ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20190503164349
Subject review

Time Series Forecasting of the Austrian Traded Index (ATX) Using Artificial Neural Network Model

Marko MARTINOVIĆ*, Anica HUNJET, Ioan TURCIN

Abstract: This paper analyses the Austrian Traded Index (ATX) of the Vienna Stock Exchange (Wiener Börse) in the period from 2009 to 2017, using the method of the artificial neural network (ANN). Sampling data are taken from the web page of the Wiener Börse and filtered on weekly basis to comply with weekly seasonality in eight years range. The aim is to construct several AAN models that meet certain criteria and evaluate them on the holdout subsample. Furthermore, the goal is to find the best model that can predict new upcoming yet unseen data with high accuracy. A data frame for testing forecasting performance is one month, a quartile, a half year, and one year period for which last year of the data sample is retained (August, 2016- August 2017). Using various criteria and different parameters, the total of thirty networks were built and tested and top five networks were analysed in more details. Results confirm high accuracy of using method of artificial neural networks, which is consistent to studies conducted on similar cases. Correlation of top three selected networks by validation subsample is over 0,9. The mean absolute percentage errors (MAPE) for the best selected network are 1,76% (month); 2,11% (quartile); 2,21% (half-year); 2,13% (year). Once again, ANN method has proven to be a powerful forecasting tool.

Keywords: artificial neural networks; ATX; forecasting; prediction; stock market; time series analyses; Wiener Börse

1 INTRODUCTION

The artificial neural network (ANN) is a popular non-linear concept often used in time series forecasting. Its popularity has gained attention especially in the last several decades. There are many other methods used for time series forecasting with different approaches, and the one most popular among them is the Autoregressive Integrated Moving Average (ARIMA) [1] linear method. In this paper the artificial neural network approach is proposed as a major method. Although combining different methods increases accuracy [2], especially putting together non-linear and linear models like artificial neural networks and linear models like the ARIMA, this paper is focused solely on the ANNs and its combinations, though leaving opportunity to benchmark and compare the same method with different variations of characteristics, parameters and configurations.

The importance of knowing future behaviour in business world is well known, especially in financial sector. Predicting future actions is the basis for making sound business decisions, particularly if they are based on more than one indicator that can elevate chance of finding right course of action. Forecasting methods are often employed in solving problems in various sectors like energy sector (energy demand), ecology (prediction of air pollution), sales and product demand, marketing, predicting business failure, bankruptcy, GDP growth, stock markets, exchange rates, traffic, science etc. Thus, accuracy of chosen prediction model makes huge impact on the economy of scale and represents one of important challenges in modern science. [3].

Using historical datasets, various models are proposed for forecasting near to future behaviour. The challenge is to find the best model that can fit underlying data pattern (but not over fit), making minimal discrepancies from original series along with making a minimal validation error. The basic assumption is that choosing the best model on sample data will represent the best solution for the future data set. But, that is not always correct because of various underlying factors like variation in the sample, data structure changes and model uncertainty [4].

There are two important approaches in choosing prediction method, casual and non-casual. The casual methods work on behalf of input explanatory variables that are deemed as important impact or for predicted variable, while time series methods capturing underlying pattern using solely datasets of the previous observations, make it very simple to analyse and study. They are not restricted to need to know underlying nature of neither generated data nor important factors that can influence the outcome. That may represent a problem in case of the causal models where selecting input variables is often resulting in guessing and subjective choice. This paper is focused on the time series model.

Second major concern is about assuming the underlying data generating process. Generally, it can be linear or nonlinear. Despite the fact that many real-world problems are inherently nonlinear, linear based methods dominated for decades. Some of the reasons are its simplicity of assuming and deploying models. The other reason may lay in the fact that new emerging nonlinear methods like the ANNs are employed in relatively recent history, especially at the end of the 20th century fostered with a rapid development of the information technology.

Some of the representatives of linear models are the ARIMA model, the moving average (MA), the exponential smoothing (ES) model, multiple linear regression and the autoregressive model (AR). One of the most popular linear models is the autoregressive integrated moving average (ARIMA) model, known as the Box-Jenkins [1]: It combines the autoregressive (AR) and the moving average (MA) models, which are special cases of the ARIMA. Additionally, some types of the exponentially smoothing models can be represented using the ARIMA.

On the other hand, the nonlinear models as extension of previous efforts, encompass the threshold autoregressive model (TAR), bilinear model, and the autoregressive conditional heteroscedastic model (ARCH), etc. [5]. These models are tailored for specific nonlinear data patterns, thus have limitation in generalisation of nonlinearity. In addition, many of nonlinear models are parametric; it means that a certain assumption has to be made prior to implementing the model. Hereof, there are prone to certain discrepancies from optimal solution if real data characteristics do not match model assumptions.

One approach that can overcome those issues is the artificial neural network model, which is more flexible in assumptions, adaption and approximation. The ANNs represent the generalized nonparametric and nonlinear model and they will be used in further analysis to match and forecast ATX index price movement.

With regard to a previous research on ATX index, Haefke C. and Helmenstein C. used cointegration and Grager-causality analysis to construct linear models and feedforward artificial neural network to forecast Austrian Initial Public Offerings (IPOX$_{\text{ATX}}$) based on relationship with Austrian Stock Market Index. The constructed model has shown better result than using classical approach such as moving average [6].

S. Chittenkopf et al. [7] used ANN models like mixture density network (MDN) to compare the performance of the volatility models to standard models. The results show the importance of long-term memory of the model for trading strategy.

In the paper Index forecasting and model selection [8], authors exploit the informational differences using various stock market index construction principles. They used one day ahead index forecast for buy or sell signal. Model is tested on Austrian Traded Index using stocks included in it. Prediction was made using standard econometric methods and feed forward neural networks.

2 ARTIFICIAL NEURAL NETWORK METHODOLOGY

An artificial neural network is a relatively new concept with regard to classical statistic methods [9] and it has been widely used in the business and science. Artificial neural networks represent nonlinear approach, which does not require any strict prior assumption of underlying data form nor of the kind of linearity of analysed sample. Regardless of its inherent nonlinear nature, the ANN can also successfully cope with linear problems. ANNs have very accurate approximation possibilities that can fit a broad set of problems.

Forecast models can be characterised by their variance and a bias. The variance represents how much model is able to adapt to dataset generated from the same static data source, or how much it is prone to changing due to change of variance. It means, how much model is stable and reliable. On the other side, the bias represents system error incorporated by choosing a certain pre-specified model. This is very important in the case of parametric models, where assumptions about underlying data process have to be made before implementing the model. Other issue is, what happened if the data structure changed over time. It means, how much model is flexible. For instance, if the data generated process changes over time, a static pre-specified parametric model will have higher bias. Preferred model characteristics are low bias and low variance, but those two are typically exclusive. With more stable model it becomes more static, and with more flexible model, the variance is arising.

The ANN represents a flexible and robust model that can easily adapt to various data patterns. But over designing (like big number of hidden nodes, and recurrent connections) can impose certain disadvantages. Overfitting is common in complex systems, where too many parameters are included. Therefore, the variance of data sample is embedded in the model design, which is not preferred. So, the best practice is to find a balance between the model bias and the model variance.

In addition, a neural network is an implicit model, it means that it cannot be nicely and explicitly explained and it behaves more like a "black box" driven by input data. An advantage of using such approach is the possibilities for capturing nonlinearity in data patterns without assumption of "nature" of the data; also, it has universal functional approximation possibilities.

The idea for ANNs comes from analogy with human brain, biological neurons and their wiring.

Though human brain is very complex, the artificial neural networks are much simpler. The main unit-neuron (perceptron) is a simple computing unit which has major function to proceed input signal into processed output. The network is organised in the interconnected layers.

Neural networks differ by their characteristics and design. There are many types of ANNs today, but ones of the most used for time series prediction problems are the feedforward networks. In this paper, three-layer feedforward fully connected neural networks with one output neuron and backpropagation learning algorithm [10] will be used. Feedforward networks (multilayer perceptron) process signal in just one direction, from input to output, without any back connection. They are very simple in design (especially in three-layer case) but very powerful in solving nonlinear problems.

The three-layer feedforward network for time series analyses uses previous observations as input points. Usually, one output neuron is used as one-step-ahead forecast. The number of output neurons can vary for multi-step-ahead approach, where more than one output neuron is used. First layer of the network acts as input layer, the second is the layer of hidden nodes (hidden layer), and the third layer represents output layer. Layers are usually full-connected, which means that each neuron from previous layer has a connection with all nodes in the current layer. Neuron calculates trivial function as the sum of multiple inputs and connection weights, and afterwards, that value is transformed by nonlinear activation function. Usually, the activation function for multilayer perceptron is a logit or sigmoid-function as special case Eq. (2).

Three-layer perceptron can be represented with mathematical expression Eq. (1).

$$y_t = w_0 + \sum_{j=1}^{q} w_j \varphi\left(w_{0j} + \sum_{i=1}^{p} w_{ij} y_{t-1} \right) + e_t \tag{1}$$

where:

$$\varphi(z) = \frac{1}{1+e^{-z}} \tag{2}$$

φ - is activation function, p - number of input neurons, q - number of hidden neurons, w - connection weights.

According to specific case, these parameters have to be chosen wisely. While connection weights are results of the learning process, all others parameters are arbitrary. For the activation or transition function the logit function is often used. Number of input neurons in the input layer represents the number of lagged observations (previous observations) and that parameter plays crucial role in the network design. The window of previous observations catches autocorrelation between adjacent data points. Another important parameter in the network design is number of hidden nodes. The hidden nodes are laid in the middle of the network and act as a network engine. Choosing the right number of hidden nodes can significantly improve network performance. If this number is too small, the network lacks the complexity for building sound model. However, if the number is too big, the network faces a problem of overfitting.

In designing the ANNs, there is no standard procedure in systematic and uniform approach, which can estimate exact values of network parameters that should be used. For time-series analysis prediction model, one output, several input and several hidden nodes are used, usually under a few dozen or more common under ten. Often, experimenting on the data sample is a guide for arbitrating parameters, which can often lead to a suboptimal solution.

Other parameters that should be taken in consideration are sample size and data splitting. It is commonly understood that the larger sample has advantage in building a good model, which makes sense, especially because of the need for further data division on training, validation and evaluation subsets. More observations mean more chances for better approximation of the underlying data structure. But on the other side, large data sets can suffer from unexpected changes of underlying data generated process, which can lead into a poor model design. Typical sample size is over 100 observations.

The data splitting is the next step toward designing the network. The sample can be divided into two subsets, in-sample data and out-sample data. The in-sample data is used for network training, while the out-sample data represent holdout-unseen data used for model evaluation. Splitting in those two sets is common in designing various linear and nonlinear models, but in the case of ANN, additional division is needed. The in-sample data is further split into a training set and a validation set. The training set is used for building networks, while the validation subsample has the purpose of selecting the best ones.

The out of sample or holdout sample represents real unseen data that were not included in the network building process. Purpose of the out-of-sample is to evaluate or find out how much new model fits real data, or how much learning process is successfully accomplished.

The core of the learning process is gained through the training sample; selection is made by the validation sample and finally, model evaluation is done by the out of sample data. One more arbitrary parameter is how to split those sample sets. Although there is no exact standard framework for that, most literature suggests that the majority of the sample is dedicated for learning purpose, about 70% to 90%, and the rest is reserved for the model evaluation. Determining an appropriate ratio is often the product of experimentation on the original sample. For the ANNs, further division of the in-sample data is needed.

After determination of right parameters and prior to building a new model, input data has to be additionally processed. A neural network is only good as its input data. An original sample has to be processed or transformed into more suitable form, including data normalisation, deseasonalization and detrending [4]. A neural network is especially sensitive on data variation, because autoregressive learning process totally relies on input data, thus more uniform input is advisable.

Network connection weights are variables that have to be gained from the training process. There are several training methods, but most widely used for multilayer perceptron are supervised learning methods called backpropagation. This algorithm gradually changes (descent) connection weights trying to find optimal solution, in other words minimise error between real and computed data values. Often, sum of squares is used for error function.

The error function can be seen as surface in the $N + 1$ dimensional space, where the N represents the number of weights. Training algorithm is trying to find the minimal point on that surface. The gradient descent algorithms start from random point on error surface looking for direction with the largest incline and make a step toward that direction. Iterative process leads to minimal or sub minimal point. Thus, number of training process, with different random start point and network parameters are needed to find the global minimum. There are other efficient nonlinear optimisation methods like BFGS-Broyden-Fletcher-Goldfarb-Shanno or Scaled Conjugate Gradient algorithms which are shown as more efficient with less iteration and more precise, but they demand more computation power for the training process.

3 THE MODEL AND DATA

Although it is known that in most cases there is no solution "best fit to all" that one method is superior over all possible scenarios, this paper is focused on the ANN as a sole method and on its combinations and variations.

3.1 The Data

Research study traces the Austrian Traded Index (ATX) of the Vienna Stock Exchange (Wiener Börse). Data are taken from the Wiener Börse official web page dating from 01.02.1997 up to 08.08.2017.

The Austrian Traded Index (ATX) is a price index that contains blue chip, prime market Austrian enterprises listed on the Wiener Börse. Index is composed out of 20 constitutes with the maximum weight share of 20% for one company. Price is updated in the real time and the index is reviewed two times per year (March and September). The main criteria for selecting companies in the index are capitalisation and exchange trading volume. Per review, no more than three components may be changed. The index was launched on January 2, 1991, with base value of 1000. Today, top 5 shares make up more than 75% value of the index, with dominating banking sector and oil/gas industry.

The original downloaded data of the ATX index prices are on daily basis, which means, for every open trade day, there is one record with the following attributes: date, open price, high price, low price and price change (in percentages). For the purpose of this paper, data were transformed in the weekly time frame. In the literature, it is more common to refer to the weekly prices [11]. In addition, the weekly values are more suitable for spotting the seasonality which is then used in further data processing.

For constructing a new model, only the records from the date 01.02.2009 onwards were considered, although older observations are available. The reason is to avoid effects of the global economic crisis, which significantly alters data structure in the period from the end of November 2007 to the beginning of February 2009. Total sample size of weekly prices counts 450 observations.

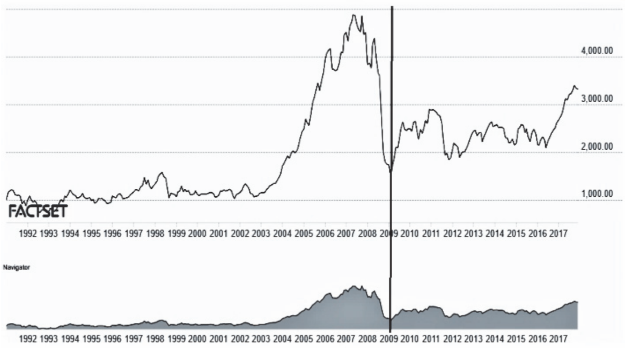

Figure 1 ATX price from 1991 to 2017

The data on the left from the vertical line in Fig. 1 (point 2009) were ignored. On the right from the vertical line are the models sample data. From the figure it can be seen the surge and huge dropdown of the price as a result of the global economic crisis.

Such surges driven by the underlying changes of generated data process can raise an issue of overfitting, especially in the case of the ANNs.

Although the ANN is a robust model that can easily adapt to various situations without prior processing, more uniform structure is desirable for training. That can be achieved with data pre-processed, making prediction models more efficient and accurate.

Data pre-processing for the ANN can include data normalisation, deseasonalization (seasonal adjustment), detrending and transformation. Data transformation translates original values into the translated area (image) via transformation function. For forecasting purpose, often natural logarithmic function is used. The seasonal adjustment removes the seasonal pattern from the time series, makes it more applicable and convenient for building a model. Detrending removes the trend from the original series leaving clear data pattern. In this research, the seasonal adjustment and detrending were used.

There are different statistical methods and solutions developed by various institutions like the United States Census Bureau (X-ARIMA), Bank of Spain (TRAMO/SEATS) and others. The time series can be shown as a sum of the components Eq. (3).

In this paper, simple time series additive decomposition is used in Microsoft Excel tool [12, 13].

$$Y_t = T_t + S_t + I_t \tag{3}$$

Chosen period for seasonal adjustment is a week of the year, this is because data are presented on the weekly basis. Firstly, the ratio of moving averages (RMA) is calculated using the difference between original series and the central moving averages (52 values). Afterwards, the seasonal indexes were calculated as normalised averages of the RMA for each week number and for each year in the relevant sample the adjusted seasonal indexes (ASI) are depicted in Fig. 2. From the figure (for example) can be seen that the difference between the week numbers 8 and 39 is more than 10%. Also, the first half of the year in average has higher values than the other half.

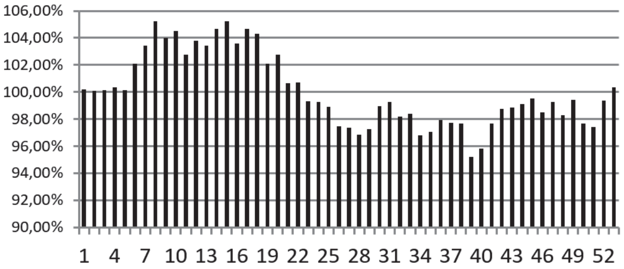

Figure 2 Adjusted seasonal indexes week no. 1-52

Using the adjusted seasonal indexes and the original data series the seasonal adjusted series (ASS) can be calculated. After determining S_t component from the equation Eq. (3), the trend has to be removed (Fig. 3).

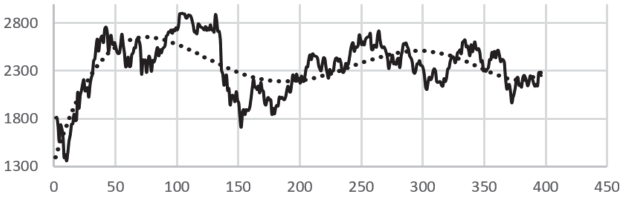

Figure 3 Adjusted seasonal series (ASS) with the trend line

Fig. 3 includes values that are engaged in the learning process, thus the last 54 observations were omitted. Despite the seasonal adjustments, some regularity still exists. Rather than using simple linear regression to make data more uniform, a polynomial regression is used. On the graph, dotted line represents the polynomial trend line of the fifth order which fits data with coefficient of determination R^2 = 0,52. Second alternative was simple linear trend line with R^2 = 0,02, and slope of 0,12, but it is not taken in consideration. The polynomial trend line coefficients are shown in Tab. 1.

Table 1 Polynomial trend line coefficients

x5	x4	x3	x2	x1	b
7,01306E−09	−8,35835E−06	0,0036208	−0,6871499	53,274462	1230,0035

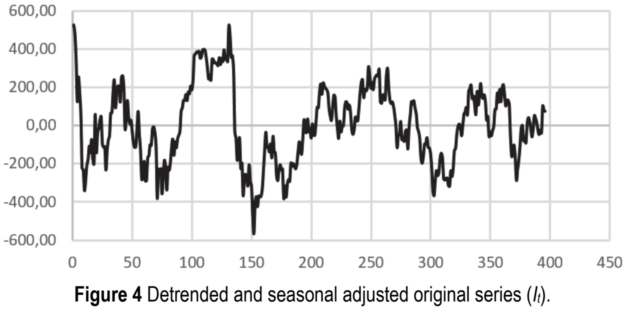

Figure 4 Detrended and seasonal adjusted original series (I_t).

From the formula Eq. (3), when trend T and seasonality S components are removed, what remains is I_t - uncaptured irregularity and error function (Fig. 4). That represents pre-processed data as input for creating the new models. In the figure, zero value (on x-axes) denotes start date of the sample (2.1.2009), and 450 denotes end date 8.8.2017. The last point in the graph is 396 (29.07.2016) and it is the last point in the subsample used for network building process.

3.2 The Model

An artificial neural network model applied in this research is the three-layer perceptron, or three-layer feedforward neural network. Building and training were conducted using the StatSoft STATISTICA Automated Neural Network (SANN) data mining method. The automated procedure combines various parameters in the manner to find the best solution. The options that can be chosen are number range of hidden nodes, activation function (both for hidden and output layer), number of input nodes and number of networks to train and retain. The options set for this particular model are: number of hidden nodes is [2-12], number of input nodes 5, output node 1, activation functions are one from the set {identity, logistic, tanh and exponential}, number of networks to train and retain is 30; error function is sum of squares.

Number of input nodes is one of the most important parameters of network design. That number represents how many lagged observations model will consider in chunk. It catches data pattern by moving the window of input nodes through all training sample. Two variants were in consideration, the first is for the period of approximately one month (5 input nodes), and the other is for the period of one year (52 input nodes). Although both variants have certain advantages, in the literature more common is the use of a small number, so number 5 is finally applied. One output neuron is set. It can be interpreted as one ahead forecasting and it is also very common in use today.

Time series data are split in two major subsets, on the in-sample data (depicted in Fig. 4), and the out-of-sample or holdout sample. The holdout sample retains price observations which are not used in network building process, and represent unseen future values. Size of the in-sample data is 396, and size of the holdout sample is 52. Every observation represents a weekly price taken as last close price on the last trade day in the week. Usually it is Friday, but it can fall on Thursday or even on Wednesday in special cases (holidays). The idea is to keep unseen observations for a period of one year, which encompasses the last 52 weekly prices (5.8.2016 - 4.8.2017) in the set. Using these observations, evaluation of the selected model will be made. Additionally, the in-sample data is further split into training and validation subsets. Using the training subsample network is built and using the validation subsample, the best networks are selected. The validation subsample encompasses last 52 observations of the in-sample data (7.8.2015 - 29.7.2016). The training subsample contains all other values. Distribution of subsamples are: {344/52/52} out of total 448 units, or in percentage {77%/11,5%/11,5%}, which is a typical ratio found in the literature.

The BFGS-Broyden-Fletcher-Goldfarb-Shanno is used as training algorithm-an iterative method of solving non-linear optimisation problems. It is also called Quasi-Newton and it is a second order training algorithm with fast convergence. It demands more memory and processor power than basic gradient descent training algorithm like backpropagation method. This is due to calculating and storing Hessian Matrix.

In the building process, total of thirty networks were constructed and top five networks were analysed in more detail. Results (graphs, performance and errors) are presented in the time frames of one month, quarter, half-year and on one-year basis.

4 RESULTS

After running the automated neural network procedure, new networks were built with the following average characteristics (Tab. 2 and Tab. 3).

Table 2 Average performance of 30, 5 and 1 top selected networks

	Training performance	Validation performance	Evaluation performance
Top 1	0,9427	0,8620	0,9156
AVR5	0,9415	0,8593	0,9074
AVR30	0,9383	0,8545	0,8990

Table 3 Average SOS error of 30, 5 and 1 top selected networks

	Training error	Validation error	Evaluation Error
Top 1	2370,71	1707,10	2664,35
AVR5	2419,60	1724,37	4155,00
AVR30	2556,50	1779,23	6544,75

Performance in the SANN tool represents correlation between real target data (model input) and predictions made by model (output). In theory, correlation coefficient is the value between −1 and −1, where 1 represents perfect match. For performance on the training set, correlation close or equal to 1 is not always a good sign. It may indicate the problem of overfitting or incorporating variance and random noise into the model which leads to poorer generalisation capabilities.

Tab. 3 shows aggregate performance of all trained networks, top 5 and the best one. First row refers to the best network, second row stands for averages of top 5 networks, and the last row calculates averages of all 30 built networks. The best selected network is not the best network (among those 30), but it is the best selected network by validation performance.

It can be seen that the performance on training subsample has slightly higher values than on the validation or evaluation performance. The average performance on the validation subsample shows the lowest value, but the evaluation results are again higher. It means that constructed models in general have better performance scores on the unseen data than on the in-sample data. Horizontally (in the third row Tab. 3), the difference in the performance averages of all trained networks between groups {training/validation, training/evaluation} is {8,9%, 4,2%}. In the case of the best selected network (first row Tab. 3) those numbers are {8,5%; 2,9%}. The difference of 2,9% indicates that efficiency of the holdout sample for the best selected network is just below 3% lower than the efficiency demonstrated on the training set. Also, there is no major deviation between the values in the same group. For example, in the validation column, the difference between the best selected network and the averages of all trained networks is 0,87%. For the case of the evaluation set that difference is 1,8% and for the training set 0,47%.

The Sum of Squares (SOS) is used as an error function and the results in Tab. 3 indicate the lowest average error of the validation sample, while the evaluation sample has the biggest error value. Again, the differences of the averages of all networks (30), top 5 networks and the top performance network within same groups (training, validation, evaluation) are 7,3%, 4,1%, and 59,3%. The biggest difference in the error value is in the evaluation group.

It indicates that the error distribution of the evaluation group is widely dispersed, despite very narrow performance results (correlation). The SOS error with similar performance (correlation) can be imagined as a distance between parallel lines with similar patterns, dislocated from each other. With the greater distance the error is bigger. The difference of the average error between subsamples can be due to the difference of data characteristics included in the samples.

Table 4 Data statistics of the sample

Datasets	ATX' (Target)
Minimum (Train)	−565,564
Maximum (Train)	526,259
Mean (Train)	−1,622
Standard deviation (Train)	212,288
Minimum (Validation)	−286,967
Maximum (Validation)	214,162
Mean (Validation)	10,733
Standard deviation (Validation)	112,595
Minimum (Evaulation)	96,570
Maximum (Evaulation)	755,578
Mean (Evaulation)	434,396
Standard deviation (Evaulation)	459,628

That can be noticed from Tab. 4, and in Fig. 5, where the range between maximum and minimum values in the validation set is the smallest, even smaller than in the training sample despite the fact that performance on the training sample shows better results. Also, from Fig. 5 can be seen that the training set has the shape closest to normal distribution. It is logical, because the train sample has over 7 times more observations than the remaining two sets.

The biggest deviation is in the evaluation sample. In absolute values, minimal deviation is in the validation group, but in percentages (deviation relatively to the range), the minimum is in the training set. Standard deviations (in the percentages) of the training, validation and evaluation groups relatively to range (max-min) are: {19,4%; 22,46%; 53,9%}. This unequal distribution can impose certain problems, thus more uniform data statistics across the groups are preferable. Strong deviation in the validation group is due to unexpected surge of the ATX price in the period of last observed year. Despite that fact, the proposed ANN models fit the problem very well.

The thirty trained models and original data series are plotted in Fig. 6. The solid arrow points to the group of networks (cluster) where the original series are located. Basically, it can be noticed that there are two major network clusters and one network that stands out from the rest. From Fig. 6 groups of networks can be easily noticed; the lowest cluster is marked with dotted arrow, the upper cluster (with solid line) and the anomaly network, above all of them.

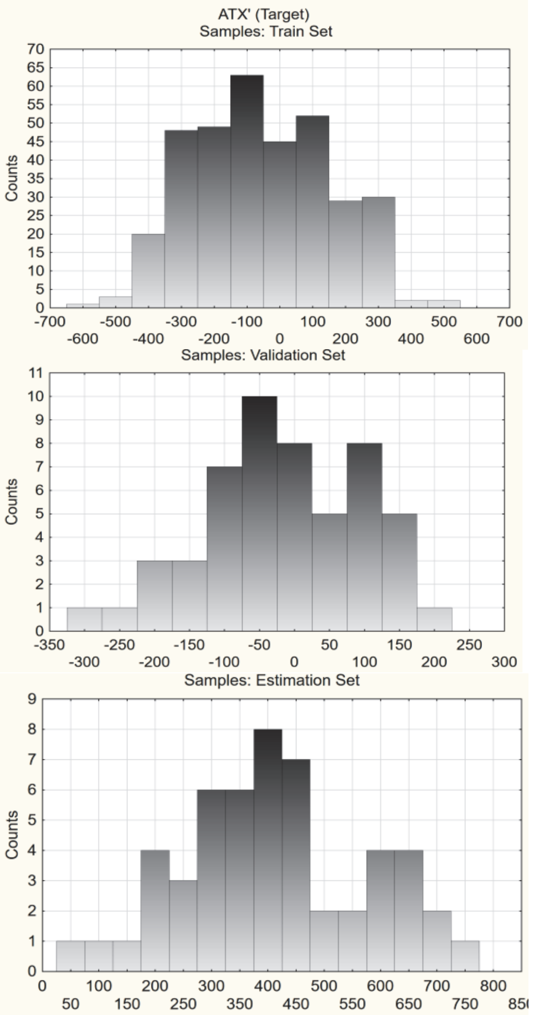

Figure 5 Histogram of training, validation and the evaluation samples

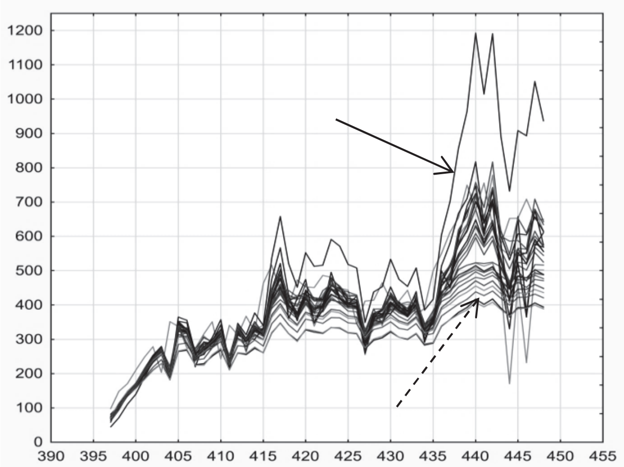

Figure 6 Predictions of all 30 trained networks

First 15 to 20 observations from the evaluation sample are pretty much in line for all networks. That can be interpreted as high forecast performance for the period of approximately first four months of observed period, before dispersion into clusters took place. Regardless of dispersion, very similar patterns (correlation) can be noticed in all cases. In addition, it can be noticed that the upper cluster, in which the original time-series is located, (solid arrow) has much more dense lines.

A benchmark method of grading network adequacy is the performance made on the evaluation subsample.

The SANN uses a cross-validation procedure combining various numbers of parameters and suggesting new networks. In this first step, network configurations like connection weights are set. Along with that, the training was guided by using the validation subsample for fine tuning the previously set parameters. The validation sample is used as selector, to propose the best network/s. At the end, evaluation sample tells how good job has been done.

After generating thirty new models, the top 5 neural networks were listed in Tab. 5. All networks were sorted by validation performance column; thus, unseen data were not influencing final model selection.

Table 5 Performance and error rates of the top 5 trained models.

ID	Configuration	Training performance	Validation performance	Evaluation performance	Training error	Validation error	Evaluation error	Learning algorithm	Error function	Activation function (hidden)	Activation function (output)
33	MLP 5-3-1	0,942748	0,861987	0,915584	2370,709	1707,102	2664,35	BFGS 127	SOS	Tanh	Exponential
8	MLP 5-5-1	0,942014	0,859308	0,909845	2400,361	1716,701	2935,67	BFGS 61	SOS	Exponential	Exponential
11	MLP 5-3-1	0,940490	0,858781	0,918686	2462,696	1729,497	3627,76	BFGS 49	SOS	Exponential	Exponential
15	MLP 5-12-1	0,941676	0,858355	0,897422	2413,984	1709,037	5086,06	BFGS 29	SOS	Exponential	Logistic
10	MLP 5-4-1	0,940783	0,857913	0,895508	2450,240	1759,531	6461,17	BFGS 36	SOS	Exponential	Tanh
28	MLP 5-4-1	0,938771	0,852333	0,928949	2531,055	1763,481	2487,30	BFGS 20	SOS	Identity	Identity
-	Max/Min	0,942748	0,861987	0,928949	2370,71	1707,10	2320,81	-	-	-	-

The validation performance closely follows the training and evaluation sets. The last row of Tab. 5 contains the best overall results of all networks for each category. The best value of the validation performance is of course in the first row, because the whole table is sorted by that column. A good indicator is that the best validated network is also the best trained network, but it is more likely expected, because both sets are included in the network building process. For evaluation group this is not true. The best models by evaluation performance (0,9289) even do not appear in the list of the top 5 networks. In the set of 25 remaining networks, the number of those with evaluation performance value over 0,9 is 15, and three networks have evaluation SOS error under 2450. That fact indicates a problem of selecting the best networks. The best selected network does not have to be the best network from the training set. The most accurate network is marked with ID28 (second last row) and it is on 25th place out of 30 (sorted by validation performance). However, interesting thing is that network 28 does not have the lowest evaluation error despite the best performance grade.

In the second row of Tab. 5 are the network configurations. The configuration structure has three connected numbers, which indicate three layers in the multilayer perceptron architecture. First number represents the number of input nodes, second-the number of hidden nodes and finally number 1 in the last layer represents one output neuron for one ahead forecasting. In the cross-validation SANN procedure, the range of hidden nodes is set between 2 and 12, but almost all top networks show that this number is below or equal to 5. From the top 5 results just one network has the configuration with a higher number of hidden neurons, in this case 12. According to the parsimony principle [2], if networks have similar performance, the one with simpler design has advantage over the complex one. Also, the smaller number of hidden nodes confirms that a simple network can be very powerful and outperform more complex counterparts. For the top several selected networks the exponential activation function dominates, but the best overall model has identity activation function.

In the next graph Fig. 7, top 5 networks are shown together with original series.

The original input time series ATX' line is marked with black dots and the best selected network is marked with a square. The ATX' represents the Austrian Traded Index after processing and preparing for model training. For calculating errors of models, back transitions to the original series (ATX) have to be made. Discrepancies in performance of the top 5 networks are due to classification in different cluster, mentioned earlier. Out of the top 5 selected networks, three of them can be put in the upper cluster (ID 33; 8; 11), and two in the lower one (ID 15, 10).

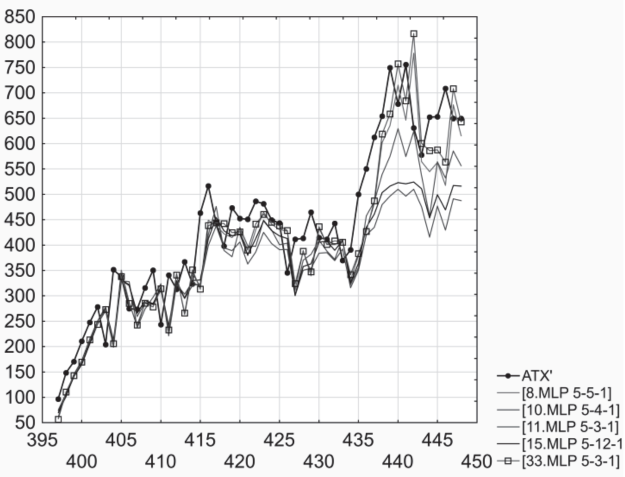

Figure 7 Predictions for the top 5 ANNs

According to Fig 6, it can be seen that approximately half of the generated models are classified in the upper cluster and the other half in the lower. A similar ratio is found in the best 5 selected networks (3/2). Although the original ATX series dominates at the top of 5 models, there are several networks in its class whose values are on average above the original series.

However, it is not uncommon for the original series and model prediction to follow each other (below or above, depending on whether the trend is up or down) which is partially caused by lagging. It can be seen from numerous examples and previous research [5, 14, 15, 16]. One of the reasons is that the model using one-step input to predict one-step in advanced and the learning process of ANN can be explained as first order differential equation

$$y(t+1) = y(t) + h\frac{\mathrm{d}y}{\mathrm{d}t} \tag{4}$$

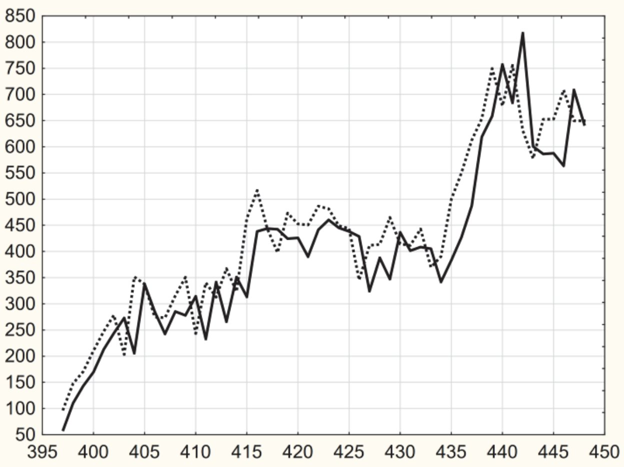

Figure 8 Forecast results of the best selected network (by validation)

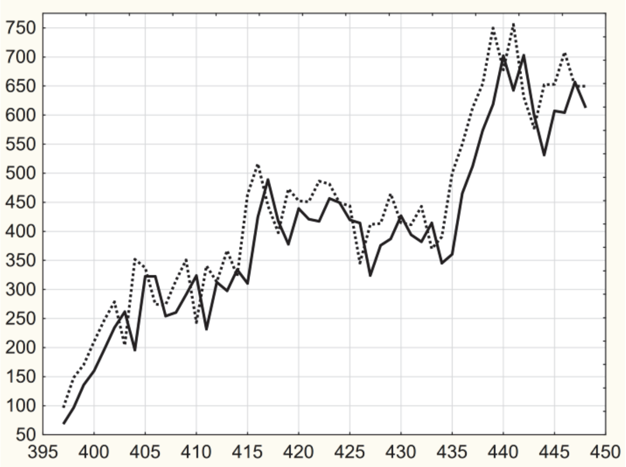

Figure 9 Forecast results of the best network (by evaluation)

Finally, in Fig. 8, comparison between original series and the best selected model series (ID 33) is plotted. The original series ATX' has been drawn with dotted line, and model series with solid line. Prediction series line closely follows target data line. Fig. 8 forecast ATX price movement in the next 52 weeks or in the period of one year, based on the best selected network (ID 33) via validation sample.

On the other hand, Fig. 9 compares original series (dotted line) with predictions of the best model (ID 28, full line) found in training set. This model would not be taken in consideration, because it had a very poor validation performance. From Fig. 9 can be seen that both lines patterns are pretty much matched, thus the correlation results in the highest value in the set (0,92849).

Table 6 Error measures

Time period	MAD ATX(28)	MAD ATX(33)	MSE ATX(28)	MSE ATX(33)	MAPE ATX(28)	MAPE ATX(33)
Month (4)	40,22	35,72	1714,36	1302,54	1,764%	1,571%
Quarter (13)	50,11	42,98	3500,89	2948,09	2,110%	1,812%
Half (26)	54,92	51,45	4453,87	3993,32	2,209%	2,070%
Year (52)	58,38	58,62	4952,48	5308,66	2,128%	2,123%

A good indicator of the model adequacy is the computed error on actual non-processed data, which measures how much predictions are close to real original data series. The ATX' series together with the prediction results have to be translated back in the original space by adding the trend and seasonality.

Error measures which are taken in consideration are MAD (Mean Absolute Deviation), MSE (Mean Squared Error) and MAPE (Mean Absolute Percentage Error). In Tab. 6 error values are shown for the best selected model (ID 33-highest validation performance) and the model with the best evaluation performance (ID 28). Errors were calculated for the period of one month (4 observations), quartile (13), half year (26) and year (53 observations). All types of errors increase their values with progressive time horizon, what is expected. Although model 33 has lower performance on the evaluation set than model 28, almost all error measures record better results for model 33, except for the case of one-year horizon, were errors are slightly in favour of the model 33. Overall lowest mean absolute percentage error is 1,57% for period of one month of the model 33.

5 CONCLUSIONS

Despite the notable difference of data structure between the holdout subsample and the in-sample data, overall results confirm high accuracy of using the method of artificial neural networks. This paper shows various combinations of constructed neural networks and their performances.

Future research can be directed in combining ANN with another method, especially linear one. Also modifying number of input and output nodes can additionally help to find more efficient solution.

6 REFERENCES

[1] Box, G. E. & Jenkins, G. M. (1976). Time series analysis. Forecasting and control. *In Holden-Day Series in Time Series Analysis, Revised ed., San Francisco: Holden-Day.*

[2] Clemen, R. T. (1989). Combining forecasts: A review and annotated bibliography. *International journal of forecasting, 5*(4), 559-583. https://doi.org/10.1016/0169-2070(89)90012-5

[3] Zhang, G. P. (Ed.). (2004*). Neural networks in business forecasting.* IGI global. https://doi.org/10.4018/978-1-59140-176-6

[4] Zhang, G. P. (2004). Business forecasting with artificial neural networks: An overview. In *Neural networks in business forecasting*, 1-22. https://doi.org/10.4018/978-1-59140-176-6.ch001

[5] Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing, 50*, 159-175. https://doi.org/10.1016/S0925-2312(01)00702-0

[6] Haefke, C. & Helmenstein, C. (1996). Neural networks in the capital markets: An application to index forecasting. *Computational Economics*, *9*(1), 37-50. https://doi.org/10.1007/BF00115690

[7] Schittenkopf, C., Dorffner, G., & Dockner, E. J. (1998). Volatility prediction with mixture density networks. *In*

International Conference on Artificial Neural Networks, 929-934). https://doi.org/10.1007/978-1-4471-1599-1_145

[8] Haefke, C. & Helmenstein, C. (2002). Index forecasting and model selection. *Intelligent Systems in Accounting, Finance & Management*, *11*(2), 119-135. https://doi.org/10.1002/isaf.214

[9] Zhang, G. P. (2004). A combined ARIMA and neural network approach for time series forecasting. In *Neural Networks in Business Forecasting*, 213-225. https://doi.org/10.4018/978-1-59140-176-6.ch011

[10] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE, 78*(10), 1550-1560. https://doi.org/10.1109/5.58337

[11] Martinović, M., Stoić, M., Duspara, M., Samardžić, I., & Stoić, A. (2016). Algorithmic conversion of data displayed on a weekly basis to the monthly level using the spreadsheet. *Procedia Engineering*, *149*, 288-296. https://doi.org/10.1016/j.proeng.2016.06.669

[12] Martinović, M., Samardžić, I., & Stoić, A. (2016). Analysis of laptop demand curve through the internet search. *Inovations, Technology, Education and Management / Proceeding of 4th international conference*, ITEM, 2016.

[13] Martinović, M., Požega, Ž., & Crnković, B. (2017). Analysis of Time Series for the Currency Pair Croatian Kuna/Euro. *Ekonomskivjesnik/Econviews-Review of Contemporary Business, Entrepreneurship and Economic Issues, 30*(1), 37-49.

[14] Li, L. K., Pang, W. K., Yu, W. T., & Troutt, M. D. (2004). Forecasting short-term exchange rates: A recurrent neural network approach. *In Neural Networks in Business Forecasting*, 195-212. https://doi.org/10.4018/978-1-59140-176-6.ch010

[15] Brownlee, J. (2019). Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. Retrieved fromhttps://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

[16] Nayak, R. K., Pavitra, S. Y. H., Tripathy, R., & Prathyusha, K. (2019). Forecasting Foreign Currency Exchange Price using Long Short-Term Memory with K-Nearest Neighbour Method. *International Journal of Engineering and Advanced Technology (IJEAT)*, *9*(2). https://doi.org/10.35940/ijeat.B3551.129219

Contact information:

Marko MARTINOVIĆ, mag.ing.rač, v.pred.
(Corresponding author)
University of Slavonski Brod,
Trg Stjepana Miletića 12, 35000 Slavonski Brod, Croatia
E-mail: marko.martinovic@vusb.hr

Anica HUNJET, PhD, Associate Professor
University North,
Trg dr. Žarka Dolinara 1, 48000 Koprivnica, Croatia
E-mail: anica.hunjet@unin.hr

Ioan TURCIN, Assistant in research and teaching
CAMPUS 02 University of Applied Sciences,
Körblergasse 126, 8010 Graz, Austria
E-mail: ioan.turcin@campus02.at