ISSN: 2147-5121 / E-ISSN: 2717-7610
Yayın Aralığı: Yılda 2 Sayı - Başlangıç: 2013

İstanbul Nişantaşı Üniversitesi
Sosyal Bilimler Dergisi

İstanbul Nişantaşı University
Journal of Social Sciences

Bilimsel Hakemli Dergi
Scientific Refereed Journal

Yıl: 2023 Cilt: 11 Sayı: Özel Sayı
Year: 2023 Vol.: 11 No: Special Issue

NUSBD

ARAŞTIRMA MAKALESİ / RESEARCH ARTICLE

DOI: 10.52122/nisantasisbd.1346658

EFFECT OF LINEAR DATA PROCESSING PROCESSES ON THE PREDICTION PERFORMANCE OF THE NEURAL NETWORK: AN APPLICATION WITH EXCHANGE RATE DATA

Dr. Öğr. Üyesi Ersin SÜNBÜL*

*İstanbul Nişantaşı Üniversitesi, Uygulamalı Bilimler Yüksek Okulu,
Uluslararası Ticaret ve İşletmecilik Program.

e-posta:

ersin.sunbul@nisantasi.edu.tr

ORCID 0000-0001-6187-2038

ABSTRACT

This study investigated the effect of traditional data preprocessing processes on the prediction performance of predictions made with artificial neural network models. For this purpose, two different models were created and time series estimation was made. The original data was used in the first model, and in the second model, the data obtained by the traditional data preprocessing method in the time series were used. The data set consists of monthly real US Dollar/Turkish Lira rates between 2000M1 and 2022M2 for Turkey. Jordan model with feedback artificial neural network architecture is used for time series estimation. Estimation errors were calculated according to the Root Squared Value of Mean Squared Error (RMSE) criteria and the results were discussed according to this statistic. In the study, it was concluded that data processing reduces the estimation error of the nonlinear method.

Keywords: Jordan feedback artificial neural network, Forecasting, Exchange rate, Time series.

Jel Codes: C22, C14, F37, F47, G17.

DOĞRUSAL VERİ İŞLEME SÜREÇLERİNİN SİNİR AĞININ TAHMİN PERFORMANSINA ETKİSİ: DÖVİZ KURU VERİLERİYLE BİR UYGULAMA

ÖZ

Bu çalışmada, geleneksel veri ön işleme süreçlerinin, yapay sinir ağları modelleri ile yapılan tahminlerin tahmin performansı üzerindeki etkisi incelenmiştir. Bu amaçla iki farklı model oluşturulmuş ve zaman serisi tahmini yapılmıştır. Birinci modelde orijinal veriler kullanılmış, ikinci modelde ise zaman serilerinde geleneksel veri ön işleme yöntemi ile elde edilen veriler kullanılmıştır. Veri seti, Türkiye'nin 2000A1 ile 2022A2 arasındaki aylık reel ABD Doları/Türk Lirası kurlarından oluşmaktadır. Zaman serisi tahmini için geri beslemeli yapay sinir ağı mimarisine sahip Jordan modeli kullanılmıştır. Tahmin hataları Ortalama Karesel Hatanın Kök Kare Değeri (RMSE) kriterlerine göre hesaplanmış ve sonuçlar bu istatistiğe göre tartışılmıştır. Çalışmada veri işlemenin doğrusal olmayan yöntemin tahmin hatasını azalttığı sonucuna ulaşılmıştır.

Anahtar Kelimeler: Jordan geri beslemeli yapay sinir ağı, Tahmin, Döviz kuru, Zaman serileri.

Jel Kodları: C22, C14, F37, F47, G17.

Geliş Tarihi/Received: 01.01.2020	Kabul Tarihi/Accepted: 01.01.2020	Yayım Tarihi/Printed Date: 30.06.2023

Kaynak Gösterme: Sünbül, E., (2023). “Effect of Linear Data Processıng Processes on the Predictıon Performance of the Neural Network: An Applicatıon with Exchange Rate Data”. *İstanbul Nişantaşı Üniversitesi Sosyal Bilimler Dergisi*, Özel Sayı(11) 33-49.

INTRODUCTION

The most important purpose of time series analysis is to have information about the future. Time series can be affected by both their own lagged values and external factors. For this reason, multivariate models are needed in addition to univariate models for future predictions. Linear methods have long been used in time series forecasting. Linear methods allow analysis with the assumption of normality. However, the nature of time series may not always satisfy the assumption of normality. Operations for normality can also cause missing data issues.

Developments in information technologies have contributed to the easier solving of complex mathematical operations and thus the development of non-linear forecasting methods. While linear methods make predictions based on assumptions, non-linear methods can make predictions based on data. These can be listed as Artificial Neural Networks (ANN), Fuzzy Time Series Methods, Fuzzy Inference Systems, and Hybrid Methods.

It is seen that financial time series generally contain non-linear data. While linear methods produce erroneous results in incomplete observations and non-stationary series, the high forecasting performance of ANN has increased interest in the field (Ağyar, 2015). ANNs topologically consist of artificial neurons and forward and backward connections between layers. Neurons change the strength of connections and encode information through networks between layers. Changing the weight values of the connections is defined as artificial learning (Magnitskii, 2001; Eryılmaz et al., 2014).

A simple ANN consists of the input layer, the hidden layers, and the output layer. What makes ANNs different from each other is their architectural structures and learning algorithms (Ağyar, 2015). The data received from the input layer to the system is transferred to the neurons via neural networks, and the information obtained as a result of a series of mathematical operations is sent to the outside world through the output layer. Information is obtained by passing the data between the output values and the input values through an activation function. Therefore, the activation function is extremely important for predicting success. These functions are; defined as a threshold, ramp, sigmoid and linear functions (Güneri and Apaydın, 2004: 174). ANN architecture can be configured with user intervention. The purpose of this intervention is to make the best guess.

ANNs are classified as Feed Forward Networks (FFN) and Feedback Networks (FN) according to the way the layers are connected. FFN transmits data applied to the input layer directly to the hidden layers. There is no data exchange between neurons in the same layer. The process is completed by transferring the data passing through the hidden layer to the output layer as information via neurons. In FN, on the other hand, while there is an FFN-like data transition, it is also seen that there is no backward data transition from both the intermediate layers and the output layer. The difference between the output data and the input data is used to improve the operation as an error. Interconnections can also be made between neurons in the intermediate layers. When the learning process is completed, the final output information is produced (Kakıcı, 2017; Elmas, 2018: 63; Güneri and Apaydın, 2004: 175).

ANN training takes place with a learning process. The learning process can be defined as supervised learning and unsupervised learning. The learning process begins with the introduction of the input and output values to the network. The weight coefficients are changed until the input and output values match. Thus, the error rate is tried to be reduced. In unsupervised learning, the output value is not introduced to the network. The network is expected to make the best classification using the input data. The weights are also changed to determine the result of this classification with the least error (Sabak and Başar, 2020; Güneri and Apaydın, 2004: 175).

The number of hidden layers and neurons can affect the learning time, increase the error, or cause over-learning. Therefore, the number of hidden layers and neurons should be kept low for optimum benefit. The number of variables in the input and output layers is an important factor in determining the number of hidden layers and neurons. However, there is no accepted rule for the optimum number in the literature. For this reason, in prediction studies with ANN, the best prediction architecture is determined by changing the number of neurons (Benli and Tosunoğlu, 2014). As the relationship between the input and output layers becomes more complex, the number of processing elements (neurons) in the hidden layer must also be increased. If the process can be divided into stages, the number of hidden layers should be increased. However, a single hidden layer should be used for single-process problems. Too many hidden layers can cause data to be cached. The amount of training data defined should be set as the upper limit for the number of rendering items in the hidden layer. That is, the number of variables in the input should be the upper limit for the number of neurons in the hidden layer (Elmas, 2018: 73-74). ANN models consisting of three-layer, simple structure, feed-forward, and supervised learning algorithms are more preferred in time series forecasting problems (Eğrioğlu and Aladağ, 2005: 2).

None of the described methods guarantee the best performance for ANN forecasting. The best result can be found by trial and error (Tosunoğlu and Benli, 2012). How well the relationship between the data is learned is controlled by the performance criterion. There are many methods for performance measurement in the literature. However, in this study, Equation 1 was used to calculate the forecast performance RMSE (Eğrioğlu and Baş, 2020: 61-62).

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{ntest}}\sum_{t=1}^{\mathrm{ntest}}(x_t - \hat{}x_t)^2} \qquad (1)$$

Gunay et al., (2007: 131) defined the time series forecasting process with ANN in six stages. (i) The observation values of the input data should be brought to the range (0, 1) by normalization. (ii) Data should be divided into training, validation, and testing data in the organization process. At this stage, it should be decided in which group the observation values will be included as a percentage. However, it is seen in the literature that it is generally divided into two training and test data. For this reason the study, the data is divided into two parts. (iii) In the modeling process, network architecture, activation function, learning algorithm, parameters, and performance measurement method are determined. (iv) Appropriate weight values are calculated for the best performance in the model improvement process. (v) The performance of the network is calculated according to the measurement criteria selected in the performance measurement process. (vi) As a result, the best model is decided.

When the literature is examined, it is seen that the purpose of the estimation of time series is to reduce the estimation error and even for this purpose, different estimation methods are compared with each other. However, such an aim was not pursued in this study.

This study aims to investigate the effect of data preprocessing on ANN methods that can make predictions without the need for data preprocessing in the literature. For this reason, within the scope of the preprocessing of the USD/TL rate data used in the study, seasonality research, which is one of the linear econometric time series analysis methods, and data cleaning operations such as seasonal correction, unit root analysis, and stationarisation proses were performed.

Two different ANN models were established for the study and their forecasting performances were compared. For the first model, the original observation values of the USD/TL exchange rate were used, and for the second model, the data set with data preprocessing was used. In this respect, the study contributes to the literature.

In the introduction part of the study, linear and non-linear analysis methods used in the forecasting of time series are discussed, in the second part, the exchange rate forecasting literature is examined. In the third part, the method and application of the study are given. In the conclusion part, the obtained findings are discussed.

1. Literature Review

It is possible to come across many linear and non-linear methods and time series forecasting studies in the literature. In addition, studies are comparing both methods. In recent years, hybrid approaches have been seen in which both methods are used together. Zhang and Hu (1998), have developed different models by changing input values, the number of neurons, and data sets in ANN architecture and investigated the effects of these changes on forecasting performance. Zhang et al., (1998) made a comprehensive review of ANN models used for time series forecasting. When these studies are examined, it is seen that non-linear methods are more successful. Hybrid studies, on the other hand, attract the attention of researchers to perfect their forecasting performance. Arisanti and Puspita (2022), preferred the Non-linear Autoregressive Extrinsic Neural Network Method, "rprop+" learning algorithm, and feedforward neural network model to determine the best USD/rupiah exchange rate, and forecasting model.

Weigend et al., (1992) used ANN and Random Walk Models to predict the Deutsche mark/USD exchange rate. Kuan and Liu (1995), used the ANN, and ARIMA models to forecasting the British pound/USD, Canadian dollar/USD, German mark/USD, Japanese yen/USD, and Swiss franc/USD rates. Kamruzzaman and Sarker (2003), used ANN, and ARIMA models to predict 6 different cross-exchange rates with the Austrian dollar. Taş et al., (2018) have used ARIMA, and ANN models to predict the Euro/Turkish lira exchange rate. Kaynar and Taştan (2009a), used Box-Jenkins, and Multilayer ANN Models to predict the Turkish lira/USD exchange rate. Panda et al., (2022) on the other hand, used ARIMA, Multilayer Perceptron (MLP), Linear Regression (LR), Random Forest Regression Layer (RFRL) and Convolutional Forecasting Neural Network Models (CNN). In these studies, it has been found that the non-linear models outperformed the others.

Zhang (2003), has proposed an ANN/ARIMA hybrid forecasting model, using the USD/British pound exchange rate between 1980-and 1993. Altan (2008), has proposed an ANN/Vector Autoregression (VAR) hybrid forecasting model, using the USD/TL exchange rate between 1987M1 to 2007M9. On the other hand, Sünbül (2022), has forecasted the USD/TL exchange rate using the hybrid approach of ANN and ANN/VAR. In addition, he has compared the forecasting performance of his proposed Multi-Stage Data Manipulation (MSDM)/ANN hybrid model with others. He has proven to improve the prediction performance of MSDM.

2. Methodology and Application

Turkey's monthly real US Dolar (USD)/Turkish lira (TL) exchange rate data from 2000M1 to 2022M2 has been used for the study. USD/TL exchange rate index data consists of 266 observations. The data used in the study were taken from EDDS, (https://evds2.tcmb.gov.tr/, Access date: 10.01.2022). In the study, seasonality and stationarity analysis from linear econometric time series analysis methods and multilayer backpropagation from non-linear analysis methods, and ANN architecture were used. Webel & Ollech Test (wo) was preferred for seasonality control. The "wo" function from the "seastests" library in R-Studio was used for testing (Ollech and Webel, 2020). Augmented Dickey-Fuller (ADF) Test was preferred for stability control (Dickey and Fuller, 1981). The "ur.df" function from the "urca" library in R-Studio was used for testing (Hamilton, 1994). Jordan's model is preferred for non-linear time-series forecasting. For testing, the "Jordan" function from the Stuttgart Neural Network Simulator (SNNS, 2020). "SNNS" library in R-Studio was used (Jordan, 1986a).

2.1. Seasonality Analysis

Seasonal decomposition in linear time series analysis is an important step for further analysis. Seasonality tests can be done to eliminate the possibility of spurious regressions and to obtain accurate results. The seasonality graph of the original data is presented in Figure 1.

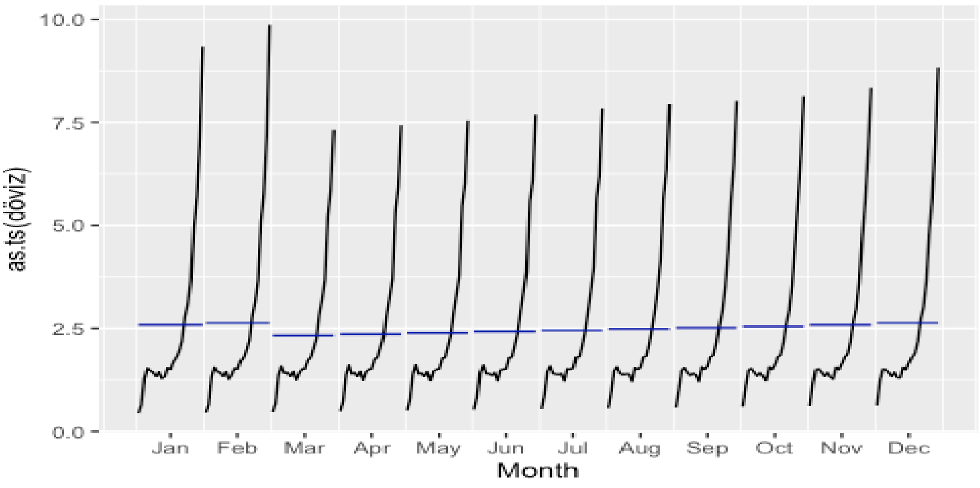


Figure 1. Seasonality Graph

Figure 1 shows the monthly fluctuations in the variable and the averages of these fluctuations. The extreme variation of the horizontal blue lines can qualitatively determine that there may be seasonality for the respective period. According to the qualitative evaluation, it has been thought that there is no seasonality in the series. Seasonality has been also statistically confirmed by the "wo" test. The test statistics have been presented in Table 1.

Table 1. Seasonality Test Statistics

Test used: "**wo**"			
Test statistic:		0	
P-value:	0.9862	1	0.9952

When Table 1 is examined, it is seen that the test statistic is calculated as "0". This result shows that there is no seasonality in the series. It is expected that the p values calculated in the non-seasonal time series are close to "1". According to the table, it is seen that the said values are 0.9862, 1, and 0.9952. Therefore, according to the test results, it has been proven that there is no seasonality in the series.

2.2. Stationarity Analysis

For further analysis in time series, the series must be stationary. Research can offer clues about non-static data and unstable datasets in datasets. The problem is also defined as the misregression problem in the literature. Therefore, the data should be investigated first whether they are static or not. In this context, the stability of the data used in the study was tested with the ADF test (Dickey & Fuller, 1981). If the series is constant and its variance does not change over time, the stationarity decision can be taken for this series.

By applying the test to the original data, a stationary decision at the I(0) level can be made for the series that satisfies the assumption. A constant term and trend can be added to the model established for testing, or a model can be written in which both are not used. It can be said that the stationarity assumption is provided in the series that provides the assumption for any model.

The process should be repeated by taking the difference of one degree "I(1)" of the non-stationary series. It is possible to repeat the difference process until the stationarity assumption is satisfied.

The null hypothesis for unit root research (for ADF) and model significance is presented below;

- For unit root research (for ADF);

H0= Series is not stationary, (F-p value <0.01, 0.05, 0.1 significance level),

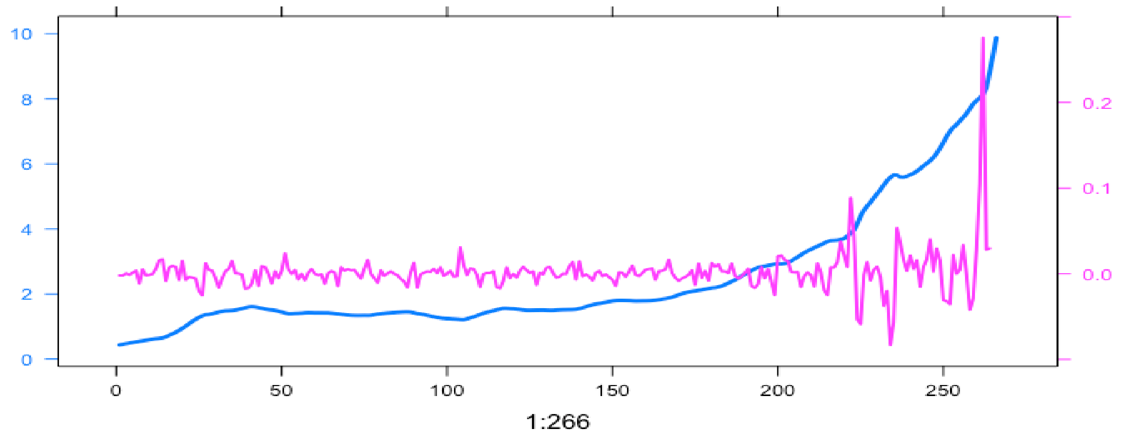
- For the significance of the model;

H0= Model is significant, (Test statistic <significance levels). Test results are presented in Table 2.

Table 2. Stability Test Results

ADF Unit Root Test for Exchange Rate (exc)				
Stability Degree	Indicator	Model without trend and a constant term	Model of the constant term	Model of trend and the constant
I (0)	Coefficients: Pr	0.22	0.007 **	0.0000***
	F- *p*-value:	0.000	0.000	0.000
	Test-statistic:	4.12	4.702	4.04
I (1)	Coefficients: Pr	0.3498	0.7142	0.0036**
	F- *p*-value:	0.000	0.000	0.000
	Test-statistic:	1.34	2.57	2,85
I (2)	Coefficients: Pr	**0.0146***		
	F- *p*-value:	**0.000**		
	Test-statistic:	**-6,482**		
tau3:	**-3.98**	-3.42	-3.13	

Since the calculated test statistic value is -6.482<-3.98, the series is stationary. The model is significant as P statistical value = 0.01460 * and significance level = 0.000 <0.05. The time series plots of the original data and the differential data are presented in Figure 2.



* In Figure 2, the blue color shows the time-series graph of the original series, while the red color shows the time-series graph of the different series.

Figure 2. Graph of the Original and Differential Series

2.3. Non-Linear Forecasting

Jordan's model is preferred for non-linear time-series predictions. This function uses multi-layer ANN architecture. In this architecture, there is an additional neuron that is not found in traditional ANNs and is defined as the state layer. The task of the state layer in architecture is to provide feedback from the output layer to the input layer. Learning algorithm in function Rumelhart et al., (1986) neurons are interconnected by a parallel distributed processing network, weighted and one-way connections. The activation function is calculated by Equation (5).

$$xj = \phi(\textstyle\sum_{i=1}^{n} wji\, xi + \theta j) \tag{5}$$

In the equation;

(xi) is the activation of unit (i).

(wji) is weighted from i to (j) units.

(θj) is the deviation associated with the unit (j).

(n) is the number of neurons in the neural network.

The expression in parentheses is defined as the net input of a neuron. The algorithm of the model contributes to the optimization of the weights in the hidden layer by spreading the calculated errors in the learning process to the output layer. Thus, it is aimed to minimize errors. The main purpose of the algorithm is; is to provide a loop that aims to reach the activation value. An error signal is generated at each output unit when the desired output is compared with the actual output. Error signals are sent back to the hidden units and errors are calculated (Jordan, 1986a).

Rumelhart et al., (1986) proved that the weight determined by the proposed algorithm varies proportionally with the partial derivative according to the weight of the sum of the squared error in the output units. Theoretically, it is possible to find the most suitable model according to algorithm theory by changing the structure of recurrent connections in the network (Jordan, 1986b). There are three basic layers in Jordan's ANN architecture: input, hidden, and output. During learning, there is a backward propagation from the output layer. There is also permeability between the input variables in the input layer. A three-layer architecture is used for the ANN Model as input layer, hidden layer, and output layer. For the best architecture, the number of neurons in the hidden layer was increased one by one and the model with the best prediction was determined.

Statistical data in the study belong to the best models. Other statistics were not included in the study. Two different data sets were obtained for prediction with ANN and two different ANN predictions were made. For Model 1, the original observation values were determined as the dependent variable, and the values with one, four, and twelve lags were determined as the independent variables. While the original observation values of the exchange rate were determined as the dependent variable for Model 2, the one, four, and twelve-lagged values of the stationary series were determined as the independent variables. The characteristics of the data set were taken into account while determining the delay intervals of the independent variables. Since the data set has monthly observations, it has been evaluated that it may be related to the data of the previous month. While choosing the four lagged values, it was evaluated that there might be seasonal differences in the series. When choosing twelve lagged values, it was evaluated that the series might be related to the data from one year ago. Finally, the contribution of the proposed model to the prediction success of nonlinear methods was examined by comparing the error rates of both models.

In the normalization process, the observation values of the variables were brought to the range (0, 1). During the organization process, the variables were divided into 70% training (201 observations) and 30% test (53 observations) data. Multilayer Jordan ANN architecture and Rumelhart learning algorithm were chosen in the modeling process. During the model improvement process, the number of neurons in the hidden layer was increased from 1 to 5 one by one and the error coefficients were examined each time. At this stage, the optimal weights of both models were determined. In the performance measurement process, the architecture with the best prediction was determined. The statistical values and data visuals given in the study also consist of the data of the best models determined. During the evaluation of the results, the best forecasting performances were discussed with the help of the summary table. The model architectures of the best models are presented in Figure 3 and Figure 4.

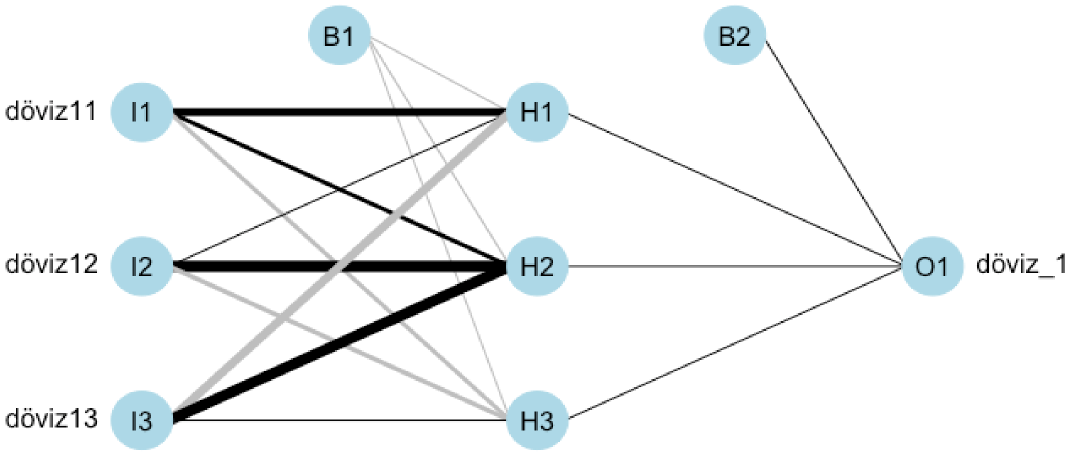


Figure 3. The Best Model Architecture for Model 2

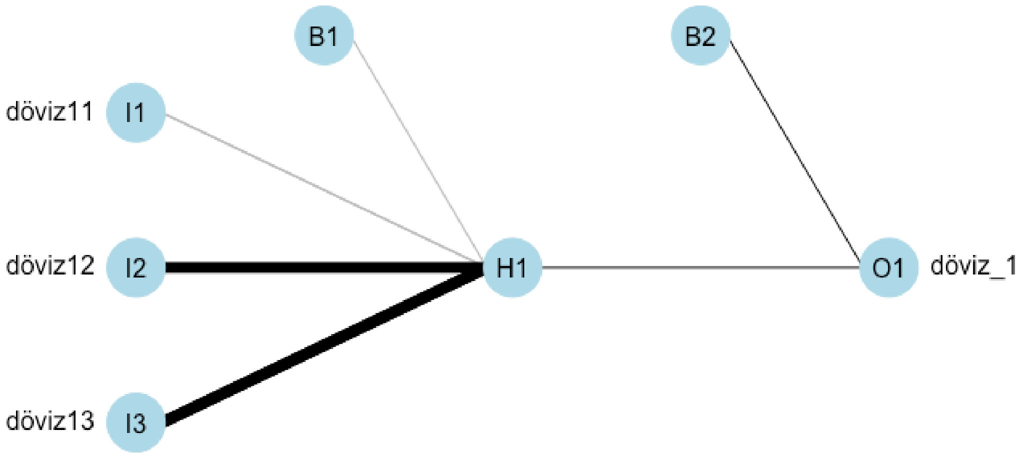


Figure 4. The Best Model Architecture for Model 1

In Figure 3 and Figure 4, "*I*" defines the input layer, "*H*" defines the hidden layer, "*O*" defines the output layer and "*B*" defines the Jordan model side layer. The connections between the layers represent the artificial neural network. The optimal weight values calculated for the models that make the best forecasting are presented in Table 3.

Table 3. Optimal Weight Values for the Best Models

Model 2

	inp1	inp2	inp3	hid1	hid2	hid3	out1	con1
inp1	0	0	0	0.212	-11.298	12.249	0.00	0.0
inp2	0	0	0	-0.523	11.283	-15.369	0.00	0.0
inp3	0	0	0	-0.523	-13.729	0.317	0.00	0.0
hid1	0	0	0	0.000	0.000	0.000	-0.81	0.0
hid2	0	0	0	0.0000	0.0000	0.0000	17.318	0.0
hid3	0	0	0	0.0000	0.0000	0.0000	17.362	0.0
out1	0	0	0	0.0000	0.0000	0.0000	0.0000	1.0
con1	0	0	0	-38.102	0.9423	0.4860	0.0000	0.3

Model 1

	inp1	inp2	inp3	hid1	out1	con1
inp1	0	0	0	76.855	0.000	0.0
inp2	0	0	0	37.017	0.000	0.0
inp3	0	0	0	-18.624	0.000	0.0
hid1	0	0	0	0.000	0.455	0.0
out1	0	0	0	0.000	0.000	1.0
con1	0	0	0	0.748	0.000	0.3

Figure 5 shows the learning process for Model 1 and Model 2.

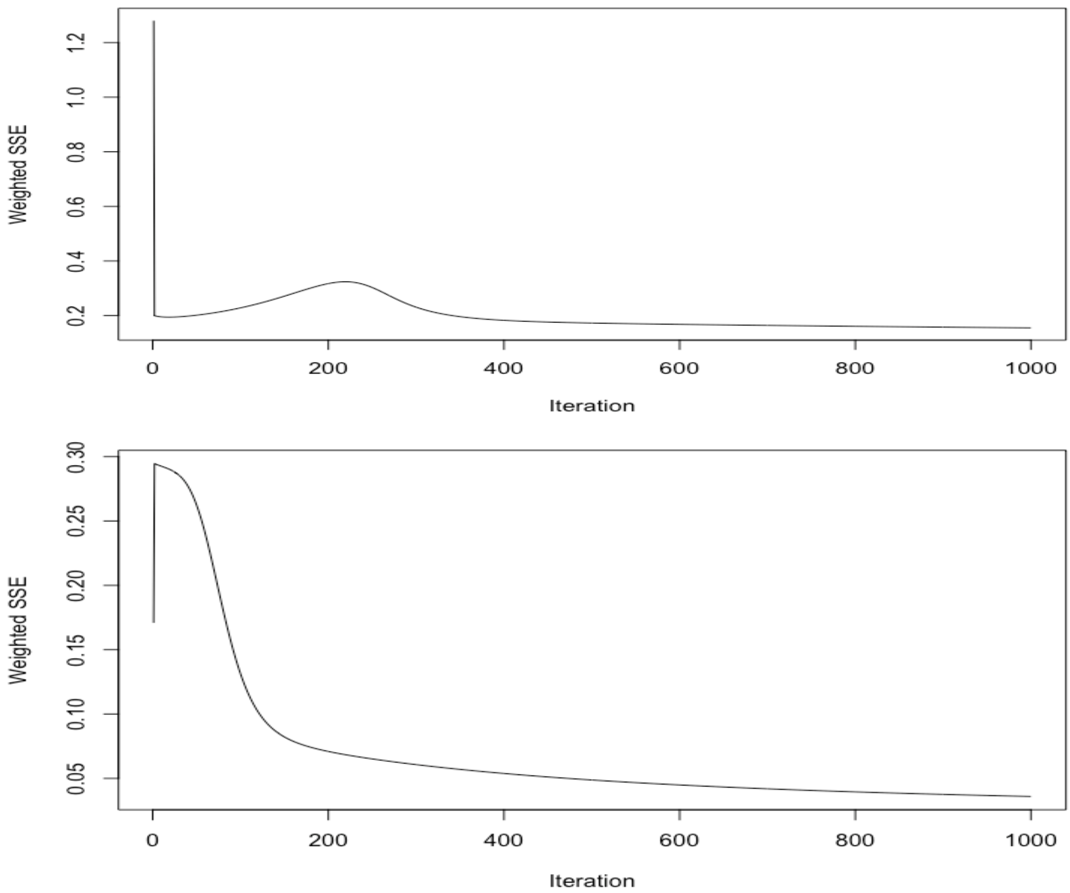


Figure 5. Learning Process Graph (Model 2/ Model 1)

According to Figure 5, it is understood that the learning process continues for 1000 iterations for both models and complete learning is realized in the 600th iteration. Therefore, it is seen that the upper limit of 1000 iterations determined for the models is sufficient. The forecasting errors obtained for the best models are presented in Table 4.

Table 4. The Forecasting Errors Obtained for the Best Models

Models	**Error Measurement Criteria**	**Number of Neurons**
	RMSE	
Model 2	2.349	1
	2.204	2
	2.194	3
	2.245	4
	2.232	5
Model 1	**2.958**	1

	4.010	2
	4.153	3
	4.282	4
	3.827	5

When Table 4 is examined, it is seen that the architecture with 1 neuron in the hidden layer is the best model for Model-1 according to the RMSE error measurement criterion. The RMSE value for Model 1 was calculated as 2.958. The error in question proved that the model made approximately 96% accurate forecasting. It is seen that the 3-neuron architecture in the hidden layer is the best model for Model-2. The RMSE value for Model 2 was calculated as 2.194. The error in question proved that the model made approximately 97% accurate forecasting. The ANN architecture created for both models is summarized in Table 5.

Table 5. Summary Model Architecture

The Best Model	Error Measurement Criteria	Architecture of the Model							Learning Algorithm	Architecture of the INN
	RMSE	Layers			Number of Hidden Layers Neurons	Training Data	Test Data	Learning Iteration		
		Input	Hidden	Output						
Model 2	2,194	3	1	1	3	70%	30%	1000	Rumelhart at al.	Back Propagation Jordan Architect
Model 1	2,958	3	1	1	1					

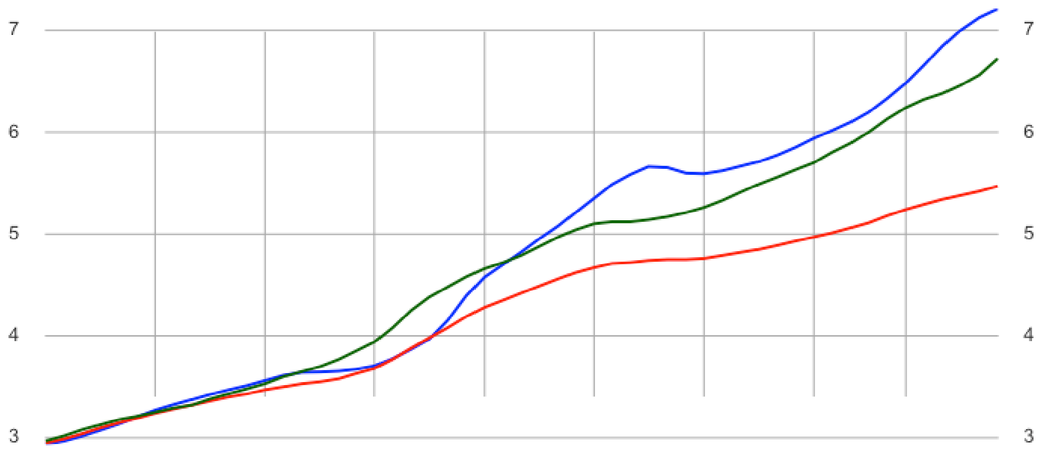
When Table 5 is examined, there are three variables in the input layer of Model 2 and 3 neurons in the hidden layer, and Model 1 has three variables in the input layer and 1 neuron in the hidden layer. The data set for both models is divided into 70% (201 observations) training data and 30% (53 observations) test data. 1000 iterations were determined for learning, Jordan architecture and Rumelhart learning algorithms were preferred. According to the RMSE criterion, the error rate of Model 2 was calculated as 2,194, and the error rate of Model 1 was calculated as 2,958. The forecasted and original observation values for the best models are presented in Table 6.

Table 6. Forecasting Results

Forecasting Results for Model 2												
Year/ month	1	2	3	4	5	6	7	8	9	10	10	11
2016										2.95	2.99	3.04
2017	3.10	3.15	3.19	3.24	3.28	3.32	3.36	3.40	3.43	3.47	3.50	3.53

2018	3.55	3.58	3.63	3.68	3.77	3.88	3.98	4.08	4.19	4.28	4.35	4.42
2019	4.49	4.56	4.62	4.67	4.71	4.72	4.74	4.75	4.75	4.76	4.79	4.82
2020	4.85	4.89	4.93	4.97	5.01	5.06	5.11	5.18	5.24	5.29	5.34	5.38
2021	5.42	5.47										
Forecasting Results for Model 1												
2016										2.97	3.02	3.08
2017	3.13	3.18	3.21	3.25	3.29	3.32	3.38	3.43	3.48	3.53	3.60	3.65
2018	3.70	3.77	3.85	3.94	4.08	4.25	4.38	4.48	4.58	4.66	4.72	4.79
2019	4.88	4.97	5.04	5.10	5.12	5.12	5.14	5.17	5.21	5.26	5.33	5.41
2020	5.49	5.56	5.63	5.70	5.80	5.89	6.00	6.13	6.24	6.32	6.38	6.46
2021	6.56	6.72										
Original Observation												
2016										2.94	2.97	3.02
2017	3.08	3.14	3.20	3.27	3.32	3.37	3.42	3.47	3.51	3.56	3.61	3.64
2018	3.65	3.65	3.67	3.71	3.78	3.87	3.97	4.15	4.39	4.58	4.70	4.82
2019	4.96	5.08	5.21	5.35	5.49	5.58	5.66	5.65	5.60	5.59	5.62	5.67
2020	5.71	5.78	5.85	5.94	6.01	6.10	6.20	6.33	6.48	6.65	6.84	7.00
2021	7.12	7.21										

When Table 6 is examined, it is seen that 53-period forecasting is made. The original observation values for the same period are also included in the table. The time path graph obtained for the forecasting and original observation values made by both models is presented in Figure 6.



* The red time path graph shows the Model 1 forecast statistics, the green time path graph shows the Model 2 forecast statistics, and the blue colored time path graph shows the original observation values.

Figure 6. Time Path Plot for Forecasting Observations

When Figure 6 is examined, it is seen that Model 2 (variables with data preprocessing) is realized closer to the original observations. The longer the observation interval for Model 1, the more clearly the deviation from the original observation is noticeable. On the other hand, it is understood that the forecasting made with Model 2 do not differ from the original observation over time and are quite accurate.

CONCLUSION

The most important purpose of time series analysis is to predict the future. Real-world data consists of linear and non-linear patterns with different components. Therefore, they are difficult to model. Three methods have been proposed for time series forecasting in the literature (Eroğlu et al., 2019). These; are statistical inferences based on probability (linear methods), inferences with ANN, and inference systems based on fuzzy logic (FL) clusters. Combining datasets with different parameters seems insurmountable with linear methods, while non-linear methods can easily solve this problem.

This study aims to investigate the effect of linear data preprocessing on the forecasting performance of the non-linear method in time series. For the study, Turkey's monthly real US Dollar/Turkish lira exchange rate data from 2000M1 to 2022M2 was used.

Within the scope of linear analysis, it was observed that there was no seasonality in the series. Again, in the stationarity analysis, it was determined that the series were not stationary at the I (0) level and at the I (1) difference. However, the data provided the stationarity assumption for I(2).

In accordance with the purpose of the study, two different ANN models were created. In the first model, a four-variable data set consisting of the original observation, one, four, and 12 lagged values of the original observation was obtained. In the second model, a four-variable data set consisting of the original observation, one, four, and 12 delay values of the processed data was obtained. For both models, the original observation was defined as the dependent variable and the others as the independent variable. At the end of the study, the forecasting performances of the two models were compared. The data set for both models is divided into 70% (201 observations) training data and 30% (53% observations) test data. 1000 iterations were determined for learning, Jordan architecture and Rumelhart learning algorithms were preferred.

According to the RMSE measurement criterion, it is seen that the architecture with 1 neuron in the hidden layer is the best model for Model-1. The RMSE value for Model 1 was calculated as 2,958. The error in question proved that the model predicted approximately 96% correct. For Model-2, it is seen that the architecture with 3 neurons in the hidden layer is the best model. The RMSE value for Model 2 was calculated as 2,194. The error in question proved that the model made approximately 97% accurate forecasting. As a result, it was seen that the data preprocessing process had a positive effect on the forecasting performance with ANN.

REFERENCES

Ağyar, Z., (2015). "Yapay Sinir Ağlarının Kullanım Alanları ve Bir Uygulama", *Mühendis ve Makine,* 56(662), 22-23.

Altan, Ş., (2008). "Döviz Kuru Öngörü Performansı İçin Alternatif Bir Yaklaşım: Yapay Sinir Ağı", *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi,* 10(2), 141-160.

Arisanti, R., and Puspita, M.D., (2022). "Non-linear Autoregressive Neural Network with Exogenous Variable in Forecasting USD/IDR Exchange Rate", *Commun. Math. Biol. Neurosci.*

Benli, Y.K., and Tosunoğlu, N.G., (2014). "Avrupa Birliği Üye Ülkelerin Morgan Stanley Capital International Endekslerinin Değerlendirilmesi ve Yapay Sinir Ağları ile Öngörüsü", *Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi,* 16(2), 72-87.

Dickey, D. A., and Fuller, W.A., (1981). "Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root", *Econometrica,* 49(4), 1057-1072. https://doi.org/10.2307/1912517

EDDS., (2022). https://evds2.tcmb.gov.tr/, (Date of access: 10.01.2022).

Eğrioğlu, E., and Aladağ, Ç.H., (2005). "Yapay Sinir Ağları ve ARİMA Modellerin Melez Yaklaşımı ile Zaman Serilerinde Öngörü", *İstanbul Üniversitesi VII. Ulusal Ekonometri ve İstatistik Sempozyumu,* 26-27.

Eğrioğlu, E., and Baş, E., (2020). *Zaman Serileri ve Öngörü Yöntemleri (R Uygulamalı).* Ankara: Nobel Akademik Yayıncılık.

Elmas, Ç., (2018). *Yapay Zekâ Uygulamaları.* Ankara: Seçkin Yayınları.

Eryılmaz, O., Özer, M., and Yumuşak, N., (2014). "Impact of Small-World Topology on the Performance of a Feed-Forward Artificial Neural Network Based on 2 Different Real-Life Problems.", *Turk. J. Elec. Eng. & Comp. Sci,* 22, 708-718.

Günay, S., Eğrioğlu, E., and Aladağ, H., (2007). *Tek Değişkenli Zaman Serileri Analizine Giriş.* Ankara: Hacettepe Üniversitesi Yayınları.

Güneri, N., and Apaydın, A., (2004). "Öğrenci Başarılarının Sınıflandırılmasında Lojistik Regresyon Analizi ve Sinir Ağları Yaklaşımı", *Ticaret ve Turizm Eğitim Fakültesi Dergisi,* 1, 170-188.

Hamilton, J., (1994). *Time Series Analysis.* Princeton University Press.

Jordan, M,I., (1986a). "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine", *In Proceedings of the Eighth Annual Conference of the Cognitive Science Society,* Hillsdale NJ. Erlbaum, 531-546.

Jordan, M.I., (1986b). "Serial Order: A Parallel Distributed Processing Approach", Technical Rep Ort Nr. 8604, *Institute for Cognitive Science, University of California,* San Diego, La Jolla, California.

Kakıcı, A., (2017). "Yapay Sinir Ağlarının Katmanları ve Sınıflandırılması", https://yapayzeka.ai/yapay-sinir-aglarinin-katmanlari-ve-siniflandirilmasi-3/ (Erişim Tarihi: 26.11.2020).

Kamruzzaman, J., and Sarker, R.A., (2003). "Forecasting of Currency Exchange Rates Using ANN: A Case Study. In Neural Networks and Signal Processing", *Proceedings of the 2003 International Conference on,* 1, 793-797. IEEE.

Kaynar, O., and Taştan, S., (2009a). "Zaman Serileri Tahmininde ARIMA-MLP Melez Modeli", *Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi,* 23(3), 141-149.

Kuan, C. M., and Liu, T., (1995). "Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks", *Journal of Applied Econometrics*, 10(4), 347-364.

Magnitskii, N.A., (2001). "Some New Approaches to the Construction and Learning of Artificial Neural Networks", *Computational Mathematics and Modeling*, 12, 293-304.

Ollech, D., and Webel, K., (2020). "A Random Forest-based Approach to Identifying the Most Informative Seasonality Tests", *Deutsche Bundesbank's Discussion Paper Series*, 55.

Panda, M.M., Panda, S.N., and Pattnaik, P.K., (2022). "Forecasting Foreign Currency Exchange Rate using Convolutional Neural Network", (IJACSA) *International Journal of Advanced Computer Science and Applications,* 13(2), 607. www.ijacsa.thesai.org

Rumelhart, D.E., Hınton, G.E., and Williams, R.J., (1986). "Learning İnternal Rrepresentations by Error Propagation" *(Tech. Rep. 8506),* La Jolla: University of California, Institute for Cognitive Science, San Diego.

Sabak, S., and Basar, D.B., (2020). "Enflasyonu Etkileyen Faktörlerin Ülkeler Bazında Yapay Sinir Ağları Analizi ile İncelenmesi". *Working Paper Series,* 1(1), 27-38, Doi: 10.5281/zenodo.4387930.

SNNS., (2020). "Stuttgart Neural Network Simulator", User Manual Version 4.2. http://www.ra.cs.uni-tuebingen.de/SNNS/welcome.html.

Sünbül, E., (2022). Finansal Para Krizi Göstergesi Olarak Döviz Kuru ve Kura Etki Eden Değişkenlerin Belirlenmesi: Konvansiyonel Ekonometrik Zaman Serisi Analizi ve YSA ile Melez Bir Model Önerisi, (Yayımlanmamış) Doktora Tezi, Ankara: Ankara Hacı Bayram Veli Üniversitesi, Lisansüstü Eğitim Enstitüsü.

Tas, O., Yakak, E., and Uğurlu, U., (2018). "Euro/TL Kuru Tahmininde İstatistik ve Yapay Sinir Ağları Kullanımı", *Press Academia Procedia,* 7(1), 414-417. DOI: 10.17261/Pressacademia.2018.926.

Tosunoğlu, N., and Benli, Y.K., (2012). "Morgan Stanley Capital International Türkiye Endeksinin Yapay Sinir Ağları ile Öngörüsü", *Ege Akademik Bakış,* 12(4), 541-547.

Weigend, A.S., Huberman, B.A., and Rumelhart, D.E., (1992). "Predicting Sunspots and Exchange Rates with Connectionist Networks", *PRE-33772*.

Zhang, G.P., and Hu, M.Y., (1998). "Neural Network Forecasting of the British Pound/US Dollar Exchange Rate", *Omega,* 26(4), 495-506.

Zhang, G.P, Patuwo, B.E., and Hu, M.Y., (1998). "Forecasting with Artificial Neural Networks: The State of the Art", *International Journal of Forecasting*, 14(1), 35-62.

Zhang, G.P., (2003). "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model", *Neurocomputing,* (50), 159-175.

EXTENDED ABSTRACT

GENİŞLETİLMİŞ ÖZET

DOĞRUSAL VERİ İŞLEME SÜREÇLERİNİN SİNİR AĞININ TAHMİN PERFORMANSINA ETKİSİ: DÖVİZ KURU VERİLERİYLE BİR UYGULAMA

Giriş ve Çalışmanın Amacı (Introduction and Research Purpose): Bu çalışmada, zaman serilerinde durağanlık, mevsimsel bileşenlerin belirlenmesi ve söz konusu etkilerin seriden ayrıştırılması ve değişkenler arasında nedensellik, eşbütünleşme gibi geleneksel veri ön işleme süreçleri sonrasında maniple edilmiş verilerin, yapay sinir ağları modelleri ile yapılan tahminlerin tahmin performansı üzerindeki nasıl bir etkisinin olduğunu araştırmayı amaçlamaktadır.

Kavramsal/kuramsal çerçeve (Literature Review): Literatürde doğrusal (lineer) ekonometrik zaman serisi tahminlemeleri için serilerin durağan olmaları ön koşullardan birisidir. Durağan olmayan serilerle yapılan çalışmalarda değişkenler arasında aslında olmayan ilişkilere yönelik bulgulara ratlanabilmektedir. Bu durum literatürde sahte regresyon sorunu olarak ta yer almaktadır. Buna karşın doğrusal olmayan zaman serisi öngörülerinde (yapay sinir ağlarıyla öngörü) serilerin ön işlemesine ihtiyaç duyulmamasına rağmen durağanlaştırılan serilerle doğrusal olmayan zaman serisi tahminlerinde daha yüksek doğrulukta sonuçlar üretip üretmediği sınanmıştır.

Yöntem ve Bulgular (Methodology and Findings): Çalışmanın amacına yönelik olarak iki farklı YSA omodeli luşturulmuş ve zaman serisi tahmini yapılmıştır.Birinci modelde orijinal veriler kullanılmış, ikinci modelde ise zaman serilerinde geleneksel veri ön işleme yöntemi ile elde edilen veriler kullanılmıştır. Veri seti, Türkiye'nin 2000A1 ile 2022A2 arasındaki aylık reel ABD Doları/Türk Lirası kurlarından oluşmaktadır. Zaman serisi tahmini için geri beslemeli yapay sinir ağı mimarisine sahip Jordan modeli kullanılarak tahminlemeler yapılmıştır.

Sonuç ve Öneriler (Conclusions and Recommendation):

Tahmin hataları Ortalama Karesel Hatanın Kök Kare Değeri (RMSE) kriterlerine göre hesaplanmış ve sonuçlar bu istatistiğe göre tartışılmıştır. Çalışmada veri işlemenin doğrusal olmayan yöntemin tahmin hatasını azalttığı sonucuna ulaşılmıştır. Her ne kadar literatürde doğrusal olmayan tahmin problemlerinde orijinal veriler ile tahminleme yapılabileceği söylense de geleneksel veri işleme süreçlerinin tahmin hatalarını azalttığı bu çalışmanın sonuçlarına göre ortaya konulmuştur. Yapay sinir ağı gibi doğrusal olmayan yöntemleri kullanan araştırmacı ve sektör profesyonellerinin benzer yöntemleri kullanabilecekleri söylenebilir.

KATKI ORANI BEYANI VE ÇIKAR ÇATIŞMASI BİLDİRİMİ

Sorumlu Yazar *Responsible/Corresponding Author*	Ersin Sünbül
Makalenin Başlığı *Title of Manuscript*	EFFECT OF LINEAR DATA PROCESSING PROCESSES ON THE PREDICTION PERFORMANCE OF THE NEURAL NETWORK: AN APPLICATION WITH EXCHANGE RATE DATA
Tarih *Date*	19.08.2023
Makalenin türü (Araştırma makalesi, Derleme vb.) Manuscript Type (Research Article, Review etc.)	Araştırma Makalesi

Yazarların Listesi / *List of Authors*

Sıra *No*	**Adı-Soyadı** *Name - Surname*	**Katkı Oranı** *Author Contributions*	**Çıkar Çatışması** *Conflicts of Interest*	**Destek ve Teşekkür (Varsa)** *Support and Acknowledgment*
1	Ersin Sünbül	%100	Çıkar Çatışması Yoktur	Yoktur