

Methods of Forecasting Based on Artificial Neural Networks

Arthur Stepchenko¹, Arkady Borisov²,¹ *Ventspils University College*,² *Riga Technical University*

Abstract – This article presents an overview of artificial neural network (ANN) applications in forecasting and possible forecasting accuracy improvements. Artificial neural networks are computational models and universal approximators, which can be applied to the time series forecasting with a high accuracy. A great rise in research activities was observed in using artificial neural networks for forecasting. This paper examines multi-layer perceptrons (MLPs) – back-propagation neural network (BPNN), Elman recurrent neural network (ERNN), grey relational artificial neural network (GRANN) and hybrid systems – models that fuse artificial neural network with wavelets and autoregressive integrated moving average (ARIMA).

Keywords – ARIMA ANN, forecasting, GRANN_ARIMA, WANN.

I. INTRODUCTION

Artificial neural networks (ANNs) are a form of artificial intelligence, which attempts to mimic the function of real neurons found in the human brain [2]. ANNs are one of the most accurate and widely used forecasting models that are used in forecasting social, economic, business, engineering, foreign exchange, stock problems and others. Structure of artificial neural networks makes them valuable for a forecasting task with good accuracy.

As opposed to the traditional model-based empirical and statistical methods such as regression and Box-Jenkins approaches, which need prior knowledge about the nature of the relationships between the data, artificial neural networks are self-adaptive methods that learn from data, and only few a priori assumptions about data are needed [1].

Neural networks learn from examples and can find functional relationships among the data even if relationships are unknown or the physical meaning is the baffling [2]. Therefore, ANNs are well suited for problems, whose solutions require knowledge that is difficult to specify but for which there are enough data or observations.

Artificial neural networks can generalize [8]. After learning from the input data (a sample or pattern), ANNs can often correctly process the early unseen sample even if the sample data are noisy. Neural networks are less sensitive to error term assumptions and they can tolerate noise, chaotic components better than most other methods. Artificial neural networks are also universal function approximators. It was proved that a neural network can approximate any continuous function with any accuracy [1].

II. TYPICAL STRUCTURE OF ANN

ANN structure includes input data and artificial neurons that are known as “processing elements”, “nodes” or “units” [10]. The multilayer perceptron includes an input layer, an output layer and one or more intermediate layers called hidden layers. The size and nature of the data set affect the number of hidden layers and neurons within each layer. ANNs with one or two hidden layers perform better than neural networks with the large number of hidden layers.

A. The Propagation of Information in MLPs

The propagation of information in MLPs starts when the input data are taken into the input layer. The inputs are weighted and passed to each node in the next layer. Each processing element in a specific layer is fully or partially connected to many other processing elements using weighted connections [2].

For a time series forecasting problem, a training pattern consists of history data with a fixed number of observations. If we have N observations $y_1, y_2, \dots, y_N$ in the input data set, then using an ANN with n input nodes, we have $N-n$ training patterns that can be used for short-term forecasting – one value ahead. The first training pattern will contain $y_1, y_2, \dots, y_n$ as inputs and y_{n+1} as the output. The second training pattern will contain $y_2, y_3, \dots, y_{n+1}$ as inputs and y_{n+2} as the output. The last training pattern will contain $y_{N-n}, y_{N-n+1}, \dots, y_{N-1}$ inputs and y_N as the output. Then pattern $y_{N-n+1}, y_{N-n+2}, \dots, y_N$ will be used to get forecasting value y_{N+1} . The ANN performs the following unknown function mapping

$$y_{t+1} = f(y_t, y_{t-1}, \dots, y_{t-p}), \quad (1)$$

where y_t is the observation at time t [1].

The scalar weights along with the network architecture store the knowledge of a trained network and determine the strength of the connections between interconnected neurons [7]. If a weight value is zero, then there is no connection between two neurons and if a weight value is negative then a relationship between two neurons is prohibitive. An individual processing element receives weighted inputs from previous layers, which are summed in each node using a combination function, and a bias neuron or threshold is added or subtracted. A bias neuron is connected to every hidden or output unit and a bias neuron value is one.

Neural networks are similar to linear and non-linear least squares regression, and the bias neuron serves a similar purpose as the intercept in regression models. The bias unit is used to scale the input to a useful range $[0, 1]$ or $[-1, 1]$ to improve the convergence properties of the neural network.

The result of this combined summation is passed through a transfer function to produce the nodal output of the processing element (Fig. 1), which is weighted and passed to the processing element in the next layer [2]. The combination function and transfer function together constitute the activation function. In the majority of cases, input layer neurons do not have an activation function, as their role is to transfer the inputs to the hidden layer. The most widely used activation function for the output layer is the linear function as a non-linear activation function may introduce distortion to the predicated output. The sigmoid (logistic), exponential (hyperbolic) tangent, quadratic or linear functions are often used as the hidden layer transfer function. The relationship between the output – predicted value (y_t) – and the inputs – past observations of the time series ($y_{t-1}, \dots, y_{t-p}$) – is given by [5].

$$y_t = w_0 + \sum_{j=1}^q w_j f \left(w_{0,j} + \sum_{i=1}^p w_{i,j} y_{t-i} \right) + \varepsilon_t, \quad (2)$$

where w_j are output layer weights, $w_{i,j}$ are input layer weights, f is a transfer function, q is the number of hidden nodes, p is the number of input nodes, ε_t is a random error at time t .

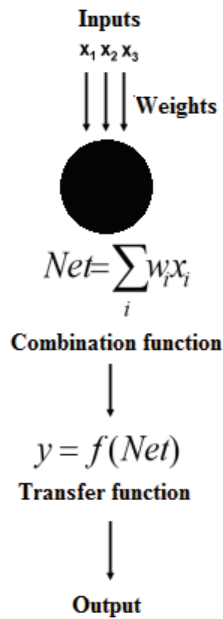


Fig. 1. The propagation of information.

The network corrects its weights and uses a learning rule until it can find a set of weights that will produce the smallest possible error between an observed value and a predicted value at time t . That process is known as “learning” or “training”. For this reason, the network training is actually an unconstrained optimization (nonlinear minimization) problem.

The neural network (2) can approximate any continuous function when the number of hidden nodes q is sufficiently large [5]. In practice, if a network structure has a small number of hidden nodes, then it works well in “out-of-sample” forecasting on data that were not used in training. There is an overfitting effect that can be found in the neural network

modeling process. An overfitted model has a good accuracy on training data, but poor accuracy on “out of the sample” data [9].

To improve the accuracy of the neural network, each data point in the input neurons needs to be normalized – rescaled within the range of $[-1, 1]$ or $[0, 1]$ and standardized to scale data and transformed to make the time series stationary. Transformation can be implemented as taking logarithmic returns of the time series, differencing the time series, etc.

B. Classes of Neural Networks

ANN learning process can be supervised and unsupervised. In supervised learning (e. g., multi-layer feed-forward neural network), the network is presented with historical data, where a training pattern contains independent variables and the corresponding (desired) outputs that are dependent variables in training data. The network then processes the inputs, the nodal output of the network is compared with the observed value of the time series, and an error is calculated. This error is used to correct the connection weights between the model inputs and outputs to reduce the error between the observed values of the time series and outputs predicted by the ANN. The input data used for learning is called the “training set”. Supervised learning is suitable for forecasting and classification tasks. In unsupervised learning (e. g., Kohonen network), there is no dependent variable specified in input (training) data. The network corrects the connection weights according to the input values. The idea of training in unsupervised networks is to cluster the input data into classes of similar features or clusters, where similar input data should generate the same output. This can be referred to as self-organization, and it is suitable for clusterization tasks.

Based on connections between processing elements, ANNs structure can be regarded as feed-forward (e. g., back-propagation) and feedback (e. g., recurrent) networks. Feed-forward network (FNN) propagates information in the forward direction only, while feedback networks propagate information in both the forward and backward directions [2].

Back-propagation neural networks (BPNNs) are a class of feed-forward neural networks with supervised learning rules. The back-propagation network is the most popular and robust multi-layer network that is used in the majority of all forecasting applications. In the learning process, back-propagation neural networks use the gradient-descent search method to correct the connection weights and reduce an error. The main problem of a standard back-propagation algorithm is its slow convergence, which is a typical problem for simple gradient descent methods [1].

Other neural networks that are also used in time series forecasting include recurrent networks, probabilistic networks and fuzzy neural networks. Although feed-forward neural networks are used in many forecasting applications, another type of neural networks – Elman recurrent neural network (ERNN) – is also used in forecasting applications with good accuracy. According to the general principle of the recurrent networks, there is a feedback connection from the outputs of some neurons in the hidden layer to neurons in the context layer that stores the delayed hidden layer outputs. The most important advantage of ERNN is a robust feature extraction ability, when the context layer stores useful information about

data points in past. Since ERNN contains the context layer, it is possible to improve forecasting accuracy by using ERNN instead of FNN [3].

III. HYBRID MODEL: ANN AND WAVELETS

The time series of the real world is often complex in nature and any single forecasting model cannot learn on different patterns equally well. Many studies in time series forecasting have found that forecasting improves in combined models and the integrated forecasting techniques outperform the individual forecasts. In hybrid models, the aim is to reduce the risk that the chosen model will be inappropriate, and combination will help obtain results that are more accurate [5]. Hybrid models can be homogeneous, such as using differently configured neural networks, or heterogeneous, such as with both linear and nonlinear models. Hybrid forecasting has been implemented using a nonlinear model, using ANN with genetic algorithm (GA) or fuzzy logic (FL), or combining linear model and the nonlinear model, using auto-regressive integrated moving average (ARIMA) model with ANN, since in reality time series data typically contain linear and nonlinear patterns [6].

In study [4], the hybrid wavelet and ANN (WANN) model were obtained by combining two methods, discrete wavelet transform and ANN model. ANN model used in this study was the multi-layer feed-forward network.

A. Wavelets

The wavelet transform is a mathematical tool that is used as a time-frequency representation of an analyzed signal. There are some important differences between wavelets and Fourier analysis that is also used as a time-frequency representation of signals. The Fourier coefficients contain only globally averaged information and the Fourier transform does not give local information. Small frequency changes in the Fourier transform will produce changes everywhere in time domain. Wavelets are local in both time and frequency domain. Wavelet transformations provide useful decomposition of the original time series and can get useful information on every decomposition level. Wavelet transforms can be very effective with nonstationary time series data [12]. Many classes of functions can be represented by wavelength in way that is more compact. For a discrete time series $x(t)$ the discrete wavelet transform (DWT) is given by

$$W_{mn} = 2^{-m/2} \sum_{t=0}^{N-1} \psi(2^{-m}t - n)x(t), \quad (3)$$

where W_{mn} is wavelet coefficient for the discrete wavelet, m is integer, which controls the scale, t is time, N is the number of time series data observations, ψ is a transforming function (mother wavelet), n is integer, which controls the time. Therefore, a time series of length N is broken into N components with zero redundancy.

The inverse DWT is given by

$$x(t) = T + \sum_{m=1}^M \sum_{n=0}^{2^{M-m-1}} W_{m,n} 2^{-m/2} \psi(2^{-m}t - n), \quad (4)$$

where, T is the time subseries mean value, M is the number of decomposition level [4].

B. WANN Model

In study [4], the Daubechies wavelet, one of the most widely used wavelet families, was chosen as the wavelet function to decompose the original time series into subtime series components, which were passed to ANN to improve the model accuracy.

To choose the number of decomposition levels, the following formula was used

$$L = \text{int}[\log(N)], \quad (5)$$

where L is the optimal (maximum) number of decomposition levels. In simple format (4) is given by

$$x(t) = A_M(t) + \sum_{m=1}^M D_m(t), \quad (6)$$

where $A_M(M)$ is approximation subseries or residual term at levels M and $D_m(t)$ ($m = 1, 2, \dots, M$) is detailed subseries which can detect small features of interpretational value in the data. The optimal number of decomposition levels for original time series data in this study was three. Original time series was decomposed into three level components ($D1$, $D2$ and $D3$) and approximation component ($A3$). Each of decomposition level component series has a determined role in the original time series and has different effects on the original time series. In this study, the effectiveness of wavelet components was determined using the coefficient of determination (R^2) between each decomposition level component subtime series and original data. The wavelet components $D2$ and $D3$ showed significantly higher R^2 compared to the $D1$ and according to the R^2 analyses, the effective wavelet components were selected as the dominant wavelet components. The combination DW was calculated by

$$DW = A3 + D2 + D3. \quad (7)$$

Fig. 2 shows the structure of the WANN model.

The hybrid model showed a great improvement in time series modeling and produced better forecasts than ANN model alone, as well as GARCH and ARIMA models alone. The study concluded that the forecasting abilities of WANN model were improved when the wavelet transformation technique was adopted for data pre-processing.

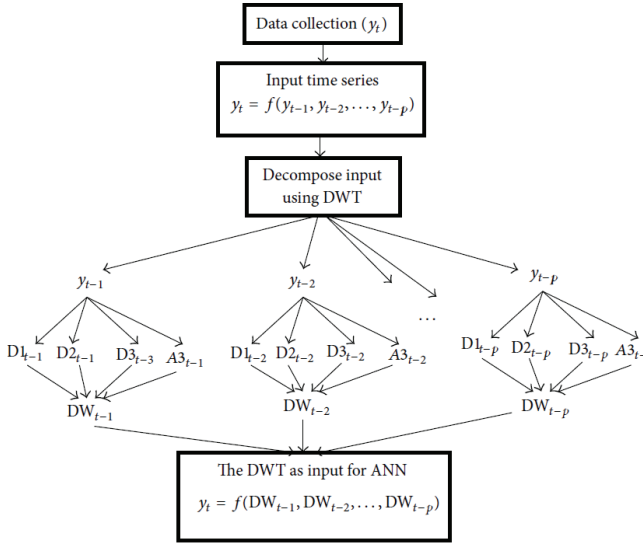


Fig. 2. The structure of the WANN model.

The decomposed periodic components obtained from DWT technique were found to be most effective in giving accurate forecast when they were used as inputs in ANN model. The accurate forecasting results showed that WANN model provides a good and potentially very useful new method for time series forecasting [4].

IV. HYBRID MODEL: ANN AND ARIMA

ANN can model both linear and nonlinear structures in time series; however, they cannot capture both structures equally well. More hybrid forecasting models have been proposed in the last years, using auto-regressive integrated moving average and artificial neural networks. These hybrid models showed good forecasting accuracy [3].

A. Auto-regressive Integrated Moving Average

One of the most widely used linear time series models is the autoregressive integrated moving average (ARIMA) model. In the *ARIMA* (p, d, q) model, the future value of a time series is assumed to be a linear function of several past observations and random errors. In the *ARIMA* (p, d, q) model order p is non-negative integer that refers to the order of the autoregressive function, d is non-negative integer - order of differencing, order q is non-negative integer, which refers to the order of moving average. An auto-regressive integrated moving average process has three different parts – an autoregressive (AR) function that describes how each time series value is a function of the previous p observations, moving average (MA) function describes how each time series value is a function of previous q errors, and an integrated (I) part that describes how to make the data series stationary by differencing d times. The ARIMA model cannot capture nonlinear patterns [11].

Before the ARIMA model can be used for forecasting, check for stationarity is carried out. A stationary time series does not depend on time and, therefore, is characterized by

statistical characteristics such as the mean, variance and autocorrelation structure. When in the observed time series there are trend and heteroscedasticity, then a time series is not stationary. Differencing and power transformation are then applied to the data to remove the trend and to stabilize the variance before an ARIMA model can be used.

If a time series is generated from an ARIMA process, it should have some theoretical autocorrelation properties. By comparing the empirical autocorrelation patterns with the theoretical patterns, it is often possible to identify one or several potential ARIMA models for the given time series. The autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the sample data is often used as the basic tools to identify the order (p, d, q values) of the ARIMA model [5].

B. ANN and ARIMA

In study [5] a time series was considered to be a nonlinear function of several past observations and random errors

$$y_t = f(z_{t-1}, z_{t-2}, \dots, z_{t-p}, e_{t-1}, e_{t-2}, \dots, e_{t-p}), \quad (8)$$

where f is a nonlinear function determined by the neural network, e_t is the residual at time t . Past observation z_t is given by

$$z_t = (1 - B)^d (y_t - \mu), \quad (9)$$

where B is the backward shift (lag) operator, y_t is the original time series value at time t , μ is an ARIMA process generated time series mean value. The residuals are given by

$$e_t = y_t - z_t. \quad (10)$$

Next, a neural network was used to model the nonlinear and linear relationships existing in residuals and original data

$$z_t = w_0 + \sum_{j=1}^q w_j f \left(w_{0,j} + \sum_{i=1}^p w_{i,j} z_{t-i} + \sum_{i=p+1}^{p+q} w_{i,j} e_{t+p-i} \right) + \varepsilon_t, \quad (11)$$

where z_t is a predicted value of the time series by a neural network, f is a transfer function, w_j are output layer weights, $w_{i,j}$ are input layer weights, q is the number of hidden nodes, p is the number of input nodes, and ε_t is a random error at time t . In such hybrids, while the neural network model deals with nonlinearity, the auto-regressive integrated moving average model deals with the non-stationary linear component.

Similar approach was used in study [3]. Here seasonal ARIMA (SARIMA) models were used to analyze the linear part of the time series. Linear part of time series, $\hat{L}_t$, was obtained from the SARIMA model. ANN model used in this study was an Elman recurrent neural network. The ERNN model was developed to fit the residuals obtained from the SARIMA model by

$$e_t = y_t - \hat{L}_t. \quad (12)$$

With p inputs the ERNN model can be written

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-p}) + \varepsilon_t, \quad (13)$$

where f is a nonlinear function determined by the ERNN. The estimation of e_t in (13) will give the forecasting of nonlinear component of time series, $\hat{N}_t$. Therefore, forecasting values of the time series were obtained by adding the estimates of linear and nonlinear components of the time series

$$\hat{y}_t = \hat{L}_t + \hat{N}_t. \quad (14)$$

It was observed that the following model gave a better result than other methods, such as Zhang's hybrid model of feed-forward neural network and ARIMA and Kajitani self-exciting threshold auto-regressive (SETAR) model.

V. HYBRID MODEL: GRANN AND ARIMA

In study [6] GRANN_ARIMA hybrid model (Fig. 3) was used. The grey relational analysis (GRA) was integrated with ANN to remove the redundancy inputs.

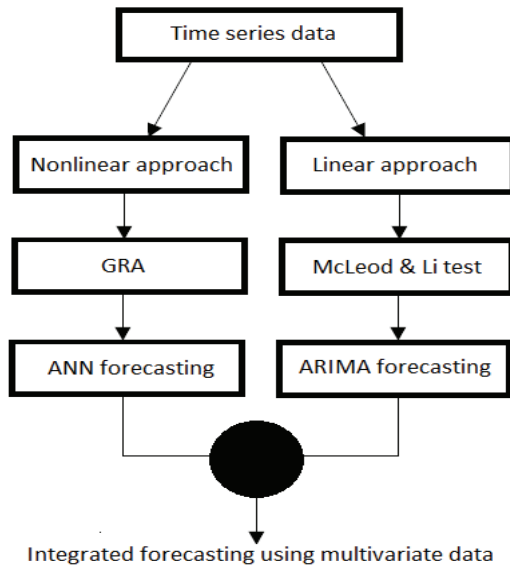


Fig. 3. GRANN_ARIMA model.

GRANN_ARIMA model integrates a nonlinear grey relational artificial neural network (GRANN) and a linear ARIMA model, combining the multivariate time series data and the grey relational analysis to select the appropriate inputs.

A. Grey Relational Analysis

Grey relational analysis is an analysis method that can be used to evaluate the degree of correlation for different data sequences. The degree of correlation between a data sequence (x) and the reference sequence (y) is expressed by a scalar value in interval $[0, 1]$. If the degree of correlation is near 1, it

indicates the high correlation between x and y . There are three main steps in the grey relational analysis. The first step is data pre-processing. Data pre-processing is normally required due to the fact that the range and unit of one data sequence may differ from others. Therefore, data must be normalized, scaled and polarized initially into a comparable sequence before proceeding to other steps. In this study the following equation for data pre-processing was used:

$$x^*(k) = \frac{x_i^0(k) - \min x_i^0(k)}{\max x_i^0(k) - \min x_i^0(k)}, \quad (15)$$

where $i = 1, \dots, m$; $k = 1, \dots, n$, m is the number of experimental data items, n is the number of parameters, $x_i^0(k)$ is the original sequences, $x_i^*(k)$ is the sequences after data pre-processing, $\min x_i^0(k)$ and $\max x_i^0(k)$ are the smallest and the largest value of $x_i^0(k)$, respectively. The range of data is corrected to be in range $[0, 1]$.

The second step is to calculate the grey relational coefficient by using

$$\xi_i(k) = \frac{\Delta \min + \zeta \Delta \max}{\Delta_{0,i}(k) + \zeta \Delta \max}, \quad (16)$$

where $\xi_i(k)$ is a grey relational coefficient at any data point (k), ζ is known as an identification coefficient within interval $[0, 1]$, and normally $\zeta = 0.5$ is used. $\Delta_{0,i}$ are deviation sequences of the reference sequence and comparability sequence

$$\Delta_{0,j} = \|x_0^*(k) - x_j^*(k)\|, \quad (17)$$

where $x_0^*(k)$ is the reference sequence and $x_i^*(k)$ is the comparative sequence. $\Delta \min$ in (16) is given by

$$\Delta \min = \min_{\forall j \in i \forall k} \|x_0^*(k) - x_j^*(k)\|, \quad (18)$$

and $\Delta \max$ in (16) is given by

$$\Delta \max = \max_{\forall j \in i \forall k} \|x_0^*(k) - x_j^*(k)\|. \quad (19)$$

The grey relational grade is the average value of the grey relational coefficients and is defined as

$$\gamma_i = \frac{1}{n} \sum_{k=1}^n \xi_i(k), \quad (20)$$

where n is the number of the reference sequence, $x_0^*(k)$. The grey relational grade γ_i represents the level of correlation between the reference sequence and the comparability sequence.

B. GRANN_ARIMA model

In the GRANN_ARIMA model, ARIMA is used as a linear model, L_t and GRANN is used as a nonlinear model, N_t

$$y_t = L_t + N_t. \quad (21)$$

Here y_t is a value of the original time series. As $\hat{N}_t$ is the forecast value of the GRANN model at time t , then the residuals e_t are obtained by

$$e_t = y_t - \hat{N}_t. \quad (22)$$

The residuals now represent the linear part of the data, and ARIMA can be used to model the residual. Residual modeling by ARIMA can be represented similarly as in (13), but here f is a linear function modeled by the ARIMA model. Therefore, the hybridized forecast model can be written as in (14), but here linear part of model $\hat{L}_t$ is obtained from (13). Unlike model in (8), a nonlinear model is implemented first rather than followed by the linear model.

To validate the performance of the proposed model, Kuala Lumpur Stock Exchange (KLSE) daily close price was used as a time series data. Grey relational analysis was used as a feature selection tool, and four factors were found as the most influential factors relating to KLSE close price: syarian index, trading/service index, composite index and industrial index. A three-layer feed-forward neural network with a single output unit, nine hidden units and four input units with the learning rate 0.5 and momentum 0.9 was used to model nonlinear part of forecast. ARIMA (0,1,3) model was used to model the residuals of close price – linear part of forecast. The network structure and learning parameters were determined by trial and error. The forecasting accuracy was compared with several models, and these include individual models (ARIMA, multiple regression, grey relational artificial neural network), several hybrid models (MARMA, MR ANN, ARIMA ANN), and the artificial neural network trained using the Levenberg Marquardt algorithm. The experiments showed that the GRANN_ARIMA model outperformed other models with MAPE error of 0.16 % and 99.84 % forecasting accuracy. The empirical results obtained showed that the GRANN_ARIMA model could be a good alternative for time series forecasting due to its promising forecasting accuracy.

VI. CONCLUSION

In this paper, an overview of artificial neural network applications in forecasting and possible forecasting accuracy improvements was presented. Artificial neural networks are computational models and universal approximators that can be applied to the time series forecasting with a high accuracy. Back-propagation neural networks are a class of feed-forward neural networks and are most popular and robust multi-layer networks used in the majority of all forecasting applications. Although feed-forward neural networks are used in many forecasting applications of ANNs, other type of neural networks – Elman recurrent neural network (ERNN) – is also used in forecasting applications with good accuracy. In the case of comparison with other types of multi-layer network, the most important advantage of ERNN is robust feature extraction ability. Since ERNN contains the context layer, it is possible to improve forecasting accuracy by using ERNN

instead of FNN. The time series of the real world is often complex in nature and any single forecasting model cannot learn on different patterns equally well. Forecasting improves in combined models and the integrated forecasting techniques outperform the individual forecasts. Hybrid models can be homogeneous, such as using differently configured neural networks, or heterogeneous, such as with both linear and nonlinear models, since in reality time series data typically contain both linear and nonlinear patterns. Hybrid models WANN, ARIMA ANN and GRANN_ARIMA showed good forecasting accuracy and outperformed other forecasting models.

REFERENCES

- [1] G. Zhang, B. E. Patuwo and M. Y. Hu, "Forecasting with artificial neural networks: the state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, March 1998. [Online]. Available: ResearchGate, <http://www.researchgate.net/>. [Accessed July 15, 2014]. [http://dx.doi.org/10.1016/S0169-2070\(97\)00044-7](http://dx.doi.org/10.1016/S0169-2070(97)00044-7)
- [2] M. A. Shahrin, M. B. Jaksa and H. R. Maier, "State of the art of artificial neural networks in geotechnical engineering," *EJGE Special Volume Bouquet 08*. [Online]. Available: EJGE, <http://www.ejge.com/>. [Accessed July 18, 2014].
- [3] C. H. Aladag, E. Egrioglu and C. Kadilar, "Forecasting nonlinear time series with a hybrid methodology," *Applied Mathematics Letters*, vol. 22, no. 9, pp. 1467–1470, September 2009. [Online]. Available: ScienceDirect, <http://www.sciencedirect.com/>. [Accessed July 20, 2014]. <http://dx.doi.org/10.1016/j.aml.2009.02.006>
- [4] A. Shabri and R. Samsudin, "Daily crude oil price forecasting using hybridizing wavelet and artificial neural network model," *Mathematical Problems in Engineering*, vol. 2014, article ID 201402, July 2014. [Online]. Available: Hindawi, <http://www.hindawi.com/>. [Accessed July 22, 2014]. <http://dx.doi.org/10.1155/2014/201402>
- [5] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Systems with Applications*, vol. 37, no. 1, pp. 479–489, January 2010. [Online]. Available: Science Direct, <http://www.sciencedirect.com/>. [Accessed July 26, 2014]. <http://dx.doi.org/10.1016/j.eswa.2009.05.044>
- [6] R. Sallehudin, S. M. H. Shamsuddin, S. Z. M. Hashim and A. Abraham, "Hybrid Grey Relational Artificial Neural Network and Auto Regressive Integrated Moving Average Model for Forecasting Time-Series Data," *Applied Artificial Intelligence: An International Journal*, vol. 23, no. 5, pp. 443–486, April 2009. [Online]. Available: ResearchGate, <https://www.researchgate.net/>. [Accessed July 30, 2014].
- [7] M. Adya and F. Collopy, "How effective are Neural Networks at Forecasting and Prediction? A Review and Evaluation," *Journal of Forecasting*, vol. 17, no. 5–6, pp. 481–495, Sep.–Nov. 1998. [Online]. Available: Weatherhead School of Management, <http://weatherhead.case.edu/>. [Accessed July 21, 2014]. [http://dx.doi.org/10.1002/\(SICI\)1099-131X\(199809\)17:5/6<481::AID-FOR709>3.0.CO;2-Q](http://dx.doi.org/10.1002/(SICI)1099-131X(199809)17:5/6<481::AID-FOR709>3.0.CO;2-Q)
- [8] S. T. A. Niaki and S. Hoseinzade, "Forecasting S&P index using artificial neural networks and design of experiments," *Journal of Industrial Engineering International*, vol. 9, no. 1, Dec. 2013. [Online]. Available: Springer, <http://link.springer.com/>. [Accessed July 24, 2014].
- [9] N. Q. Hung, M. S. Babel, S. Weesakul and N. K. Tripathi, "An artificial neural network model for rainfall forecasting in Bangkok, Thailand," *Hydrology and Earth System Sciences*, vol. 13, no. 8, pp. 1413–1425, 2009. [Online]. Available: Hydrology and Earth System Sciences, <http://www.hydrol-earth-syst-sci.net/>. [Accessed July 25, 2014]. <http://dx.doi.org/10.5194/hess-13-1413-2009>
- [10] I. Kasstra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, no. 3, pp. 215–236, Apr. 1996. [Online]. Available: Geocities, <http://www.geocities.ws/>. [Accessed July 27, 2014]. [http://dx.doi.org/10.1016/0925-2312\(95\)00039-9](http://dx.doi.org/10.1016/0925-2312(95)00039-9)
- [11] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003. [Online]. Available: Geocities, <http://www.geocities.ws/>. [Accessed July 28, 2014]. [http://dx.doi.org/10.1016/S0925-2312\(01\)00702-0](http://dx.doi.org/10.1016/S0925-2312(01)00702-0)

- [12] L. F. Ortega, "A Neuro-wavelet Method for the Forecasting of Financial Time Series," *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, San Francisco, USA, Oct. 24–26, 2012. [Online]. Available: International Association of Engineers, <http://www.iaeng.org/>. [Accessed July 29, 2014].

Arthur Stepchenko is the first-year Doctoral Student majoring in Information Technology at Riga Technical University (RTU). He received his Mg.sc.comp. degree in Computer Science from Ventspils University College (VUC) in 2013.

He is a Research Assistant at the Department of Space Technology of the Engineering Research Institute "Ventspils International Radio Astronomy Centre", Ventspils University College (VIRAC).

His research interests include remote sensing, digital image processing, machine learning, data mining and programming.

E-mail: arturs1312@gmail.com.

Arkady Borisov received his Doctoral Degree in Technical Cybernetics from Riga Polytechnic Institute in 1970 and Dr.habil.sc.comp. degree in Technical Cybernetics from Taganrog State Radio Engineering University in 1986.

He is a Professor of Computer Science at the Faculty of Computer Science and Information Technology, Riga Technical University. His research interests include fuzzy sets, fuzzy logic, computational intelligence and bioinformatics. He has 235 publications in the field.

He is a member of IFSA European Fuzzy System Working Group, Russian Fuzzy System and Soft Computing Association, honorary member of the Scientific Board, member of the Scientific Advisory Board of the Fuzzy Initiative Nordrhein-Westfalen (Dortmund, Germany).

Address: 1 Kalku Str., Riga, LV-1658; phone: +371 67089530.

E-mail: arkadijs.borisovs@cs.rtu.lv.

Artūrs Stepčenko, Arkādijs Borisovs. Prognozēšanas metodes, kas balstītas uz mākslīgajiem neironu tīkliem

Mākslīgie neironu tīkli (MNT) ir elastīga skaitļošanas struktūra un universāls aproksimators, kurus var pielietot plaša spektra laika rindu (ekonomikā, biznesā, inženierzinātnēs, valūtu apmaiņā un citās sfērās) prognozēšanai ar augstu precizitāti. Atšķirībā no tradicionālām empiriskām un statistiskām metodēm, kā, piemēram, regresijas analīzes vai Boksa-Dženkinsa pieejām, kur ir nepieciešamas plašas zināšanas par problēmas datu īpašībām, mākslīgie neironu tīkli ir pašmācības metode, kas prasa zināt tikai nelielu informācijas daudzumu par konkrēto problēmu. MNT spēj vispārināt – tie mācās no paraugiem un spēj atrast sakarības starp datiem pat tad, ja šīs sakarības nav zināmas, vai to fiziskā nozīme ir grūti izprotama, kā arī ja ieejas dati satur troksni un haotiskas komponentes. Tāpat MNT var aproksimēt jebkuru nepārtrauktu funkciju ar jebkādu vēlamu precizitāti. Visplašāk laika rindu prognozēšanai tiek lietoti daudzslāņu kļūdu atgriezeniskās izplatīšanas neironu tīkli, kas pieder vienvirziena tīklu klasei, taču ir arī cits neironu tīklu veids – Elmana rekurentais neironu tīkls. Tas ir uzrādījis labus prognozēšanas rezultātus. Daudzslāņu mākslīgais neironu tīkls sastāv no ieejas slāņa, viena vai vairākiem slēptiem slāņiem un izejas slāņa. Uz treniņa datiem apmācīto modeli ar noteiktiem svāriem var pielietot jau nezināmas laika rindas vērtības prognozēšanai testa datos. Tā kā reālās problēmas ir sarežģītas pēc dabas, tad neviens atsevišķs prognozēšanas modelis nespēj mācīties no dažādiem datu paraugiem vienādi labi. Prognozēšanas precizitāte uzlabojas, ja modeļi tiek kombinēti. Prognozēšanas hibrīdmodeļu mērķis ir samazināt risku, kas var rasties, izmantojot nepiemērotu modeli, un paaugstināt prognozēšanas precizitāti. Hibrīdmodeļi var būt homogēni, piemēram, apvienojot dažādu konfigurāciju neironu tīklus, vai heterogēni, apvienojot lineāru un nelineāru modeļi, tādā reālās laika rindas parasti satur gan lineāras, gan nelineāras komponentes. Labu prognozēšanas precizitāti ir sasnieguši modeļi, kas apvieno veivletu analīzi (dekompozīciju) ar MNT, un modeļi, kas apvieno autoregresijas integrēto mainīgo vidējo (ARIMV) ar MNT. Veivletu dekompozīcija tiek izmantota, lai iegūtu laika rindas laika-frekvenču reprezentāciju. Oriģināla laika rinda ar veivletu dekompozīcijas palīdzību var tikt sadalīta vairākās laika apakšrindu komponentēs, un jauna laika rinda, kas sastāv no šīm komponentēm, tiek nosūtīta MNT prognozēšanai. Kā citu hibrīdu veidu var izmantot MNT ar ARIMV. Tā kā ARIMV ir paredzēts lineāru procesu prognozēšanai, tad to var izmantot lineāru prognozes komponentu iegūšanai, bet MNT var izmantot nelineāro prognozes komponentu iegūšanai, un vēlāk šīs komponentes tiek saskaitītas un dod galīgo prognozi.

Артур Степченко, Аркадий Борисов. Методы прогнозирования, основанные на искусственных нейронных сетях

Искусственные нейронные сети (ИНС) являются гибкой вычислительной структурой и универсальным аппроксиматором, они могут быть использованы в прогнозировании временных рядов широкого спектра (в экономике, бизнесе, инженерной науке, обмене валют и других сферах) с высокой точностью. В отличие от традиционных эмпирических и статистических методов, как например, регрессионного анализа или подходов Бокса-Дженкинса, где необходимы широкие знания о свойствах данных проблемы, ИНС являются методом самообучения, которым необходим только небольшой объем данных о проблеме. ИНС способны к обобщению – они обучаются по образцам, и могут найти взаимосвязь между данными, даже если эта взаимосвязь неизвестна или её физический смысл трудно понять, или даже если входные данные содержат шум и хаотические компоненты. ИНС могут аппроксимировать любую непрерывную функцию с любой желаемой точностью. Самый распространённый вид ИНС для прогнозирования временных рядов это многослойная сеть обратного распространения ошибки, которая принадлежит к классу сетей прямого распространения, но и другой вид нейронных сетей – рекуррентная сеть Элмана – показала хорошие результаты прогнозирования. Модель, обученную на тренировочных данных с определёнными весами, можно использовать для прогнозирования неизвестной величины временного ряда в тестовых данных. Так как реальные проблемы по природе являются сложными, то ни одна отдельная модель не может обучаться на разных выборках данных одинаково хорошо. Точность прогнозирования увеличивается, если модели комбинируются. Целью гибридных моделей прогнозирования является уменьшение риска, который может возникнуть при выборе неправильной модели, и повышение точности прогнозирования. Гибридные модели могут быть однородные, например, объединяя ИНС разных конфигураций, или неоднородные, объединяя линейную и нелинейную модель, так как реальные временные ряды содержат и линейную и нелинейную компоненту. Хорошей точности прогнозирования достигли модели, которые объединяют вейвлет-анализ (декомпозицию) с ИНС, и модели, которые объединяют интегрированную модель авторегрессии скользящего среднего (ИМАРСС) с ИНС. Вейвлет-декомпозиция используется, чтобы получить частотно-временное представление временного ряда. Оригинальный временной ряд с помощью вейвлет-декомпозиции может быть разложен на несколько компонент, и новый временной ряд, который состоит из этих компонент, передаётся в ИНС для прогнозирования. Другим видом гибрида может быть ИНС вместе с ИМАРСС. Так как ИМАРСС предназначен для прогнозирования линейных процессов, то с его помощью можно получить линейную компоненту прогноза, а ИНС может использоваться для прогнозирования нелинейной компоненты прогноза, потом эти компоненты складываются и дают окончательный результат.