is for the predictive algorithm to perform well when the network is fed with new data.

The ratio of the input to the output layer of an artificial neural network with a radial basis function is defined as follows:

$$y(i) = \sum_{j=1}^{K} w_j(n)\exp\left(-\frac{1}{2\sigma^2(n)}\|\mathbf{x}(i) - \boldsymbol{\mu}_j(n)\|^2\right), \qquad i = 1, 2, \ldots, n \tag{4}$$

where $\mu_j(n)$ is the center point of the j-th Gaussian unit, the width σ(n) is common to all the K units, and $w_j(n)$ is the linear weight assigned to the output of the j-th unit; all these parameters are measured at time n- the cost function used to train the network is defined by

$$\mathscr{E} = \frac{1}{2}\sum_{i=1}^{n} e^2(i) \tag{5}$$

where:

$$e(i) = d(i) - y(i)$$

This is a convex function of the linear weights in the output layer, but not convex with respect to the centers and widths of the Gaussian units.

Radial basis function neural networks can also be hybridized in several ways. The radial layer (hidden layer) can be trained using the Kohonen and Learned Vector Quantization training algorithms, which are alternative methods of identifying centers to reflect data distribution, and the source layer (linear or not) can be trained , using any of the iterative point algorithms available to the author.

The third network type used for the forecasting procedure is the general regression neural network. The general regression neural network performs regression where the target variable is continuous as opposed to the probabilistic neural networks (they both have similar architectures), which performs classification where the target variable is categorical. Proposed by D. Spach, they are a one-way training algorithm for passing training information through it, with a highly parallel structure. These types of neural networks are capable of providing a smooth transition from one observation to another even with incomplete data in a multidimensional measurement space. The algorithmic form can be used for any regression problem in which the assumption of linearity is not justified.

The mathematical representation and schematic form of the general regression networks are shown in Figure 9 below

$$\hat{Y}(X) = \frac{\sum_{i=1}^{m} A^i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^{m} B_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \qquad (6)$$

$$\begin{Bmatrix} A^i(k) = A^i(k-1) + Y^j \\ B^i(k) = B^i(k-1) + 1 \end{Bmatrix} \qquad (7)$$

where: m is the number of clusters

$A^i(k)$ и $B^i(k)$ – are the values of the clusters coefficients i after k observations

$A^i(k)$ is the sum of Y values

$B^i(k)$ is the number of patterns for cluster i

σ is the width probability

$$\hat{Y}(X) = \frac{\sum_{i=1}^{m} A^i \exp\left(-\frac{C_i}{\sigma}\right)}{\sum_{i=1}^{m} B^i \exp\left(-\frac{C_i}{\sigma}\right)}. \qquad (8)$$

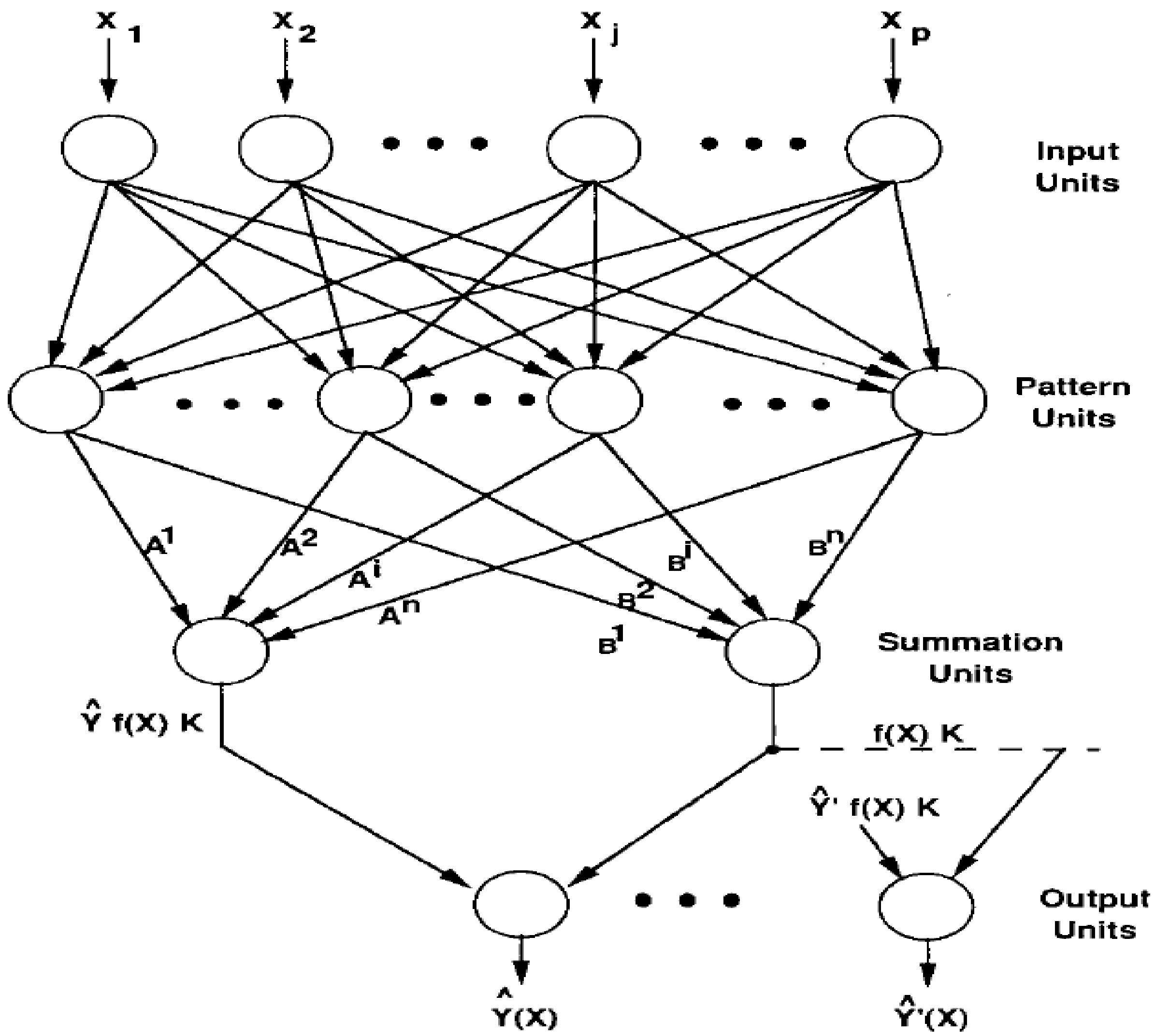

Figure 9: General regression neural network

A general regression neural network performs a regression in which the target variable is continuous, which performs a classification in which the target variable is categorical. The main advantages over multilayer perceptrons are:

- It is usually much faster to train a general regression neural network than a multilayer perceptron network.
- Probabilistic neural network/General regression neural networks often are more accurate than multilayer perceptron networks.
- Probabilistic neural network/General regression neural networks are relatively insensitive to outliers (wild points).
- Probabilistic neural network/General regression neural networks generate accurate predicted target probability scores.

The respective shortfalls of General regression neural networks are:

- Probabilistic neural network/General regression neural networks are slower than multilayer perceptron networks at classifying new cases.
- Probabilistic neural network/General regression neural networks require more memory space to store the model.

The only control factor to choose when training networks with general regression is the smoothing factor (ie, the radial deviation of the Gaussian functions). As with networks with radial basis functions, this factor must be chosen to cause reasonable overlap - too small deviations cause a very tangible approximation, which cannot generalize while too large deviations smooth out the details. The appropriate digit can easily be selected by experiment by selecting a number that results in a low selection error, and fortunately, networks with general regression are not too sensitive to the exact selection of the smoothing factor.

Apart from Sofix, the other index to be considered in this section is BG40. It is based on emission pricing and covers 40 issues of ordinary shares of the companies with the highest number of transactions and the highest average daily turnover over the last 6 months, with both criteria being equally weighted.

The object of the experiment will again be the prediction of the values of the two indices in the future, based on previous information.

The data for the experiment are from the official site of BSE - Sofia AD, composed of price information on a daily basis for the two indices from 04.01.2010 to 28.02.2013 and more precisely: closing value, opening value, max. higher value for the day and lowest value for the day. On the basis of previous studies conducted by numerous researchers, as well as the experience of the previous experiment, it was decided to differently distribute the data in the three subsets, namely training, testing and validation. This was done with 70% of the data set for training, 15% for testing and 15% for validation. This is mainly due to the results of the multilayer perceptron in predicting the Sofix index from earlier and the author's desire to do less pre-processing of the data.

At the beginning of this experiment, it was decided not to perform any refinement of the initially generated databases for Sofix and BG40, due to the fact that the first results of the three types of networks showed no improvement after converting the digital values of the indices into daily changes. Selected criteria for evaluating network performance is again the network error of subsets used during training - the mean square error. In mathematics, the term is explained as a statistical measure of the magnitude of a variable.

The multilayer perceptron was again trained with back propagation and conjugate gradient descent algorithms, while the other two network architectures did the same. Radial basis function networks were trained based on the k-means (K-Means), k-nearest-neighbor (K-Nearest Neighbor) and pseudo-inverse algorithms. Artificial neural networks with general regression were trained with Subsampling algorithm.

The results of the forecasting of the SOFIX index are presented in Table 2 below.

	Neural network name	Training algorithm	Inputs	Hidden (1)	Hidden (2)	Test Error
1	Multilayer perceptron 1	Back-propagation, Conjugate gradient descent	1	3	0	0.023607
2	Multilayer perceptron 2	Back-propagation, Conjugate gradient descent	2	5	0	0.024466
3	Radial Basis Function NN 1	K-Means, K-Nearest Neighbor, Pseudo-inverse	1	42	0	0.001892
4	Radial Basis Function NN 2	K-Means, K-Nearest Neighbor, Pseudo-inverse	4	42	0	0.002034
5	General regression neural network 1	Subsampling algorithm	1	533	2	0.000832
6	General regression neural network 2	Subsampling algorithm	2	533	2	0.000741

Table 2: Forecasting Sofix Index

As can be seen in the test results table, the best result, expressed as the smallest validation error (Test Error), was achieved by neural networks with a common regression. The most successful network consists of 2 elements in the input layer, 553 elements in the first hidden layer, 2 elements in the second hidden layer, and one element in the output layer corresponding to the requested variable. The error generated by the better performing network was 0.000741 and for the second one 0.000832, both of which were trained on the basis of the Subsampling algorithm.

Artificial neural networks with radial basis function show constant and good results, with the two best performing models showing error of 0.001892 and 0.002034 respectively. Structured of 1 and 4 elements in the input layer, both have 42 elements in the hidden layer and one output element corresponding to the desired variable, both of which are trained using the k-means (K-Means) algorithms, k - closest neighbor (K-Nearest Neighbor) and pseudo-inverse. The K-Means algorithm attempts to select the optimal set of points that are located in the centers of training data groups. For given K radial units, it adjusts the positions of the centers so that each training point belongs to the cluster center and is closer to that center than to any other center, and each cluster center is the center of the training points that belong. Once the centers are set, the deviations are set. The magnitude of the deviation, also known as the smoothing factor, determines how prominent the Gaussian functions are. For K-Nearest Neighbor, each unit deviation is individually set at a mean distance to K-Nearest Neighbor. Therefore, the deviations are smaller in tightly filled spaces, retaining details and higher in sparsely filled spaces (interpolating where necessary). After the centers and deviations are set, the output layer is optimized using the standard linear optimization technique - the pseudo-inverse (singular value decomposition) algorithm.

Multilayer perceptron-type networks are the worst of the three models. The results of their two best representatives are 0.023607 and 0.024466, respectively, produced by networks made up of three layers having 1 input, 3 elements in the hidden layer for the first and 5 elements in the hidden layer for the second, and one element in the output layer , corresponding to the requested variable for both models. Trained with 100 epochs and 34 epochs of conjugated gradient descent for the first and 100 epochs of reverse propagation and 34 epochs of conjugated gradient descent for the second, they perform significantly worse.

Graphically, the forecasts of the six networks listed in Table 2 for the selected period are shown in Figure 10

Figure 10: The values of the index and the predicted values by the three different models (two of each) for the last 50 days of the observed period.

At this stage, some surprising facts cannot be ignored. First is the good results that show both neural networks with general regression and those with radial basis function. They seem to do much better in modeling the input data, managing to produce equally good results by isolating a different number of input parameters. For example, the second best network with general regression uses two neurons in its input layer, respectively, responsible for the target variable - the last index value of the day (close value) and the lowest value for the day, while the second best network with radial base function uses all 4 input parameters (except the above two and the opening value and lowest value for the day).

Although the results of the multilayer perceptron were better than those achieved during the first experiment, they remained many times worse than those of the models described above. This trend is visible not only in validating data, but also on the basis of error data during training and network testing. Table 3 presents the error data of the six models during training and in the test data set.

	Neural network name	Training algorithm	Train Error	Select Error
1	Multilayer perceptron 1	back-propagation, conjugate gradient descent	0.020964	0.023328
2	Multilayer perceptron 2	back-propagation, conjugate gradient descent	0.020729	0.023204
3	Radial Basis Function NN 1	K-Means, K-Nearest Neighbor, Pseudo-inverse	0.001732	0.002120
4	Radial Basis Function NN 2	K-Means, K-Nearest Neighbor, Pseudo-inverse	0.001937	0.002068
5	General regression neural network 1	Subsampling algorithm	0.000962	0.000877
6	General regression neural network 2	Subsampling algorithm	0.000785	0.000697

Table 3: Neural network's error during training and testing on Sofix Index data

When predicting the BG40 index, the situation was not much different. Artificial neural networks with general regression were again the most productive and reached the lowest values in the validation set. The lowest error value was 0.004963 achieved by a network consisting of 2 elements in the input layer, 550 elements in the first hidden layer, 2 elements in the second hidden layer, and one element in the output layer corresponding to the predicted variable. This time the input parameters used are the last index value of the day and its highest value for the day. Unlike before, the second best network of this type uses one element in the input layer, 550 elements in the first hidden layer, 2 elements in the second hidden layer, and one element in the output layer corresponding to the predicted variable. The achieved results for its error value is 0.005467. Both networks are trained on the basis of Subsampling algorithm.

The second result is a network architecture with a radial basis function with 2 elements in the input layer, corresponding to the parameters last index of the day and highest index of the day, 30 elements in the hidden layer and one element in the output. This structure also has the second best network of its kind, but it uses all 4 parameters to make its forecasts. The results obtained with respect to the error value are respectively 0.014879 for the first and 0.0155 for the second. Both networks are trained using the k-means (K-Means), k-nearest-neighbor (K-Nearest Neighbor) and pseudo-inverse algorithms.

Again, third place is occupied by multilayer perceptrons, which manage to shorten the distance between their presentation and that of the next model, by utilizing one element in the input layer, one for the better mesh and two elements in the hidden layer of the second network, respectively. one element in the output layer. The results obtained with respect to the error value are respectively 0.032738 for the first and 0.032934 for the second. In this case too, training based on 100 epochs and 28 epochs of conjugate gradient descent for the first and 100 epochs of dissemination and 18 epochs for conjugated gradient descent for the second is not good enough for these models to compete with the others.

A table summarizing the results of the three network architectures used to predict BG40 is shown in Table 4.

	Neural network name	Training algorithm	Inputs	Hidden (1)	Hidden(2)	Test Error
1	Multilayer perceptron 1	Back-propagation, Conjugate gradient descent	1	1	0	0.032738
2	Multilayer perceptron 2	Back-propagation, Conjugate gradient descent	1	2	0	0.032934
3	Radial Basis Function NN 1	K-Means, K-Nearest Neighbor, Pseudo-inverse	2	30	0	0.014879
4	Radial Basis Function NN 2	K-Means, K-Nearest Neighbor, Pseudo-inverse	4	30	0	0.015500
5	General regression neural network 1	Subsampling algorithm	1	550	2	0.005467
6	General regression neural network 2	Subsampling algorithm	2	550	2	0.004963

Table 4: Forecasting BG40 Index

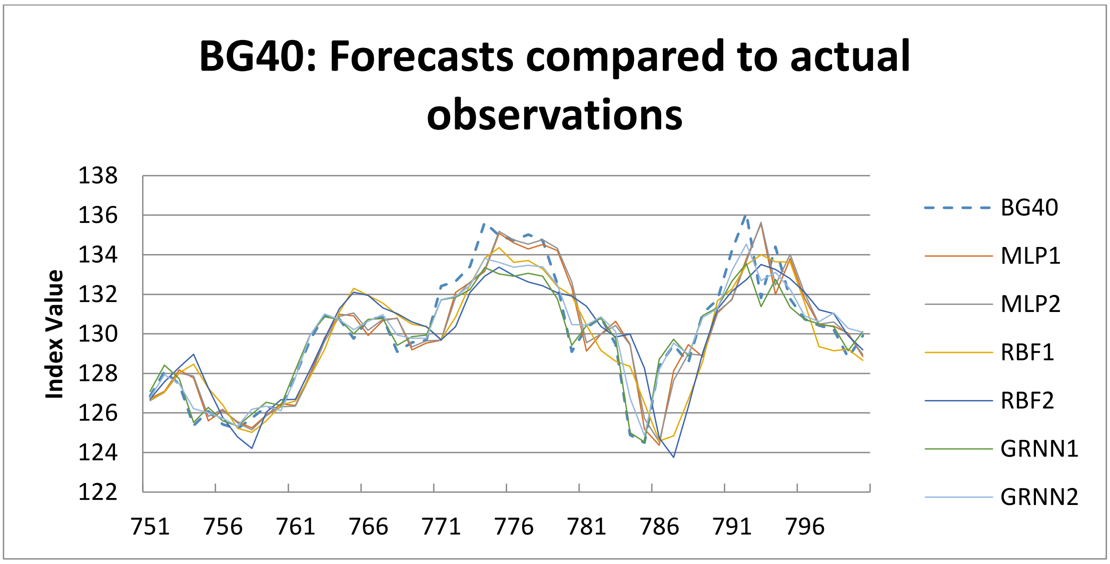

Figure 11: The values of the index and the predicted values by the three different models (two of each) for the last 50 days of the observed period.

The BG40 forecasting task again confirms the good results shown by neural networks with general regression. Generally, the errors in all three varieties increase over those produced by Sofix's forecasting, which is probably due to the higher volatility of the index and some other factors related to the selection of input data.

Based on the study of the possibility of forecasting the indices on the Bulgarian stock market with three of the numerous types of artificial neural networks, it can be concluded that this class of models in the field of artificial intelligence can successfully cope with the task of predicting future values related to with financial instruments based on time series financial information. It should be made clear at once that this does not in any way mean that they are capable of trading successfully on the financial markets, furthermore that the experiments carried out do not imply such an opportunity. On the other hand, they are certainly capable of supporting a more complex model aimed at real trading on the stock market, even in conditions such as those on the Bulgarian market.

Introducing a hybrid model built on neural networks and rule-based systems for trading on BSE - Sofia.

Capital markets in different parts of the world are very distinct due to the difference in many of its features. This work has already described in detail how big the differences between developed and underdeveloped markets are. This inevitably leads to equally large differences in the techniques applied in these markets, in order to gain an advantage over the other players and, of course, to make a profit.

Sofia is Sofix. It is based on the market capitalization of the ordinary shares included in it adjusted for the free float of each of them. The index covers 15 stocks that meet the general emission selection requirements as well as additional quantitative requirements. The issues included in Index calculation need to meet certain metrics including: a minimum trading period before inclusion, a minimum market value of their free float, a minimum number of shareholders holding the issue and a spread between buy and sell orders. Sofix is rebalanced twice a year, with trading and other information being taken after the end of the trading sessions on March 1st and September 1st, respectively. The final composition of the index is determined after all the qualifying emissions are ranked according to 4 quantitative indicators with all four weighted equally:

- biggest market value of their free float
- highest number of deals over the past six months
- highest median weekly turnover value over the past six months;
- the lowest average spread between buy and sell orders.

As of September 2016, the first exchange traded fund based on the Sofix Index started trading on the BSE-Sofia floor (Expat Bulgaria SOFIX UCITS ETF). As of April 2019 the fund’s total assets are slightly below BGN 25 million. The fund aims full physical replication of the Index and is listed on the London and Frankfurt stock exchanges. The purpose of the fund is to follow the performance of the index, with a minimum tracking error.

The Index is also used as a performance benchmark for the few foreign investment funds that invested in the local market as well as lots of professionally managed individual portfolios. All this makes the Index composition and rebalance extremely important. Which companies remain in its composition and which companies leave it and which are added to its calculation is of great importance for all market players. According to a observation conducted by the authors on the basis of the stock price change for the last 5 rebalances of Sofix, the average price change of the companies that fall out of its calculation is -8.35% for the period of one month preceding the final due date for determining its composition, and the average change in the prices of the companies added to its calculation is + 12.00% for the period of one month before the final date for determining its composition.

The author of this paper proposes a hybrid model based on artificial intelligence techniques: rule-based systems and neural networks, which predicts the composition of the Sofix index before this being made by other market players

The information required to feed the model is publicly available and is generated by the daily bulletin published by BSE – Sofia (Figure 1), containing all information on the daily trading of all issues listed on it as well as the free float and number of shareholders newsletter of each traded issue published by the Central Depository (Figure 2). The rules for determining and calculation of the composition of the index will be taken from the rules for calculating the indices of BSE-Sofia.

Rule-based systems use human experience-based knowledge to solve a variety of problems that typically require human intelligence to solve. This knowledge is shaped as rules or data in computer systems and programs. Depending on the problem being solved, the system can use this knowledge. For this purpose they are composed of the so-called. a decision-maker that evaluates which available knowledge to select and apply, as well as the knowledge modules and rules attached to it. This element produces action instructions, classifications, etc., and one of its advantages is that they are transparent and can be justified by the programmed rules and patterns in the model or combinations thereof. In this respect, rule-based systems are different from neural networks, which are often referred to as black boxes, the results of which are very difficult to explain.

Based on the numerous considered works in the field of neural network financial time series forecasting, it is proposed to use several network architectures, similar to the experiment described in the preceding points, to ensure the selection of a better model for this vital part of experiment. The network architectures chosen are those of the multilayer perceptron, neural networks with radial basis function, and neural networks with general regression with which the author has extensive experience. The neural network models will be used to predict the future development of the main 4 indicators on the basis of which the emission classification for the composition of Sofix is made. The idea is to make 2 types of forecasts: First one day ahead based on the initially available information for the entire remaining period and second, to generate dynamic data and include the results for each day forward produced by the networks in the training data .

A schematic representation of the proposed hybrid model built from neural networks and rule-based systems can be seen in Figure 12.

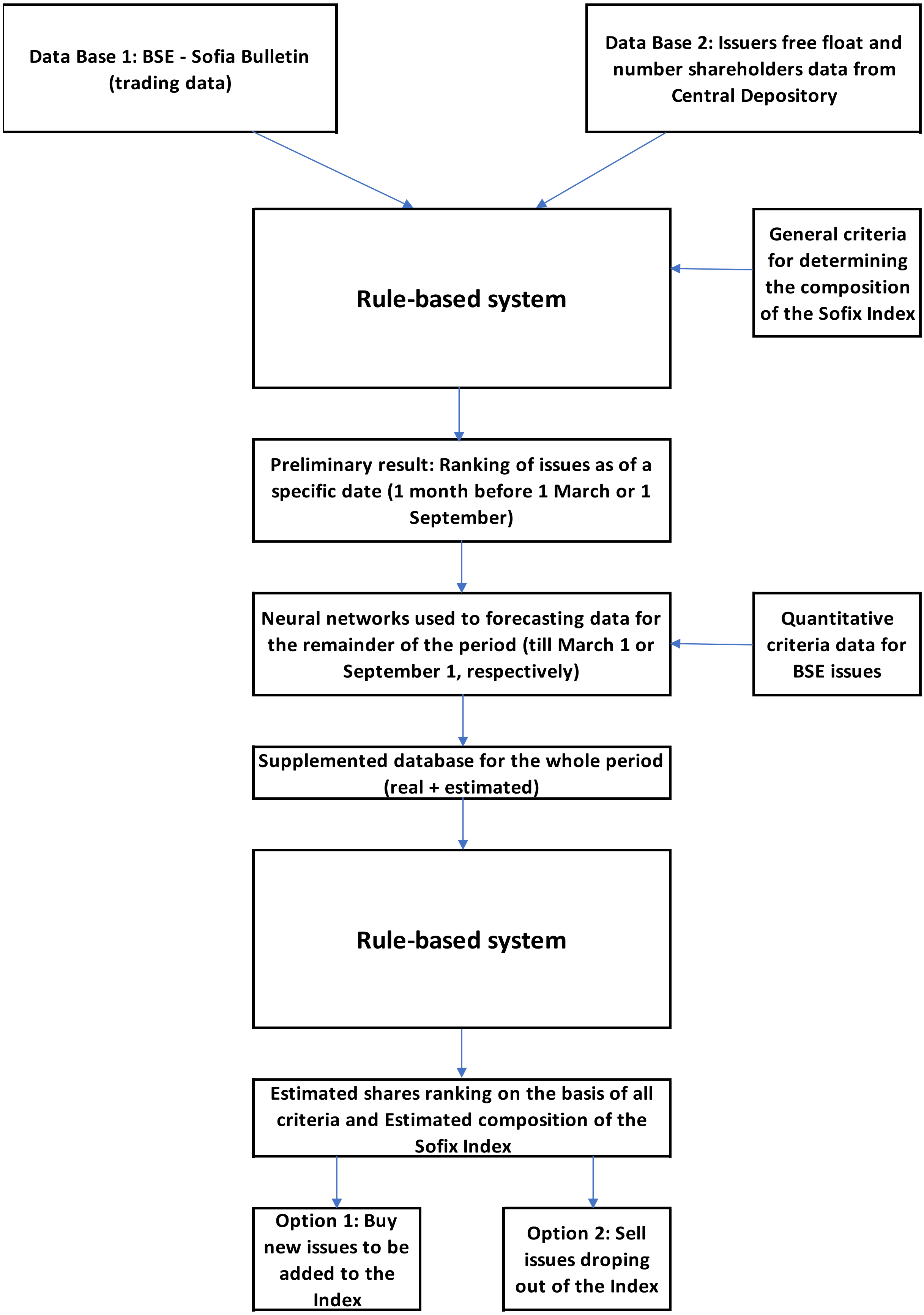

Figure 12: Rule-based system and neural network-based hybrid model

As can be seen in Figure 12, the proposed model shows, the module with rules-based systems is fed with information from the daily bulletins of Bulgarian Stock Exchange and the Central Depository, for a period of five months, This comprises knowledge base of the module. The rule module at this stage is made up of the general criteria for determining the Sofix index. The result produced, in the form of a list of qualifying shares is submitted to the neural networks module, which in turn performs forecasting of each of the four quantitative criteria on the basis of a database of these criteria for each individual share on the list. After the neural networks perform the forecasting, the resulting new database (built up of real and forecasted values) is fed to a rule-based system for compiling new company rankings. It determines the final forecast composition

CHAPTER IV: FUTURE WORK

Guidelines for future research can be drawn along two interconnected lines. The first, with regard to refinement of neural network based capital market forecasting models, through experiments with other types of networks, architectures, training techniques and various pre-processing of the information provided in the models in order to improve their forecasting capabilities. In this respect, the author sees potential in exploring the possibilities offered by relatively new techniques in the field of machine learning, such as support vector machines (SVMs) used for regression and classification. The second guideline is to further develop the proposed hybrid model of neural networks and systems based on rules for forecasting underdeveloped and low liquid capital markets aimed at complementing it by including an additional element to deal with the highly subjective quantitative element for evaluating the liquidity of the shares, namely the test of the spread between the buy and sell prices of fixed amounts, which is calculated after discretionary observations at the exchange operator's discretion during the exchange session. The author finds the technique in the field of Artificial Intelligence - fuzzy decision-making systems suitable for this purpose.

It could also work towards testing and applying these models on other capital markets, with similar performance in the Bulgarian capital, although this is related to many other conditions, such as specific trading conditions in similar markets, different rules for defining, maintaining and rebalancing indexes and more.

CHAPTER V: CONCLUSION - SUMMARY OF THE PRODUCED RESULTS WITH THE DECLARATION OF ORIGINALITY OF THE THESIS WORK

In conclusion, the following findings can be made on the basis of the experiments performed and the proposed hybrid model. In the first experiment with multilayer perceptron, averaging the submitted data in an illiquid market conditions results and manages to reduce the error in the validating data set significantly. There is a tendency for the error to continue to decrease as the averaging period is extended to 7 days. Then the error starts to increase again and the continuation of the averaging loses its meaning. Epochs of error propagation training are limited to 100, due to the time consuming nature of this type of training and the lack of better results when conducting experiments without setting a maximum number of cycles.

In particular, neural networks and multilayer perceptron perform well when forecasting on a time series basis for consolidation-phase market information and lose precision in trend conditions, especially in a downward trend.

Another observation from the averaged data experiment is the continued poorer performance of neural networks using the full set of 10 parameters for training, testing and validation.

The second experiment, conducted with three types of neural networks, produced some surprising results. First, the good performance shown by both neural networks with general regression and those with radial basis function. They seem to do much better in modeling the input data, managing to produce equally good results by isolating a different number of input parameters. For example, the second best network with general regression uses two neurons in its input layer, respectively, responsible for the target variable - the last index value of the day (close value) and the lowest value for the day, while the second best network with radial base function uses all 4 input parameters (except the above two and the opening value and lowest value for the day). Although the results of the multilayer perceptron were better than those achieved during the first experiment, they remained several times worse than those of the other models. This trend is visible not only in validating data, but also on the basis of error data during training and network testing.

During the second part of the second forecasting experiment (the prediction of the more volatile index), the good results again showing the neural networks with general regression were confirmed. Generally, the errors in all three varieties increase over those produced in the estimation of Sofix, which is probably due to the higher volatility of the BG40 index and some other factors related to the selection of input data. There were no major shifts in the ranking of the models compared to the previous experiment. To some extent, the use of 2 input parameters from the most successful model, which happens for the first time, can be considered unexpected, but in the previous case, the difference between the one that only relies on the requested variable and the second most successful was too small.

The results of multilayer perceptron are worse than those achieved during the Sofix test, but this, as mentioned above, is a general trend. They observe a convergence with the results of networks with a radial basis function, which can be considered as a positive factor. The results of the three models in terms of error during training and that of the test data retain the trends imposed by the baseline assessment.

In summary, it can be stated that the results of the study of the prediction capabilities of the three types of neural networks can be considered useful. Consistency in the results of neural networks with general regression makes them unprecedented winners in this kind of competition. Thus, they largely confirm their positive attestations, noted a priori by the experiments performed, as being faster and more successful, able to handle data containing deviations.

Very good and consistent performance is also reported for the model using a radial basis function. They show success in handling more than one input parameter to predict the target variable, which sets them apart.

The multilayer perceptron can be defined as the worst performing network in the prediction of both indexes, and its poor results are confirmed both by the amount of error during the validation data and by its training and testing. Although the size of the error it produces is not large and if we abstract from the results of other models, its estimates can be considered satisfactory and reliable, looking at things on a relative basis it is difficult to be complacent. It is also a fact that for the more volatile BG40 index data, the difference from other models, and in particular from those with a radial basis function, is greatly diminished.

Based on the study of the possibility of forecasting the indices on the Bulgarian stock market with three of the many types of artificial neural networks, it can be concluded that this class of models in the field of Artificial Intelligence can successfully cope with the task of predicting future values related to with financial instruments traded on the Bulgarian capital market based on time series financial information. It should be made clear at once that this does not in any way mean that they are capable of successfully trading on the capital markets as a standalone strategy for generating sales alerts, furthermore that the experiments conducted do not imply such an opportunity. On the other hand, they are certainly capable of supporting a more complex model aimed at real trading on the stock market, even in conditions such as those on the Bulgarian market.

As a logical continuation of the conclusion from the first two experiments comes the author's proposed hybrid model of neural networks and rule-based systems for trading on the BSE - Sofia. The model is considered to be operational and applicable in the context of a low liquidity and underdeveloped capital market. It is promising and has huge potential to be used both to experiment with historical data for academic purposes, such as validating or rejecting the theory of an effective market, and to support a real trading strategy in line with the low liquidity, the absence of rich trading instruments (the practical impossibility of short selling, margin purchases and the lack of derivative instruments) and the lack of many quality issuers that can be characterized by and the Bulgarian stock market.

As a result of the research presented in this dissertation, certain scientific and applied results have been achieved:

The possibilities for forecasting capital markets with neural networks have been investigated. The formation, development and ability of neural networks to forecast the capital market was investigated. Particular attention has been paid to the evolution and development of this endeavor globally over the years, as well as to the various approaches that have been offered and used.

A study was conducted on the state of play in the forecasting of the neural network capital markets, with a review of a large number of books, articles and publications on the topic, of which more than 50 papers were reviewed and analyzed, from which interesting conclusions were drawn for preferred from the authors models, architectures and methods of training neural networks. Also, a summary of the capital markets, stock indices and types of inputs predicted and used by the models is made, as well as an examination of the use of other techniques in the Artificial Intelligence field for capital market forecasting.

As part of the study on the Bulgarian stock market, which is the subject of forecasting, a presentation was made on the Bulgarian regulated capital market in the face of BSE - Sofia and the main qualitative and quantitative differences between it and the developed exchanges around the world were determined, on the basis of which defines as not well developed.

A hybrid model of neural networks and systems based on rules for forecasting a undeveloped and low liquid capital market such as that in Bulgaria is proposed. The author finds this model workable and extremely promising, with enormous potential to put into use both those who wish to experiment on past historical data for academic purposes, such as exposing the “Effective Market Hypothesis” and supporting a real-world one. trading strategy tailored to low liquidity conditions, lack of rich trading tools.

The experiments carried out with the created hybrid model in the real conditions on the Bulgarian capital market, on the basis of data from past periods, confirm its capabilities.

The summarized results of the experiments performed and the results obtained can be summarized as follows. On the basis of the acquired knowledge from the research of the previous attempts in the field, an attempt was made to choose the appropriate neural network structure for the purposes of forecasting a low liquid financial market such as the Bulgarian one. Taking this particular feature of the domestic capital market into account, an appropriate selection of input parameters was made for the selected forecasting model so that it could make accurate predictions. The preliminary preparation and processing of the data was successfully carried out, resulting in significant improvements to the outputs produced by the models. The results clearly demonstrate the effectiveness of performing value transformations in changes and averaging of input data. This manages to reduce the validation error significantly. There is also a dependence of the magnitude of the calculated error of all the models used on the so-called market situation. It has been found that for large amplitudes in the value of the stock indices considered, known as the term volatility, the tendency of which is to increase in periods when the market is in a trend and especially a lot in a downward trend when stock prices go through a long period of decrease, artificial neural networks lose accuracy in their forecasts. This is especially pronounced in a downward trend. There has been a tendency for neural networks to perform well when forecasting on the basis of time series of financial information for consolidation phase markets. Experiments for forecasting BSE-Sofia indices were conducted with 3 types of neural networks and the results obtained are found to be good and extremely interesting. A curious tendency has been found, namely, that the most commonly used type of neural network, according to the scientific works described by the author, is that the multilayer perceptron does not perform as successfully as the other two varieties - the neural network with radial basis function and the most the precisely predicted neural network with general regression. They seem to be much better at modeling input data, managing to produce equally good results by isolating a different number of input parameters. For example, one of the best performing general regression networks uses two neurons in the input layer corresponding to the target variable, the latest value of the forecast index of the day and the lowest value of the day, while another very successful model of a radial basis network uses all 4 input parameters provided to it. The author finds the attempt to call into question the validity of the "Effective Market Hypothesis" to be successful, with the low levels of model error making it suggest that new and more in-depth research in the field is likely to further improve the results.

CHAPTER VI: PUBLICATIONS ON THE SUBJECT OF THESIS AND CIATATIONS

Shahpazov V., Doukovska L., Karastoyanov D., Artificial Intelligence Neural Networks Applications in Forecasting Financial Markets and Stock Prices. Proc. of the International Symposium on Business Modeling and Software Design – BMSD'14 Luxemburg, Grand Duchy of Luxemburg, SCITEPRESS – Science and Technology Publications, 2014, ISBN: 978-989-758-032-1, DOI:10.5220/0005427202820288, 282 – 288.

Shahpazov V., Doukovska, Forecasting financial markets with artificial intelligence. Proc. of the International Workshop on Advanced Control and Optimisation: Step Ahead ACOSA'14 Bankya, Bulgaria, Prof. Marin Drinov Publishing House, 2014, ISSN:1314-4634. 67-74.

Shahpazov V., Velev V., Doukovska L., Forecasting Price Movement of Sofix Index on the Bulgarian Stock Exchange – Sofia Using an Artificial Neural Network Model. Proc. of the International Symposium on Business Modeling and Software Design – BMSD'13, Noordwijkerhout, The Netherlands, SCITEPRESS – Science and Technology Publications, 2013, ISBN:978-989-8565-56-3, DOI:10.5220/0005427202820288, 298-303.

Shahpazov V., Velev V., Doukovska L., "Design and Application of Artificial Neural Networks for Predicting the Values of Indexes on the Bulgarian Stock Market. Proc of the Signal Processing Symposium – SPS'13, Jachranka Village, Poland, IEEEXplore, 2013, ISBN:978-1-4673-6319-8-13, CD Proc.

Шахпазов В., Високочестотната търговия. Списание Техносфера ISSN 1313-3861 Брой 4 (34)/2016 г.

Shahpazov V.L., Popchev I., A Neural Network and Rule-based Systems Hybrid Model for Forecasting the Capital Market, Международна Научна Конференция „Изкуствен Интелект и Е-Лидерство ", Пловдив, Октомври 2019 г. (под печат)

Citations:

1. Samit Bahnja, Abhishek Das, Impact of Data Normalization on Deep Neural Network for Time Series Forecasting, 2018

2. Cavalcante R.C., Brasiliero, R.C., Souza V.L., Nobrega J.P., Oliveira A.L., Computational Intelligence and Financial Markets: A Survey and Future Directions. Expert Systems with Applications, 55, pp. 194-211, 2016.

3. Gabriel F.C. Campos, Rodrigo A. Igawa, Jos'e L. Seixas Jr., Alex M.G. de Almeida, Rodrigo Capobianco Guidoy, Sylvio Barbon Jr., Supervised Approach for Indication of Contrast Enhancement in Application of Image Segmentation, Proc. Of the Eighth International Conference on Advances in Multimedia, ISBN 978-1-61208-452-7, pp. 12-18, 2016.

4. Tarun Dash, Vinayak Jaiswal, Anoosha Sagar, Gaurav Vazirani, Nupur Giri, Analysis of associativity among mirror neurons for financial profiling, Proc. Of the Second

International Conference on Cognitive Computing and Information (CCIP), DOI 10.1109/CCIP.2016.7802869, 2016.

5. Tarun Dash, Vinayak Jaiswal, Anoosha Sagar, Gaurav Vazirani, Nupur Giri, Analysis of associativity among mirror neurons for financial profiling: A proposal, Proc. of the International Conference on Computing Communication Control and automation (ICCUBEA), DOI 10.1109/ ICCUBEA.2016.780034, 2016.

6. Tukur U.M., S.M. Shamsuddin, Radial Basis Function Neural Network Learning with Modified Backpropagation Algorithm. Indonesian Journal of Electrical Engeneering and Computer Science, vol. 13, 2, pp. 369-378, 2015

All citations refer to the following:

Shahpazov V., V. Velev, L. Doukovska - Design and Application of Artificial Neural Networks for Predicting the Values of Indexes on the Bulgarian Stock Market, Proc. of the Signal Processing Symposium – SPS’13, Jachranka Village, Poland, CD, ISBN 978-1-4673-6319-8-13- 2013 IEEE, 2013.

CHAPTER VII: SCIENTIFIC PROJECTS PARTICIPATIONS

- IICT Project - Intelligent Functional Diagnosis of Complex Systems and Investigation of Uncertainty and Risk Structures (2011-2014).

- EEMPAM Project - “Building and Development of Young Highly Qualified Researchers for Effective Application of Biomedical Research to Improve Quality of Life”, BG051PO001-3.3.06-0048 / 04.10.2012