Frequency Decomposition in Predictive Error Compensating Wavelet Neural Network

Ajla Kulaglic

Istanbul Technical University, Istanbul, Turkey

Abstract

This paper presents an extended study of the previously proposed Predictive Error Compensation Neural Network (PECNET) model. Different frequencies are used as input, in addition with the use of the Butterworth filter and the model performances are compared. The results show that the PECNET with frequency decomposition and Butterworth filter applied to input data provides significantly more accurate predictions for stock price prediction problem with respect to previous studies and conventional machine learning and time series prediction methods without changing any hyperparameter or the structure. In addition, the time and space complexity of the PECNET model is less than all other compared machine learning methods.

Key Words: *predictive error compensated neural network, Butterworth filter, frequency decomposition, wavelet transform, stock price forecasting*

1. Introduction

In the time of global financial changes, forecasting financial time series data is a significant challenge that even trading robots can hardly predict. 80% of the stock markets nowadays are controlled by machines, and according to Forbes (Kindig, 2020), robots will replace 200,00 banking businesses in the next ten years. High-Frequency trading technologies are a type of algorithmic trading that uses machine learning algorithms to implement investment strategies in brief time intervals. Stock market prediction is essential when making the proper decision (Fama, 1993). However, evaluating input data and their appropriate frequencies is critical regarding machine learning problems. For this purpose, more relevant frequencies have been determined that can be used to improve the forecasting performance without causing overfitting problems or increasing the complexity of the proposed algorithm. On the other hand, stock price data are disposed to frequent changes that cannot be derived from the historical trend. Changes are influenced by real-world factors, such as political, social, and environmental factors (Novak *et al.*, 2016). In addition, the noise-to-signal ratio is very high in such conditions, and it is difficult to analyze and forecast future data. The use of econometric models is convenient for describing and evaluating the relationships between variables using statistical inference, with some limitations. These limitations can be seen in the inability to capture the nonlinear nature of stock prices. In addition, (Abu-Mostafa and Atiya, 1996) in their study assumed constant variance while the financial time series are boisterous and have time-varying volatility.

Statistical methods such as Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), and vector autoregression have generally achieved reasonable predictive results based on the results found in the literature (Box, 2013) and (Reddy, 2019). However, according to the statistical models, Artificial Neural Networks (ANNs) are one of the most accurate prediction models (Khashei and Bijari, 2010). According to (Hornik *et al.*, 1989), ANNs with a given sufficient amount of data can approximate any finite and continuous function based on the universal approximation theorem. The first significant study of a neural network model to predict stock price returns were made by (White, 1988), where he introduced a prediction model based on IBM's daily common stock and achieved promising results. Various hybrid systems using ANN have been proposed to increase prediction performance, the Hidden Markov Model (HMM), (Hassan *et al.*, 2007), exponential smoothing, and ARIMA (Wang *et al.*, 2012), and ANN with exponentially generalized autoregressive conditional heteroscedasticity model (Hajizadeh *et al.*, 2012). The two most popular deep-learning architectures for stock market forecasting in recent years are the Long Short-Term Memory (LSTM) model and the Gated Recurrent

Unit (GRU) model with its hybridization (Shahi *et al.*, 2020). LSTM models are appropriately structured to learn temporal patterns and overperform the conventional recurrent neural networks (RNNs). LSTM and GRU deep-learning architectures are proposed and performances of these two models are compared for stock market predictions in (Shahi *et al.*, 2020). In their study, they compared the performance of the LSTM and GRU models under the same conditions. Also, they showed that the predicting model could be significantly improved by including contextual information such as financial news sentiments and stock market features. Bao *et al.* (Bao *et al.*, 2017) used the LSTM for stock price forecasting using different types of sequential data. Using the sentiment features, Li *et al.* (Li *et al.*, 2017) have shown that LSTMs surpass benchmark models of SVM and improve the accuracy of next-day opening price predictions. In prediction models, the problem of overfitting and getting stuck in local optima are additional issues that must be considered. The problem is due to the limited amount of data and appropriate model configuration. The financial time series data yearly obtain approximately 252 data points. However, this is insufficient for the Deep Neural Networks (DNN) models compared to the number of model parameters (Goodfellow *et al.*, 2016). Sufficient data is needed as the number of model parameters increases as the number of features used is enlarged. Although NN models achieve better generalization, they are prone to overfitting due to their high capacity. Regularization techniques can prevent overfitting, but they cannot improve generalization performance. Hence, data augmentation is a method used in order to avoid overfitting while improving generalization accuracy. However, regarding financial time series, the data augmentation distorts the original data. Consequently, in recent times, signal processing techniques have recently been used to transform data into a format that reveals certain characteristics. The results showed that the extracted features could achieve more accurate predictions than the data without feature extraction. In this study, the proposed studies in (Ustundag and Kulaglic, 2020), (Kulaglic and Ustundag, 2021), and (Kulaglic and Ustundag, November 2021) are extended by using different frequencies as input to the Predictive Error Compensation Neural Network (PECNET) model. In addition, the use of the Butterworth filter on the input data is tested, and the model performances are compared. The rest of this chapter is organized as follows. Section 2 describes the differences between the proposed model. The Butterworth filter is a signal processing technique with a frequency response as flat as mathematically possible in the bandwidth. Experimental results with discussion are given in section 3. Conclusions and proposed remarks on future work are given in section 4.

2. Predictive Error Compensated Neural Network Model with Butterworth Filter

The Predictive Error Compensating Neural Network (PECNET) model presented in this section is an extension of the previously proposed model. The model utilized in this study also consists of four separately trained neural networks, as demonstrated in Figure 1.

The main discrepancy is manifested in the first network, wherein place of the average filtering that was previously applied, the Butterworth filter (BF) is used. Butterworth filters have the sharpest roll-off possible without inducing a peak in the Bode plot, and because of that, they are called maximally flat filters (Ellis, 2012). The general formula for BF depends on the order of the applied filter. For continuous-time Butterworth filters, the poles associated with the squares of the frequency response magnitude are equally distributed in the angle on the circle in the s-plane, concentric with the origin and radius equal to the cut-off frequency. The poles that characterize the system function are readily obtained when the cutoff frequency and filter order are specified. Once the poles are specified, getting the differential equation characterizes the filter is straightforward. The response of the Butterworth filter is given in Equation (1):

$$| B(j\cdot\omega)|^2 = 1/(1+(j\cdot\omega/ j\cdot\omega_c)^2) \qquad (1)$$

Where the constant is the 68 400 sample frequency (one day is 68 400 seconds). It is easy to show that the first derivative of $B(j\cdot\omega)^2$ at ω is equal to zero (ω =0). For this reason, Butterworth's response is maximally flat at ω =0.

Later, the current input of each parameter is shifted to the previous values using the unit time delay

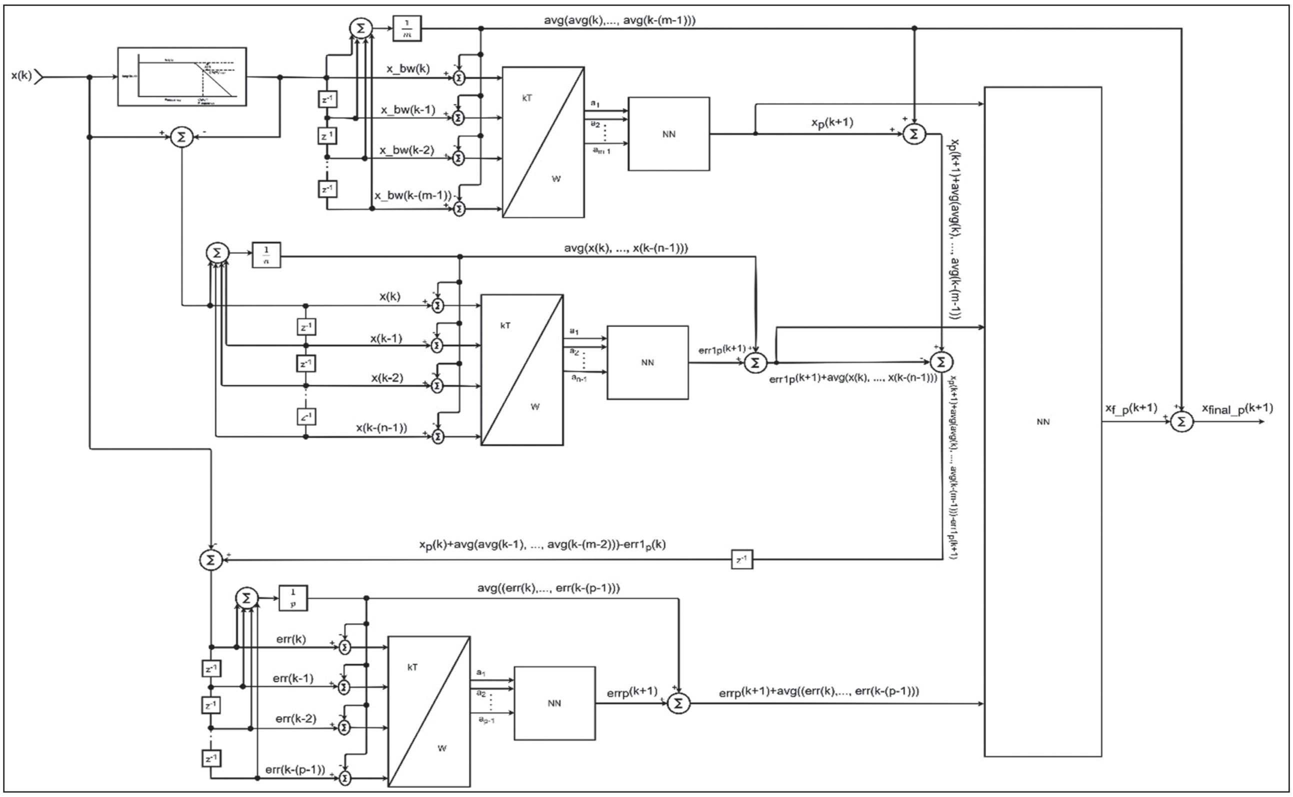

Figure 1 The predictive error compensated neural network model with Butterworth filtering

operator z^{-1} as in previous studies. The error pattern obtained in the main network is applied as the output of the second network, where the difference between the primary data and the BF filtered data is used as input. Additional and separately trained NN with error data patterns of previously trained network have been used. The final neural network merges the outputs of the neural networks in a cascaded part. The average subtraction normalization used as a normalization technique due to the problems in traditional normalization approaches has been proposed and discussed in (Kulaglic and Ustundag, 2021). Normalization of the average subtraction allows us to build a normalization method representing differed volatilities and preserving the original time series properties inside the input sequence. The normalized time series data are preprocessed by discrete wavelet transform (DWT), where the obtained coefficients are used as input to the NNs. The decomposed signal y[n] consists of high- and low-frequency components, as shown in Equation (2). The input signal is presented with x[n]. The low- and high-pass filters are represented by h[n] and g[n], respectively. The Haar wavelet filters have been used as they significantly reduce distortion rate during signal decomposition and reconstruction, and also considerably reducing processing and computational time (Ustundag and Kulaglic, 2020).

$$y[n] = y_{high}[n-1] + y_{low}[n-1] \quad (2)$$

The low-pass outputs recursively pass through an identical group of filter banks in order to use different resolutions in each phase. The filtering process is mathematically expressed using Equations (3) and (4). Equations (3) and (4) provide an approximation and detailed signal, respectively.

$$y_{high}[n-1] = \Sigma\, g[k].x[2n-k] \quad (3)$$

$$y_{low}[n-1] = \Sigma\, h[k].x[2n-k] \quad (4)$$

The neural network configuration consists of input, hidden, and output layers for predicting n-step-ahead time series data. Employed networks have the same network configurations. Regarding different formulas found in the literature (Goodfellow, 2016), (Moshiri and Cameron, 2000) and (Patterson, 1996), the number of neurons in the hidden layers are selected using (Patterson, 1996). The activation function used for these networks is Rectified Linear Unit (ReLU). In comparison with sigmoid and hyperbolic tangent activation functions, the linear activation function ReLU notably improves the achievement of feed-forward networks (Goodfellow, 2016). The learning rate and momentum of the Stochastic Gradient Descent (SGD) optimization algorithm are 0.05 and 0.75, respectively. In addition to the effect of different filtering methods, the impact of frequency decomposition on the presented model has been investigated. In this regard, high, medium, and low frequencies have been applied to the financial data and ways to improve model performance without compromising model accuracy or causing overfitting have been explored. In order to obtain the spatial resolution, we also included the additional parameters to see the performance of the proposed model. An additional network with supplementary data is added to the proposed model. In order not to increase complexity or cause the overfitting rate by applying the supplementary data to the same NN, we added new parameters into the additional network.

3. Model evaluation

The experimental setup uses the stock price data obtained from the Istanbul Stock exchange, Borsa Istanbul. For this purpose, data from the banking sector and the stock exchange index are used, Is Bank, Garanti Bank, AK Bank, and BIST30 (index of 30 companies). Data were collected using web services Investing (investing.com) for daily closing prices and Matriks (Information Distribution services Matrix) for hourly data. The experimental results are measured with the root-mean-square error (RMSE) and root-mean-square percentage error (RMSPE) (Equations (5) and (6)). Mathematical formulations are given below, where Pi is accurate, and Oi is estimated values in time i. The number of data samples is given by n.

$$RMSE = \sqrt{[\ \Sigma(Pi - Oi)^2 / n\ ]} \quad (5)$$

$$RMSPE=100/n * \Sigma|(Pi-Oi)/Pi \quad (6)$$

The RMSPE (Table 1) and RMSE (Table 2) errors for disseminated frequencies are presented. First, the results for the average filtering used in the primary network are shown. Secondly, the results of the improvements done by applying different frequencies to the PECNET model are presented. The frequency decomposition is done using weekly, daily, and hourly data.

Table 1. The RMSPE (%) average filtering results for PECNET.

W/D	AK	IS	GARANTI	BIST30
1NN	2.420932	1.684884	1.929106	1.581213
2NN	1.884708	1.735879	1.673334	1.247089
3NN	1.831507	1.621118	1.658614	1.168715
4NN	1.615228	1.117738	1.634853	0.842569

Table 2. The RMSE (TL) average filtering results for PECNET.

W/D	AK	IS	GARANTI	BIST30
1NN	0.131046	0.090225	0.176524	0.247961
2NN	0.09956	0.07617	0.240385	0.191119
3NN	0.098091	0.066546	0.242836	0.170376
4NN	0.089519	0.059838	0.150628	0.130173

The results are presented in Table 3, (RMSPE) and in Table 4 (RMSE). The improvements in forecasting performances can be seen by increasing the frequencies in the proposed model comparing the results where only weekly and daily data are used.

Table 3. The RMSPE results for increasing frequency applied to the PECNET.

W/D/H	AK	IS	GARANTI	BIST30
1NN	2.422607	1.581548	1.925032	1.587106
2NN	1.750624	1.229699	1.679927	1.266596
3NN	1.710756	1.220028	1.662537	0.960022
4NN	1.311037	1.07663	1.556721	0.860772
5NN	1.263475	1.070055	1.219931	0.797137

Table 4. The RMSE results for increasing frequency applied to the PECNET.

W/D/H	AK	IS	GARANTI	BIST30
1NN	0.131534	0.08455	0.176067	0.24241
2NN	0.09478	0.065632	0.191322	0.194238
3NN	0.092534	0.065115	0.170537	0.147811
4NN	0.069717	0.057155	0.143131	0.131245
5NN	0.062936	0.057059	0.112433	0.128643

If the frequencies are reduced, and the amount of information and data used increases, the model performances are significantly reduced (Tables 5 and 6).

Table 5. The RMSPE results for decreasing frequency applied to the PECNET.

H/D/W	AK	IS	GARANTI	BIST30
1NN	2.962316	2.228355	2.779509	2.085103
2NN	1.496557	1.489793	1.578333	0.918139
3NN	1.309792	1.438456	1.502039	0.747594
4NN	1.276866	1.368016	1.313222	0.811673
5NN	1.205566	1.238978	1.30025	0.737473

Improvements in performance results are also noticed when the Butterworth filter is applied. First, the appropriate frequencies were selected.

The sampling frequency, since daily data are used was 1/days (in seconds). One day has 86 400 seconds, so the sampling frequency is 1/86 400. The cut-off frequency 1/n· days(in seconds) where n is selected as 4. The lowest error is obtained when the ninth order of the BF filter is applied.

Table 6. The RMSE results for decreasing frequency applied to the PECNET.

H/D/W	AK	IS	GARANTI	BIST30
1NN	0.163362	0.121518	0.25242	0.333386
2NN	0.082705	0.080917	0.245359	0.246327
3NN	0.07862	0.077063	0.239065	0.216419
4NN	0.083928	0.074037	0.221789	0.19813
5NN	0.080243	0.067509	0.210701	0.181385

Table 7. The RMSPE results for when the Butterworth filter is applied to the input signal.

D	AK	IS	GARANTI	BIST30
1NN	3.716282	2.55976	3.189103	2.760657
2NN	2.607084	2.077328	2.519306	2.009513
3NN	1.541605	1.02821	1.576386	0.91528
4NN	1.520222	0.998691	1.57281	0.774124

Table 8. The RMSE results for when the Butterworth filter is applied to the input signal.

D	AK	IS	GARANTI	BIST30
1NN	0.202757	0.137869	0.286422	0.436493
2NN	0.144097	0.113562	0.229782	0.318334
3NN	0.078491	0.05527	0.146178	0.200641
4NN	0.072492	0.053673	0.143009	0.140798

The results indicated that applying BF filter to the input data set can further improve the model performances (Table 7 and Table 8).

Table 9. The correlation between Far-eastern stock indices to the BIST30 index.

	N225	*HIS*	*ASX*	*BIST30*
N225	1			
HIS	0.525851	1		
ASX	0.779896	0.404533	1	
BIST30	0.861042	0.652458	0.718566	1

Last but not least, the spatial dimension is included in the proposed model. The spatial dimension is obtained using the different indices from stock exchanges that close before the stock market in Turkey. For this purpose, a parameter that contains information from Far Eastern stock market indices has been constructed. This idea is based on the correlation between Far East market indices and the Istanbul stock market index (BIST30). Only a few Far Eastern indices are used at this stage, such as the Nikkei index (N225), the Tokyo Stock Exchange index, Hang Seng Index (HSI), the stock market index in Hong Kong, and the Australian Securities Exchange index (AXS), and its correlation with the BIST30 index is given in Table 9. The new parameter is constructed from the average normalized values of Far Eastern indices. The constructed parameter is used as an input to the proposed model. Performance improvement is noticed by applying new parameters (Table 10 and Table 11) and together with BF filtering (Table 12 and Table 13).

Table 10. The RMSPE results when the Far-eastern index is used.

Far-eastern index	*AK*	*IS*	*GARANTI*	*BIST30*
1NN	2.523778	1.563303	1.743394	1.535997
2NN	2.102215	1.272339	1.935211	1.089865
3NN	2.024797	1.271858	1.88175	1.265789
4NN	1.577906	1.101763	1.699738	0.993911
5NN	1.369125	1.088723	1.361959	0.802536

Table 11. The RMSE results when the Far-eastern index is used.

Far-eastern index	*AK*	*IS*	*GARANTI*	*BIST30*
1NN	0.139754	0.084788	0.161343	0.252468
2NN	0.097526	0.067915	0.181596	0.155434
3NN	0.102322	0.072752	0.195592	0.16116
4NN	0.070979	0.058526	0.163982	0.158479
5NN	0.069592	0.054038	0.155418	0.147533

Table 12. The RMSPE results when the Far-eastern index is used with the Butterworth filter.

Far-eastern index+BW	*AK*	*IS*	*GARANTI*	*BIST30*
1NN	3.556565	2.436843	3.092161	2.686676
2NN	2.68649	2.299159	2.397482	2.276949
3NN	2.599607	2.501083	2.322024	2.262921
4NN	1.362777	0.83993	2.16978	0.900213
5NN	1.281084	0.71907	1.359017	0.762704

Table 13. The RMSE results when the Far-eastern index is used with the Butterworth filter.

Far-eastern index+BW	*AK*	*IS*	*GARANTI*	*BIST30*
1NN	0.195383	0.131386	0.277857	0.426849
2NN	0.14941	0.125319	0.216807	0.363872
3NN	0.144235	0.135966	0.237084	0.361765
4NN	0.068886	0.044471	0.197342	0.139435
5NN	0.060181	0.038127	0.168986	0.117781

4. Conclusion

This work introduces an improved PECNET machine learning algorithm that yields reliable and improved prediction performance for the closing stock price prediction model. The model has been enhanced by including the Butterworth filter in the proposed model. In addition, the spatial dimension has been included in the proposed model by constructing additional parameters. The constructed parameter contains the average normalized values of selected Far Eastern indices. Performance improvement of the proposed PECNET model is noticed by applying different filtering methods as well as including the additional parameters to the model.

References

1. *Kindig, B. (2020), "New age of stock market volatility driven by machines". Available at: https://www.forbes.com/sites/bethkindig/2020/04/10/new-age-of-stock-market-volatility-driven-by-machines/?sh=a53ca66dda21 (accessed 19 December 2022)*
2. *Fama, E.F. and French, K.R. (1993), "Common risk factors in the returns on stocks and bonds". Journal of financial economics, Vol. 33 No. 1, pp.3-56. Available at: https://doi.org/10.1016/0304-405X(93)90023-5 (accessed 19 December 2022)*
3. *Gorenc, N.M. and Velušček, D. (2016), "Prediction of stock price movement based on daily high prices". Quantitative Finance, Vol. 16 No.5, pp. 793-826. DOI: 10.1080/14697688.2015.1070960*
4. *Abu-Mostafa, Y.S. and Atiya, A.F. (1996), Introduction to financial forecasting. Applied intelligence, Vol. 6 No.3, pp.205-213. Available at: https://doi.org/10.1007/BF00126626 (accessed 19 December 2022)*
5. *Box, G., (2013), "Box and Jenkins: time series analysis, forecasting and control". In A Very British Affair. Palgrave Macmillan, London, pp.161-215.*
6. *Reddy, C.V. (2019), Predicting the stock market index using stochastic time series ARIMA modelling: the sample of BSE and NSE. Ind J Finance, Vol. 13 No.8, pp.7-25. Available at: http://dx.doi.org/10.2139/ssrn.3451677 (accessed 19 December 2022)*
7. *Khashei, M. and Bijari, M. (2010), An artificial neural network (p, d, q) model for timeseries forecasting. Expert Systems with applications, Vol. 37 No.1, pp.479-489. Available at: https://doi.org/10.1016/j.eswa.2009.05.044 (accessed 19 December 2022)*
8. *Hornik, K., Stinchcombe, M. and White, H. (1989), Multilayer feedforward networks are universal approximators. Neural networks, Vol. 2 No.5, pp.359-366. Available at: https://doi.org/10.1016/0893-6080(89)90020-8 (accessed 19 December 2022)*
9. *White, H. (1988), July. Economic prediction using neural networks: The case of IBM daily stock returns. ICNN Vol. 2, pp. 451-458. DOI: 10.1109/ICNN.1988.23959.*
10. *Hassan, M.R., Nath, B. and Kirley, M. (2007), A fusion model of HMM, ANN and GA for stock market forecasting. Expert systems with Applications, Vol.33 No.1, pp.171-180. Available at: https://doi.org/10.1016/j.eswa.2006.04.007 (accessed 19 December 2022)*
11. *Wang, J.J., Wang, J.Z., Zhang, Z.G. and Guo, S.P. (2012), Stock index forecasting based on a hybrid model. Omega, Vol. 40 No.6, pp.758-766. Available at: https://doi.org/10.1016/j.omega.2011.07.008 (accessed 19 December 2022)*
12. *Hajizadeh, E., Seifi, A., Zarandi, M.F. and Turksen, I.B. (2012), A hybrid modeling approach for forecasting the volatility of S&P 500 index return. Expert Systems with Applications, Vol. 39 No.1, pp.431-436. Available at: https://doi.org/10.1016/j.eswa.2011.07.033 (accessed 19 December 2022)*
13. *Shahi, T.B., Shrestha, A., Neupane, A. and Guo, W. (2020), Stock price forecasting with deep learning: A comparative study. Mathematics, Vol. 8 No.9, p.1441. Available at: https://doi.org/10.3390/math8091441 (accessed 19 December 2022)*
14. *Bao, W., Yue, J. and Rao, Y. (2017), A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one, Vol.12 No.7, p.e0180944. Available at: https://doi.org/10.1371/journal.pone.0180944 (accessed 19 December 2022)*
15. *Li, J., Bu, H. and Wu, J. (2017), June. Sentiment-aware stock market prediction: A deep learning method. 2017 international conference on service systems and service management. IEEE, pp. 1-6. DOI: 10.1109/ICSSSM.2017.7996306.*
16. *Goodfellow, I., Bengio, Y. and Courville, A. (2016), Deep learning. MIT press.*
17. *Ustundag, B.B. and Kulaglic, A. (2020), High-performance time series prediction with predictive error compensated wavelet neural networks. IEEE Access, Vol. 8, pp. 210532-210541. DOI: 10.1109/ACCESS.2020.3038724*
18. *Kulaglic, A. and Ustundag, B.B. (2021), "Stock price prediction using predictive error compensation wavelet neural networks". Comput. Mater. Continua, Vol. 68 No.3, pp. 3577-3593. DOI:10.32604/cmc.2021.014768.*
19. *Kulaglic, A. and Ustundag, B.B. (2021), "Predictive Error Compensating Wavelet Neural Network Model for Multivariable Time Series Prediction". TEM Journal, Vol.10 No.4. DOI: 10.18421/TEM104-61.*
20. *Ellis, G. (2012), Control system design guide: using your computer to understand and diagnose feedback controllers. Butterworth-Heinemann.*
21. *Moshiri, S. and Cameron, N. (2000), Neural network versus econometric models in forecasting inflation. Journal of forecasting, Vol.19 No.3, pp.201-217.*
22. *Patterson, D.W. (1996), Artificial Neural Networks: Theory and Applications, Prentice Hall.*

Corresponding Author
Ajla Kulaglic,
Istanbul Technical University,
Istanbul,
Turkey,
E-mail: kulaglic@itu.edu.tr