Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Research Article

Stock Closing Price Prediction with Machine Learning Algorithms: PETKM Stock Example In BIST

Şevval TOPRAK [a,*], Gültekin ÇAĞIL [b] Abdullah Hulusi KÖKÇAM [b]

[a] *Department of Industrial Engineering, Institute of Natural Sciences, Sakarya University, Sakarya, TURKEY*

[b] *Department of Industrial Engineering, Faculty of Engineering, Sakarya University, Sakarya, TURKEY*

** Corresponding author's e-mail address: sevval.toprak@ogr.sakarya.edu.tr*

DOI: 10.29130/dubited.1096767

ABSTRACT

This study predicts the stock price of Petkim Petrokimya Holding Corp. (PETKM), which is listed in Borsa Istanbul (BIST), using PETKM stock price, US dollar (USD/TRY) price and BIST Chemical, Petroleum & Plastic (XKMYA) index price. A time series data set with three inputs and one output is created using these data. Random Forest Regression (RFR), Long-Short Term Memory (LSTM), and Convolutional Neural Network (CNN) algorithms are used in the prediction model. The success of these methods is compared using performance metrics such as MSE, RMSE, MAE, and R^2. According to the calculated error metrics, LSTM and RFR algorithms gave better results than CNN with an MSE value less than 0.02. However, the fact that the R^2 values of the most successful models created with all three algorithms were greater than 95% revealed that all the algorithms mentioned could be used to estimate this data set.

Keywords: *Stock price prediction, Machine learning, RFR, Deep learning, LSTM, CNN*

Makine Öğrenmesi Algoritmalarıyla Hisse Senedi Kapanış Fiyat Tahmini: BIST'te Yer Alan PETKM Hisse Senedi Örneği

ÖZET

Bu çalışmada Borsa İstanbul'da (BİST) yer alan Petkim Petrokimya Holding A.Ş.'nin (PETKM) hisse senedi fiyatından, Dolar (USD/TRY) fiyatından ve BİST Kimyasal, Petrol & Plastik (XKMYA) indeks fiyatından yararlanılarak, PETKM hisse senedi fiyatının tahmin edildiği üç girdili ve bir çıktılı bir zaman serisi veri seti oluşturulmuştur. Zaman serisi modelleri için Random Forest Regression (RFR), Long-Short Term Memory (LSTM) ve Convolutional Neural Network (CNN) algoritmalarının ayrı ayrı çalışmalarda başarılı sonuçlar elde ettikleri görüldüğünden hisse senedi fiyatının tahmini için bu üç algoritma kullanılmıştır. Literatürde belirtilen kapsamda bu üç yöntemin karşılaştırıldığı bir çalışmaya rastlanmamıştır. Algoritmaların başarısı, genellikle bu tür çalışmalarda kullanılan MSE, RMSE ve MAE olmak üzere üç hata metrik değerleriyle ve R^2 değeriyle karşılaştırılmıştır. Hesaplanan hata metriklerine göre LSTM ve RFR algoritmalarında MSE değeri 0.02'den küçük olup CNN'den daha başarılı sonuçlar vermesine rağmen R^2 değerlerinin %95'te büyük olmasıyla her üç algoritmadan oluşturulan en başarılı modellerin bu veri setinin tahmininde kullanılabileceği görülmüştür.

Anahtar kelimeler: *Hisse senedi fiyat tahmini, Makine öğrenmesi, Derin öğrenme, RFR, LSTM, CNN*

Received: 03/04/2022, Revised: 26/08/2022, Accepted: 29/08/2022

I. INTRODUCTION

Time series are continuous series created using unit time data within a certain time period. Time series analysis, on the other hand, is done to extract specific features from the data and to obtain meaningful statistics. With these statistical and data analyses, meaningful inferences can be made, and accurate past-future predictions can be easily made. Time series estimations give accurate and reliable results when made with meaningful data [1].

One of the best examples of these today is the financial time series. It is possible to obtain significant returns with accurate estimates, especially considering certain features in investing and portfolio creation in stock markets that have become a critical investment gateway [2]. Although it is possible to obtain significant returns from this dynamic investment tool, it is also tricky. Stock prices have a noisy and volatile structure [3]. The reason for this structure is that stock prices are affected by many factors in a wide range from investor sensitivity, economic and political situation of the country, production, the condition of the sector to which the relevant stock belongs, and the diplomatic relations of the country with other countries [4].

Investors can use their intuition and personal experience in stock trading. However, this situation further increases the risks of economic loss due to the fluctuation of stock market movements and the fact that they are not based on any scientific forecasting method. Investors who realize this are looking for more solid bases to invest [5]. There are three different estimation approaches to make the right decisions for stock transactions. These are 1- technical analysis is used in estimating future values by graphical analysis of past data in short-term forecasts. 2- fundamental analysis is used to make forecasts by examining the financial information of the relevant company and the political and economic situation of its country. 3- time series analysis includes linear and non-linear models and algorithms used in future prediction [6].

There are various methods in time series analysis. These methods are generally divided into three as traditional linear estimation methods, traditional non-linear estimation methods, and artificial intelligence estimation methods [7]. Today, it is seen that artificial intelligence methods are more prominent in these stock index and price predictions [8]. Traditional linear estimation methods (AR, MA, ARIMA, etc.), which have been used for many years, cannot be used in all kinds of data due to the linear structure and only the data when the model is established [9]. Traditional non-linear methods (ARCH, GARCH, etc.) can make a more reliable estimation if the variance of the time series data is linear [10]. After the emergence of artificial intelligence models, it has been proven by the results obtained from many studies that these methods produce more accurate and reliable prediction results than traditional methods. The reason for this is that artificial intelligence methods have the ability to make decisions while making future predictions by using past data [11].

Machine Learning (ML), one of the method groups covered by artificial intelligence, includes linear and non-linear artificial neural networks and deep learning prediction algorithms [4]. ML algorithms are safe methods that can extract patterns and extract information from existing data [7]. Akşehir and Kılıç [8] applied Multiple Regression (MLR), Decision Trees (DT), and Random Forest (RF) methods on a linear financial time series that they created using fundamental and technical indicators that can be used in the estimation of bank stocks. As a result, they determined that all the methods they used gave successful results according to the R^2 value. Tan et al. [12] observed that tree-based algorithms provide better and more accurate results than traditional regression methods for financial time series with a fluctuating structure. Kaczmarczyk and Hernes, [13] used the RF algorithm in their decision support system. They used this algorithm to determine the indicators to be used in technical analysis. Ciner [14] compared the RF method, which can also be used with linear and non-linear data, with the advanced methods in the literature and proved that the predictive ability of RF is much better than other estimation methods. When the algorithms included in artificial neural networks (ANN) are compared with different ML algorithms, it has been seen that ANN algorithms give better results [11]. Keskin and Yücel, compared the BİST100 and Gold Index values which between 1988-2019 with the

data of 2020 using a ANN model. The aim of the study is to determine whether there is a relation between BIST100 and Gold index prices. As a result of the comparison of the model they created, 54% showed that ANN is a method that can be used to predict the connection between BIST100 and Gold index [15]. Deep Learning (DL) methods, which emerged with the development of the ANN method, have started to be among the solution methods frequently used in real-life problems of financial time series, as they can learn and make sense of the data [16]. Ghosh et al. [7] stated that DL methods such as Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) give successful results with multivariate financial time series. Hiransha et al. [9] created prediction models for various sectors by using Multi-Layer Perceptron (MLP), RNN, CNN, and Long-Short-Term Memory (LSTM) algorithms in two different exchanges and determined that the CNN algorithm gave the best results. Sakarya & Yılmaz [6] estimated the BIST30 index with a high success rate by using the Wavelet Transform method to remove noise in the data, increase the data quality, stacked autoencoders to detect rare features in the data, and the LSTM method as the estimation method. Dalkıran and Ozan formed that a new data set with stock prices of ISCTR, GARAN, VAKBN, QNBFB, AKBNK, index values of Borsa İstanbul Banking Index (XBANK), BIST30 and US Dollar/Turkish Lira (USD/TR) price and estimated the ISCTR stock price value with the LSTM algorithm[17]. Sarıkoç and Çelik applied Factor Analysis (FA), Principal Component Analysis (PCA) and Independent Component Analysis (ICA) methods for data preprocessing to the data set they created with the BIST100 index and the technical indicators that are thought to affect it. By combining these methods with LSTM, they observed whether the methods used in the data processing step affect the estimation result [18]. Akşehir and Kılıç [19, 20] establish that the CNN algorithm gave successful results in eliminating data imbalance and in data set feature selection. Ozbayoglu et al. [21] analyzed the methods used for financial time series estimation in the literature and found that DL methods provided much higher performance than other algorithms. Although it is seen that these algorithms give successful results in separate studies, when the literature is examined, no study was found in which RF, LSTM, and CNN algorithms were applied and compared within the same research in stock price estimation.

This study estimated Petkim Petrokimya Holding Corp's stock price (PETKM) using Random Forest Regression (RFR), LSTM, and CNN algorithms. The rest of the study is organized as follows: In the second part, the data set, the algorithms, the statistical analysis methods, and the method used to optimize the results are explained. Application details are given in section 3. In the last section, the obtained results are presented and discussed.

II. MATERIALS AND METHODS

A. DATA SET DESCRIPTION

In this study, daily closing prices of US dollar (USD/TRY), BIST Chemical, Petroleum & Plastic (XKMYA), and Petkim Petrokimya Holding Corp. (PETKM) for eleven years (03.05.2010-03.05.2021) were used. The USD/TRY price is used in estimating PETKM stock closing price because it is thought to affect PETKM stock price since oil purchases are generally made in USD. At the same time, as the USD price is greatly affected by the critical events in the world, many areas will affect the stock market values on this input. The reason for using the XKMYA index price is that the PETKM index is thought to be affected by the situation of other stocks in the sector in the country. The number of data for each index is 2760. Weekends and holidays are not included in the data set as stock markets are closed on those days. All data is available as open access, which can be obtained at [22]. The graphs of the data are given in Figure 1.

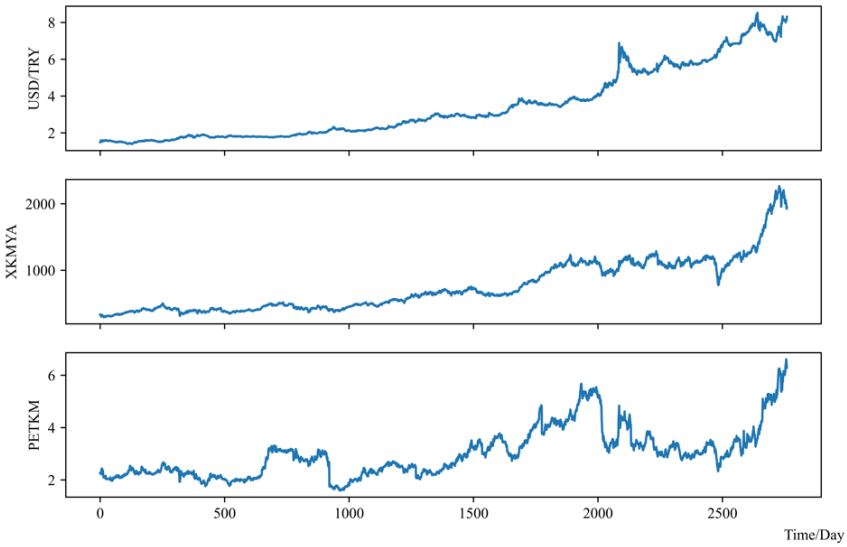


Figure 1. *Data set variables.*

Since the data set covers a long period, the difference in the value range of each variable is large. The value ranges of data set variables are given in Table 1.

Table 1. *The value range of each variable in the dataset.*

Data Set Variable	Lower Value	Upper Value
USD/TRY	1.3953	8.5250
XKMYA	292.16	2267.44
PETKM	1.595	6.610

B. MACHINE LEARNING (ML)

Machine learning is computer modeling that enables making accurate and reliable predictions with the help of statistics and mathematical science from past data [4]. Using historical data with models created with ML, new predictions can be made with similar and different data, and data from new sources can be analyzed. It gives more successful results, especially in time series models containing large amounts of data [23]. ML methods are divided into four main groups.

ML methods are divided into four main groups as supervised, unsupervised, reinforcement, and instance-based learning algorithms. All algorithms used in this study are included in the supervised learning method because of historical data for stock price prediction. The purpose of the supervised learning type is to obtain a trained model that makes sense of the relationship between the established models and the input and output data and to test new data with the information obtained from this trained model. Depending on the data type and the desired result from the problem, this learning method can be used in classification and regression problems. The most common uses for regression algorithms are weather forecasting, stock index, and price forecasting [23].

B. 1. Random Forest Regression (RFR)

Random Forest is one of the powerful techniques of machine learning (ML) used in both classification and regression problems consisting of many decision trees [14]. It consists of many independent decision trees in terms of structure [12]. The final decision is made according to the average of the information obtained from all the decision trees created [24]. RF models can be trained with a training set consisting of a part of the data set and easily learn the principles and rules. This method is a model

that contains many decision nodes and gives successful results in the decisions of risky situations [14]. It has been concluded that, by its nature, a problem can make objective decisions by preventing overfitting even when the number of variables is large [12]. The structure of the Random Forest Regression algorithm is given in Figure 2.

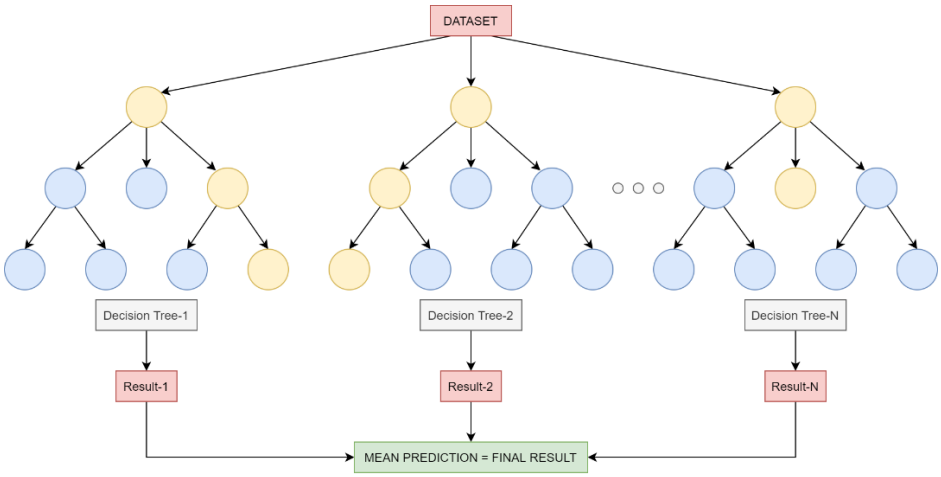


Figure 2. *RFR Algorithm Structure.*

The number of trees to be used in estimating the creation of an RFR model with X inputs in the data set is shown in Equation 1.

$$h(X;\theta_k), k = 1,2,3 \dots K \tag{1}$$

The estimation model of the K group of trees is created as in Equation 2.

$$Sequence = \{h(X;\theta_1), h(X;\theta_2), \dots, h(X;\theta_k)\} \tag{2}$$

The final estimation result is found as in Equation 3 [25].

$$\overline{h}(X) = (1/K) \sum_{k=1}^{K} h(X;\theta_k) \tag{3}$$

The reason for choosing this method for this study is that it is a powerful method that can make unbiased and successful predictions in complex data [25]. The RFR method was implemented in the Python 3.7 scripting language using the Pandas, SciKit Learn, Numpy, Math, and Matplotlib libraries in the Spyder 3 editor.

C. DEEP LEARNING (DL)

Deep Learning algorithms are large-scale ANNs and are machine learning (ML) methods more similar to human learning. As the amount of data increased, the capacity of ML methods began to be insufficient, which has led to a decrease in performance in producing accurate output. For this reason, DL algorithms have been created that achieve better results than traditional ML algorithms. There are various layers in DL and neurons like the human brain [21]. Thanks to the neurons in the layers, new information and features are obtained from each data used as input in the system. DL algorithms can decide whether to use this information and features. In this way, high performance can be achieved in predicting or classifying new data entered into the system with the information obtained from the training set used in the algorithm's training [23].

C. 1. Long Short-Term Memory (LSTM)

A recurrent Neural Network (RNN) is a neural network structure that uses the new information entered into the system and the information obtained from the previous unit's output, accepts this information as input in the following unit, and creates loops. The aim here is to use the information sequentially, which helps establish a hierarchical order in learning information. A simple RNN network consists of input, output, and an RNN cell. There are hidden layers inside the RNN cell and neurons inside these hidden layers. The learning process takes place in these layers [26]. It uses the information learned from the inputs of previous units to decide the outputs in RNN structures [6].

Long Short-Term Memory (LSTM) is a type of repetitive neural network developed by creating solutions to problems in RNN [27]. While RNN can remember short-term information, LSTM can remember both short-term and long-term input information [28-29]. A simple RNN cell has only one layer containing the tanh activation function. There are four different components in LSTM units [30]. These components are:

- *Cell:* It is the structure that stores a value in the LSTM unit for its operations in other components [30].

- *Forget gate (f_t):* It is the gate that decides whether the information coming to the current unit through the previous units will be transferred to the next unit or not [31].

- *Input gate (i_t):* It is the part where the data to be taught to the model enters the unit. Here, the new information entering the system and the information from the previous unit are combined [31]. The decision mechanism in this door consists of two steps. In the first step, the status of the new information is decided. In the second step, depending on the decision of the input gate, the inputs are combined if necessary [6].

- *Output gate (o_t):* The forget gate, which decides whether the information from the previous units will be transmitted to the next unit or not, and the decisions of the input gates, which determine whether to use the new input, come to the output gate as combined [31].

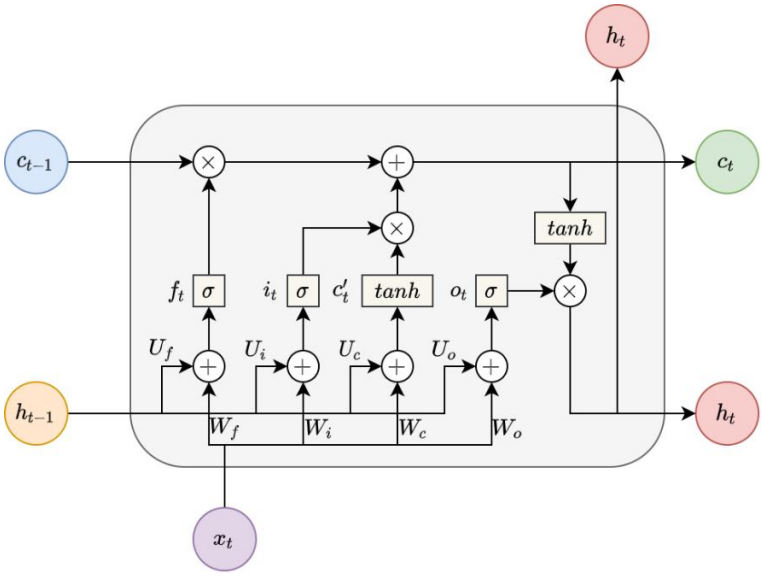


Figure 3. *LSTM cell structure.*

The structure of an LSTM cell is given in Figure 3. LSTM cell input x_t, memory state c_t, and cell output h_t and σ are shown as activation functions. Each cell input is the new input x_t and h_{t-1} the output of the previous LSTM cell. These input values are associated with the U and W weight sets at the forget gate (f_t), input gate (i_t) and output gate (o_t) inputs.

The forget gate has a decision mechanism based on the h_{t-1} information from the output of the previous cell and the x_t input. This mechanism is illustrated in Equation 4.

$$f_t = \sigma(W_f x_t + U_f h_{t-1}) \quad (4)$$

c_{t-1} is a vector representing the memory state of the previous LSTM cell. Whether the information in this vector will pass to the next LSTM cell is determined by a function applied to each element of the vector. With this function operation, vector values take the values 0 or 1. The value of 0, obtained from the process, deletes the information, while the value of 1 decides that the information should pass to the next LSTM cell.

Decisions regarding the information to enter the LSTM cell are made at the Input gate. The blacks are associated with the new input x_t and the output h_{t-1} of the previous LSTM cell. With the help of the activation function, 0 or 1 values are obtained. When the output of the function is 0, no new information is entered into the cell. When it is 1, new information can enter the cell. This operation at the input gate is given in Equation 5.

$$i_t = \sigma(W_i x_t + U_i h_{t-1}) \quad (5)$$

The input candidate vector designated c_t' is associated with the output (h_{t-1}) and the new input (x_t) of the previous LSTM cell. c_t' is converted to the required format by an activation function. This operation is given in Equation 6.

$$c_t' = \tan h\,(W_c x_t + U_c h_{t-1}) \quad (6)$$

The input and forget gates decide the new cell memory combination created with the current state and new inputs. The decision process is given in Equation 7.

$$c_t = f_t * c_{t-1} \oplus i_t * c_t' \quad (7)$$

For the next LSTM cell block, the output gate decides whether the current block will output a piece of information and, if it does, which information will pass to the next LSTM cell. With a specified activation function, the output value takes the values 0 or 1. When the output value is 0, there is no information transfer from the current LSTM cell to the next cell, while when it is 1, information is transferred to the next cell. This process is shown in Equation 8.

$$o_t = \sigma(W_o x_t + U_o h_{t-1}) \quad (8)$$

By using the output gate's decision and an activation function, the information set to pass to the next LSTM cell is transmitted to the next cell by the process in Equation 9 [6, 32].

$$h_t = o_t * \tan h\,(c_t) \quad (9)$$

This method was chosen for this study because it is a powerful ML method that can make predictions on solid grounds, with the ability to make connections with past data, remember the past, and decide on the information to use [29].

C. 2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a linear ML method often used in visual data. The purpose of hidden layers is to find non-linear patterns in the data and make accurate predictions [23]. Especially the filter parameter given in the convolution layer makes it easier to find the hidden features in the data. In recent studies, it has been seen that the CNN method gives effective and efficient values in time series analysis. CNN layers consist of two main structures. One of them is feature extraction

layers called convolution layers. The other is the layer where regression and classification operations are performed, called the pooling layer [33].

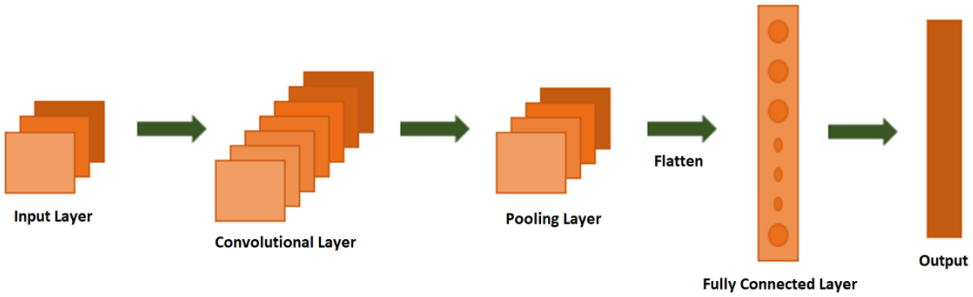


Figure 4. *CNN algorithm structure.*

Figure 4 shows the structure of the CNN algorithm. In the Convolutional Layer, the input properties are placed in rows. Input data is divided into small parts associated with weights and enters the matrix form. It contains neurons that display filtering behavior to determine feature maps. The segmented input sets form various feature sets with the filtering process given in Equation 10.

$$F_l^k = (A * K_l^k) \tag{10}$$

In Equation 10, A represents the input matrix, K_l^k is the lth filter of the kth layer, and F_l^k is the feature map value of the layer.

The Pooling Layer reduces the number of parameters by finding the dominant feature in each part of the input set with the help of Equation 11.

$$P_l = f_p(F^l) \tag{11}$$

In equation 11, f_p represents the pooling operation, F^l represents the lth input, and P_l the feature map of the lth input.

After the pooling layer, flatten operation transforms the input into a column vector in matrix form. On the other hand, the Fully Connected Layer creates a non-linear feature combination by taking into account all the outputs of the previous layers. It is tried to obtain the best estimation result with the feature combination created by considering the outputs in the previous layers [34].

Using this method in this study is that CNN algorithms give much more successful results than other ML algorithms in extracting pattern features from the data and making correct predictions [35]. Another reason is that CNN algorithms are used less frequently in such problems, although they have essential features for time series problems when the literature is examined.

CNN and LSTM methods are implemented in the Python 3.7 scripting language in Spyder 3 editor using the Pandas, SciKit Learn, Keras, Numpy, Math, and Matplotlib libraries. The algorithms are run in a machine that has 8.00GB RAM, Intel(R) Core(TM) i7-4510U CPU processor, and Windows 10 Pro 64 bit operating system.

D. HYPERPARAMETER TUNING

Hyperparameters are parameter types that do not cause a change in the formation of the model, such that they can be found by trial and error and can be used in optimizing the model. The method of finding hyperparameter values by trial to optimize the model's error value is called Hyperparameter Tunning [34].

Hyperparameters used for Random Forest Regression (RFR) algorithm are training set data count, test set data count, random_state, n_estimator values.

The hyperparameters used for the Long-Short Term Memory (LSTM) algorithm are the number of data in the training set, the number of data in the test set, the number of layers, the optimizer, the number of units, the learning rate value, the activation function, etc.

The hyperparameters used for the Convolutional Neural Network (CNN) algorithm are the number of data in the training set, the number of data in the test set, the number of layers, the number of optimizers, the number of filters, the kernel size, the learning rate value, the activation function, etc.

E. STATISTICAL ANALYSIS OF MODEL'S RESULTS

Statistical analysis is applied as the last step of each model. In this study, error metric methods such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2) was used. The error values, near 0, and the R^2 value, near 1, show that the model gives a good result [14]. The reason for choosing these metric error values to evaluate the performance of the models is that they are the most used evaluation indicators in stock price prediction studies in the literature [36]. The MSE, RMSE, MAE, and R^2 are calculated using Equations 12, 13, 14, 15, respectively.

$$MSE = \frac{1}{n}\sum_{j=1}^{n} e_j^2 \tag{12}$$

$$RMSE = \sqrt{\frac{\sum_{j=1}^{n} e_j^2}{n}} \tag{13}$$

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |e_j| \tag{14}$$

$$R^2 = 1 - \frac{\sum(y_i - \bar{y}_i)^2}{(y_i - \bar{y})^2} \tag{15}$$

In all performance metrics, n represents the number of data to be processed, and e_j represents the error value between j^{th} observation value and its prediction value. In the R^2 Equation, y_i represents the actual values, $\bar{y}_i$ represents the predicted values, and $\bar{y}$ represents the mean value. In this study, the error values were found by applying the mean_squared_error(), mean_absolute_error(), r2_score() and sqrt() (square root) function in the metrics method in the SciKit-Learn library.

III. APPLICATION

In this study, the value of the PETKM index was tried to be estimated by using the closing values of Dollar (USD/TRY) price, BIST Chemical, Petroleum & Plastic (XKMYA), and Petkim Petrokimya Holding Corp. (PETKM) stock prices. Random Forest Regression (RFR), Long-Short Term Memory (LSTM), and Convolutional Neural Network (CNN) methods were used as estimation methods. The training and test set sizes and the other hyperparameters used in the algorithms have been optimized by the Hyperparameter Tuning method. This section explains the parameters used in the algorithms and presents the values given to these parameters.

Different training and test sets have been experimented with to bring out the best results in the estimation process with the RFR algorithm. Test set sizes were determined as 20% (552), 30% (828), and 50% (1380) of the whole data. There are two essential parameters in the creation of this algorithm. The first is "N_estimators," which shows the number of decision trees selected randomly from all the

data. The second one is the "Random_States" value ensures the same result every time the algorithm is run with the same parameter values. Hyperparameter values used to optimize the RFR algorithm are given in Table 2, along with the error metrics and R^2 value.

Table 2. *Optimization experience of hyperparameter and model error values of test set for RFR algorithm.*

Model	Train Set Size	Test Set Size	Random_States	N_estimators	MSE	RMSE	MAE	R^2
R1	2.208	552	0	50	0.017	0.130	0.030	0,980
R2	2.208	552	0	80	0.017	0.130	0.0305	0,980
R3	**2.208**	**552**	**0**	**100**	**0.016**	**0.129**	**0.030**	**0,981**
R4	2.208	552	10	80	0.017	0.130	0.031	0,980
R5	2.208	552	10	100	0.017	0.129	0.302	0,981
R6	1.932	828	0	50	0.024	0.155	0.0459	0,970
R7	1.932	828	0	70	0.024	0.155	0.459	0.969
R8	1.932	828	10	100	0.025	0.159	0.0486	0,971
R9	1.380	1.380	0	50	0.804	0.892	0.514	0,12

LSTM algorithms have many hyperparameters that help optimize models. Number of data in test and train sets, number of layers, number of units/threshold value in layers, activation function, optimizer of LSTM model, features of the optimizer, and number of epochs are some of these hyperparameters.

To find the number of training and test data values that will bring out the best results in the estimation process with the LSTM algorithm, the test set values were determined as 20% (552), 30% (828), and 50% (1380) of the whole data. Different LSTM models have been established with these values. Among the models, the number of layers is given between 3-5, unit values between 50-100, threshold values between 0.10-0.20, and epoch values between 70-90. ReLU, tanh, and sigmoid were used as activation functions. Increasing or decreasing these values and the type of activation used significantly change the prediction and speed performance of the model. Some of the hyperparameter and error values of different models created using the LSTM algorithm are given in Table 3.

Table 3. *Optimization experience of hyperparameter and model error values of test set for LSTM algorithm.*

Model	Train Set Size	Test Set Size	Layer	Unit/ Threshold	Activation Function	Optimizer/ Learning Rate	Epoch	MSE	RMSE	MAE	R^2
L1	2.208	552	LSTM Dropout Dense	50 0.10 1	Sigmoid	Adam 0.02	70	0.059	0.243	0.190	0.920
L2	**2.208**	**552**	**LSTM Dropout Dense**	**70 0.15 1**	**Tanh**	**Adam 0.02**	**70**	**0.016**	**0.128**	**0.087**	**0.985**
L3	1.932	828	LSTM Dropout LSTM Dropout Dense	50 0.10 80 0.20 1	Tanh	Adam 0.01	70	0.065	0.255	0.223	0.920
L4	2.208	552	LSTM Dropout LSTM Dropout Dense	90 0.20 80 0.20 1	Tanh ReLU	Adam 0.01	90	0.033	0.183	0.152	0.960
L5	2.208	552	LSTM Dropout LSTM Dropout Dense	90 0.20 80 0.20 1	Sigmoid Tanh	Adam 0.01	70	0.045	0.213	0.139	0.940
L6	1.932	828	LSTM Dropout LSTM Dropout Dense	90 0.20 80 0.20 1	Sigmoid Tanh	Adam 0.01	70	0.163	0.404	0.342	0.763
L7	1.932	828	LSTM Dropout Dense	70 0.15 1	Tanh	Adam 0.02	70	0.088	0.297	0.268	0.887
L8	1.380	1380	LSTM Dropout Dense	70 0.15 1	Tanh	Adam 0.02	70	0.487	0.698	0.508	0.241
L9	2.208	552	LSTM Dropout Dense	50 0.10 1	Tanh	Adam 0.02	70	0.023	0.122	0.150	0.970

To find the number of training and test data values that will produce the best result in the estimation process with the CNN algorithm, the test set values were determined as 10% (552), 20% (552), and 30% (828) of the whole data. Different CNN models have been established with these values. In the models, the filter value is between 64-128, the unit value is between 50-70, and the epoch value is between 50-80. ReLU and tanh functions were generally used in the models since the sigmoid function was largely unsuccessful in its predictions in all experimental models. Some of the hyperparameter and error values of different models created using the CNN algorithm are given in Table 4.

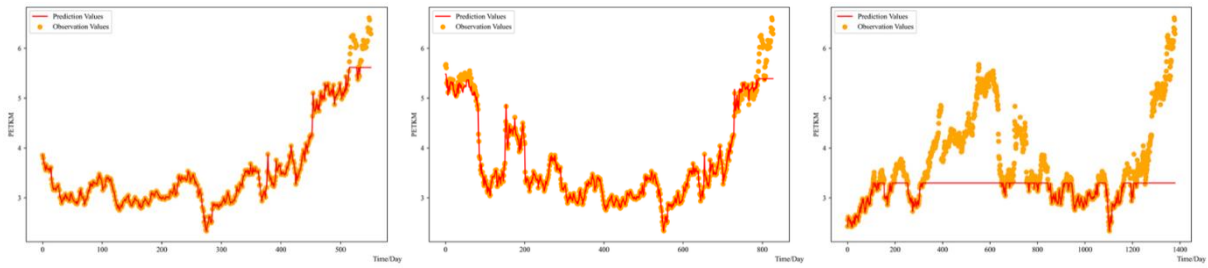
Table 4. *Optimization experience of hyperparameter and model error values of test set for CNN algorithm.*

Model	Train Set Size	Test Set Size	Layer	Unit/ Filter/ Pool Size	Activation Function	Optimizer	Epoch	MSE	RMSE	MAE	R^2
			Conv1D	64	ReLU						
			MaxPooling1D	2							
C1	2.208	552	Flatten			Adam	50	0.196	0.442	0.369	0.797
			Dense	50							
			Dense	1							
			Conv1D	128	ReLU						
			MaxPooling1D	2							
C2	2.208	552	Flatten			Adam	50	0.137	0.370	0.301	0.886
			Dense	70							
			Dense	1							
			Conv1D	128	ReLU						
			MaxPooling1D	2							
C3	2.208	552	Flatten			Adam	80	0.051	0.226	0.174	0.926
			Dense	70							
			Dense	1							
			Conv1D	128	Tanh						
			MaxPooling1D	2							
C4	2.484	276	Flatten			Adam	80	0.286	0.534	0.385	0.771
			Dense	70							
			Dense	1							
			Conv1D	128	ReLU						
			MaxPooling1D	2							
C5	2.484	276	Flatten			Adam	80	0.072	0.269	0.195	0.912
			Dense	70	Tanh						
			Dense	1							
			Conv1D	**128**	**ReLU**						
			MaxPooling1D	**2**							
C6	**2.484**	**276**	**Flatten**			**Adam**	**80**	**0.041**	**0.202**	**0.141**	**0.969**
			Dense	**70**							
			Dense	**1**							
			Conv1D	128	ReLU						
			MaxPooling1D	2							
C7	1.932	828	Flatten			Adam	80	2.589	1.609	1.394	-1.973
			Dense	70							
			Dense	1							

IV. RESULTS

In this study, a successful model has been tried to be created in PETKM's closing price estimation by using the closing prices of US Dollars (USD/TRY), BIST Chemical, Petroleum & Plastic (XKMYA), and Petkim Petrokimya Holding AŞ (PETKM) stocks as inputs. For the estimation of PETKM value, when the literature is examined, no study evaluates the outputs of Random Forest Regression (RFR), Long-Short Term Memory (LSTM), and Convolutional Neural Network (CNN) algorithms for time series models. Since they have obtained successful results in separate studies, this is the case for estimating the model. Three algorithms were used, and many models were created separately. In this section, the results of the models made were compared and evaluated.

The test set of the R1 model consists of 552 data, the test set of the R6 model consists of 828, and the test set of the R9 model consists of 1380 data. In all three models, N_estimators and random_state_values are given as 50 and 0, respectively. The hyperparameter value that changes for these three models is the number of training and test sets. The estimation graphs of the test sets of the R1, R6, and R9 models are given in Figure 5.

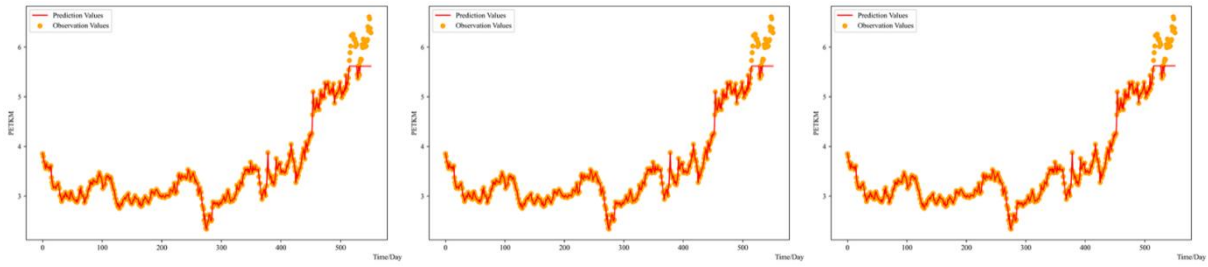


(a) ***(b)*** ***(c)***

Figure 5. (a) *The Prediction Result of The Test Set of R1 Model and* ***(b)*** *The Prediction Result of The Test Set of R6 Model and* ***(c)*** *The Prediction Result of The Test Set of R9 Model.*

The MSE value for the R1 model is 0.017; RMSE value is 0.130; MAE value is 0.030; R^2 value is 0.98. The MSE value for R6 model is 0.0242; RMSE value is 0.155; MAE value is 0.0459; R^2 value is 0.97. The MSE value for the R9 model is 0.8036; RMSE value is 0.892; MAE value is 0.514; R^2 value is 0.12. In the RFR model, the importance of the change in the number of data for the test and training sets for this data set appears to affect performance greatly. The R1 model gave the best results.

The test set size of the R1, R2, and R3 models is 552, and the random_state value is 0. The N_estimators value is 50, 80, and 100, respectively. The estimation graphs of the test sets of the R1, R2, and R3 models are given in Figure 6.

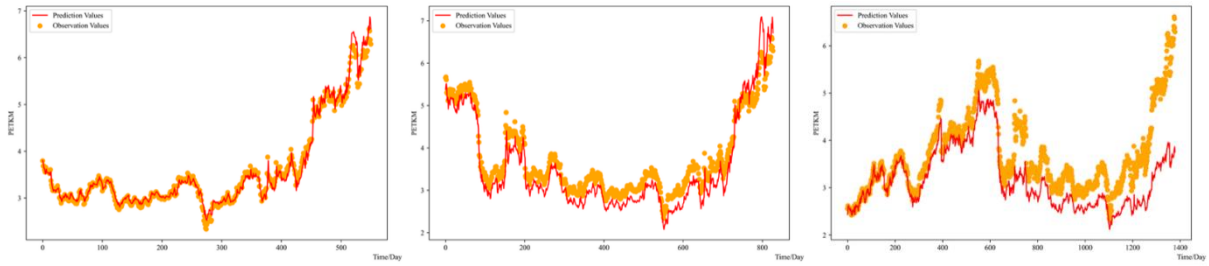


(a) ***(b)*** ***(c)***

Figure 6. (a) *The Prediction Result of The Test Set of R1 Model and* ***(b)*** *The Prediction Result of The Test Set of R2 Model and* ***(c)*** *The Prediction Result of The Test Set of R3 Model.*

The MSE value for the R1 model is 0.017; RMSE value is 0.130; MAE value is 0.030; R^2 value is 0.98. The MSE value for the R2 model is 0.0169; RMSE value is 0.130; MAE value is 0.0305; R^2 value is 0.98. The MSE value for the R3 model is 0.0166; RMSE value is 0.129; MAE value is 0.030; R^2 value is 0.981. In the RFR model, it is seen that the change of the N_estimator value for the test set in this data set does not have a great effect.

The test set of the L2 model consists of 552 data, the test set of the L7 model consists of 828, and the test set of the L8 model consists of 1380 data. In all three models, the number of layers is 3. The layer types are LSTM, dropout, and dense. Unit/threshold values are 70, 0.15, and 1, respectively. The activation function value is given as tanh. The optimizer type is Adam (momentum). The learning rate value is 0.02, and the epoch value is 70. The estimation graphs of the test sets of the L2, L7, and L8 models are given in Figure 7.

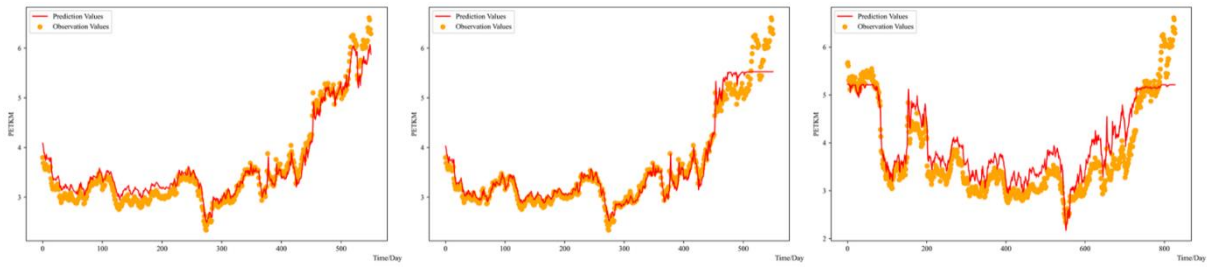


(a) ***(b)*** ***(c)***

Figure 7. (a) *The Prediction Result of The Test Set of L2 Model and* ***(b)*** *The Prediction Result of The Test Set of L7 Model and* ***(c)*** *The Prediction Result of The Test Set of L8 Model.*

The MSE value for the L2 model is 0.016; RMSE value is 0.128; MAE value is 0.087; R^2 value is 0.985. The MSE value for L7 model is 0.08; RMSE value is 0.297; MAE value is 0.268; R^2 value is 0.887. The MSE value for L8 model is 0.487; RMSE value is 0.698; MAE value is 0.508; The R^2 value is 0.241. In the LSTM model, the importance of the change in the size of the test and training sets for this data set greatly affects performance. L2 model gave the best results.

The number of the data in the test set of the L4 model is 552. The activation function of the first LSTM layer is tanh, and the unit value is 90. The threshold value of two dropout layers is 0.20. The activation function of the second LSTM layer is ReLU, and the unit value is 80; the epoch value is 90. The number of the data in the test set of the L5 model is 552. The activation function of the first LSTM layer is sigmoid, and the unit value is 90. The threshold value of two dropout layers is 0.20. The activation function of the second LSTM layer is tanh, and the unit value is 80; the epoch value is 70. The number of the data in the test set of the L6 model is 828. The activation function of the first LSTM layer is sigmoid, and the unit value is 90. The threshold value of the dropout layer is 0.20. The activation function of the second LSTM layer is tanh, and the unit value is 80; the epoch value is 70. The estimation graphs of the test sets of the L4, L5, and L6 models are given in Figure 8.

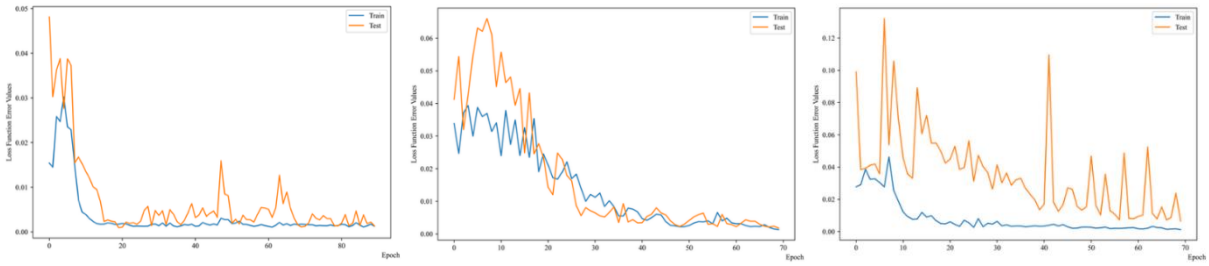


(a) ***(b)*** ***(c)***

Figure 8. (a) *The Prediction Result of The Test Set of L4 Model and* ***(b)*** *The Prediction Result of The Test Set of L5 Model and* ***(c)*** *The Prediction Result of The Test Set of L6 Model.*

The MSE value for the L4 model is 0. 033; RMSE value is 0.183; MAE value is 0.190; R^2 value is 0.96. The MSE value for L5 model is 0.045; RMSE value is 0.213; MAE value is 0.139; R^2 value is 0.94. The MSE value for L6 model is 0.163; RMSE value is 0.404; MAE value is 0.342; The R^2 value is 0.763.

Looking at the graphs given in Figure 8 and the error values of the models, the difference in the activation function values and epoch values between the L4 and L5 models caused a slight change in the error values. The fact that test set size of the L5 model is 552, and the test set size of the L6 model is 828. It is seen that the L4-L5 models have a higher margin of error than the difference in error values. The comparison of these error values for training and test sets is given in Figure 9.

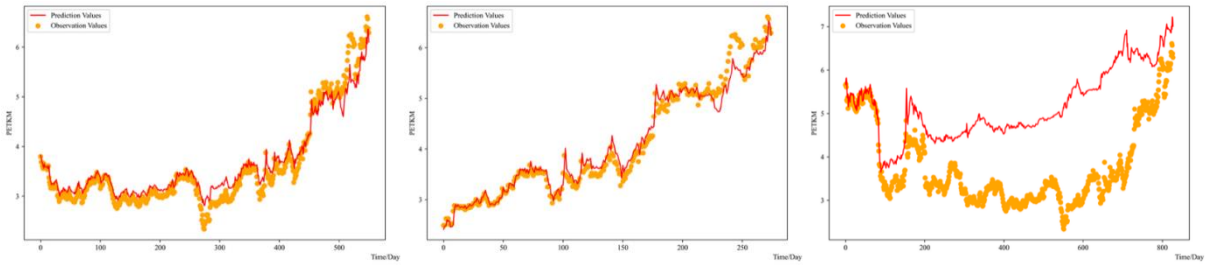


(a) ***(b)*** ***(c)***

Figure 9. (a) *The Loss Function Value of L4 Model and* ***(b)*** *The Loss Function Value of L5 Model* ***(c)*** *The Loss Function Value of L6 Model.*

In Figure 9, it is seen that the error values of the training and test sets of the L4 and L5 models are in harmony with each other, while there is a big difference between the error values of the training and test sets of the L6 model. The reason for this is that stock market movements are under the influence of many factors. When the number of data used for testing increases, large fluctuations are observed in stock prices.

The test set of the C3 model consists of 552 data, the test set of the C6 model consists of 276, and the test set of the C7 model consists of 828 data. In all three models, the number of layers is 5. The layers are Conv1D, MaxPooling1D, flatten, and dense. The filter, pool, size, and unit values are 127, 2, 70, and 1, respectively. The activation function value is ReLU, optimizer type is given as Adam (momentum), and the epoch value is 80. The estimation graphs of the test sets of the C3, C6, and C7 models are given in Figure 10.

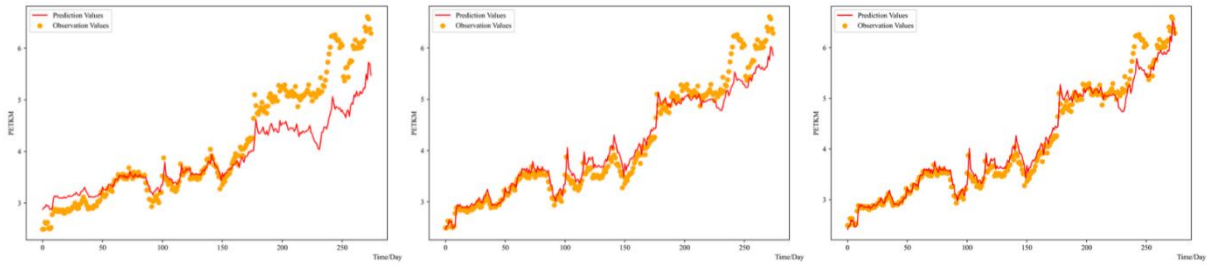


(a) ***(b)*** ***(c)***

Figure 10. (a) *The Prediction Result of The Test Set of C3 Model and* ***(b)*** *The Prediction Result of The Test Set of C6 Model and* ***(c)*** *The Prediction Result of The Test Set of C7 Model.*

The MSE value for the C3 model is 0.051; RMSE value is 0.226; MAE value is 0.174; R^2 value is 0.926. The MSE value for model C6 is 0.041; RMSE value is 0.202; MAE value is 0.141; R^2 is 0.969. The MSE value for the C7 model is 2.589; RMSE value is 1.609; MAE value is 1,394; R^2 value is -1.973. A negative R^2 value indicates that the variables of the data set and/or the model do not comply with PETKM stock closing price predictions. The differences between the C3, C6, and C7 models are the training and test set sizes. While the C3 and C6 models have acceptable error and R^2 values, the error values of the C7 model are unacceptably high, and the R^2 value is negative, which indicates that increasing the number of data to be tested for CNN models for this data set may yield results with low success levels.

The activation functions of the Conv1D layer of model C4 and C5 is tanh and ReLU, respectively. In the C6 model, the activation function of the first dense layer is tanh, and the activation function of the Conv1D layer is ReLU. The test set sizes of all three models are 276; the number of filters in the Conv1D layer is 128; the pooling size value in the MaxPooling1D layer is 2; the unit value in the first dense layer is 70; the unit value in the second dense layer is 1; the model's optimizer is Adam (momentum), and the epoch value is 80. The estimation graphs of the test sets of the C4, C5, and C6 models are given in Figure 11.

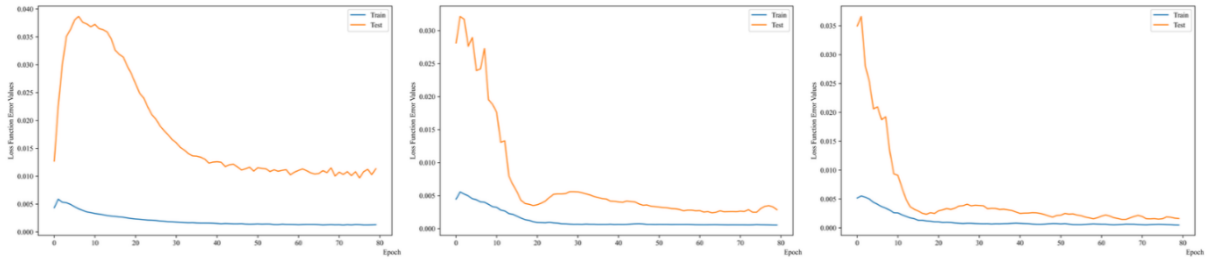


(a) ***(b)*** ***(c)***

Figure 11. (a) *The Prediction Result of The Test Set of C4 Model and* ***(b)*** *The Prediction Result of The Test Set of C5 Model and* ***(c)*** *The Prediction Result of The Test Set of C6 Model.*

The MSE value for the C4 model is 0.286; RMSE value is 0.534; MAE value is 0.385; R^2 value is 0.771. The MSE value for the C5 model is 0.072; RMSE value is 0.269; MAE value is 0.195; R^2 value is 0.912. The MSE value for the C6 model is 0.041; RMSE value is 0.202; MAE value is 0.141; R^2 value is 0.969. Only the activation functions are different for the C4, C5, and C6 models. Tanh activation function was used in the C4 model, Tanh and ReLU activation functions were used in the C5 model, ReLU activation function was used in the C6 model. When the error metrics and R^2 values were compared, it was observed that the ReLU function gave better results for this data set.

Comparison of prediction results of the test set of C4, C5, and C6 models are given in Figure 11. The fact that the different activation functions of the C5 and C6 models were caused a slight difference in the error values. Tanh activation function, which is used in the C4 model, significantly increased the error value. Therefore, it is seen that the tanh activation function does not give a good result for this model and data set. The comparison of these error values for training and test sets is shown in Figure 12.



(a) ***(b)*** ***(c)***

Figure 12. (a) *The Loss Function Value of C4 Model and* ***(b)*** *The Loss Function Value of C5 Model* ***(c)*** *The Loss Function Value of C6 Model.*

In Figure 12, it is seen that the error values of the training and test sets of the C5 and C6 models are in a consistent difference. On the contrary, there is a big difference between them in the C4 model. This difference is thought to be due to the tanh activation function in the C4 model.

Among the models created with the RFR algorithm, R3 gave the best results. Among the models made with the LSTM algorithm, L2 gave the best results. Among the models created with the CNN algorithm, C6 gave the best results. The estimation graphs of the test sets of the R3, L2, and C6 models are presented in Figure 13.

(a) ***(b)*** ***(c)***

Figure 13. (a) *The Prediction Result of The Test Set of R3 Model and* ***(b)*** *The Prediction Result of The Test Set of L2 Model and* ***(c)*** *The Prediction Result of The Test Set of C6 Model.*

The MSE value for the R3 model is 0.0166; RMSE value is 0.129; MAE value is 0.030; R^2 value is 0.981. The MSE value for the L2 model is 0.016; RMSE value is 0.128; MAE value is 0.087; R^2 value is 0.985. The MSE value for the C6 model is 0.041; RMSE value is 0.202; MAE value is 0.141; R^2 value is 0.969.

On models created for all algorithms in study, the effectiveness of hyperparameters that are important on the basis of algorithms were checked. It has been seen that the most sensitive hyperparameter of all models created with these algorithms is the size of the training and test sets. The reason for this is that the stock markets are highly affected by many factors. When the data set is examined, it is seen that the recent Corona Virus epidemic has had a significant impact on it. Thus, it is seen that the prediction accuracy decreases when the date range of the data to be tested increases. The changes in the N-estimator and Random_State hyperparameters in the RFR algorithm did not make a significant difference in the error results. However, the changes in the LSTM and CNN algorithms' hyperparameters caused significant difference for error metrich values of these algorithms.When the error values of the three most successful models among the algorithms were compared, according to the MSE and RMSE error metrics, LSTM and MAE error metrics gave the best results. Although CNN fall behind these two algorithm models for all error metrics, the error values are at a perfect level. When the R^2 values were compared, this value was above 0.95 in all three models. Three independent variables used as inputs according to R^2 values significantly affect the changes in the closing price of the PETKM index used as output. With this study, PETKM stock closing price was successfully estimated. Among the algorithms, the models that give the best error and R^2 values are the models that can successfully predict the closing price of PETKM stock. It has been seen that the LSTM model which is the most successful in terms of error metrics and R^2 value in the study, achieved more successful results than the models created in studies [17] and [18] which can be considered similar in the literature. In future studies, other variables that will affect PETKM stock closing prices can be added to the models by developing the data set, and also forecasting studies can be carried out to include other companies in the BIST.

V. REFERENCES

[1] J. C. Jackson, J. Prassanna, Md. Abdul Quadir and V. Sivakumar, “Stock Market Analysis and Prediction using time series analysis,” Materials Today: Proceedings, 2021.

[2] W. Chen, H. Zhang, M. K. Mehlawat and L. Jia, “Mean-Variance portfolio optmization using machine learning-based stock price prediction,” *Applies Soft Computing Journal*, vol. 100, 2021.

[3] S. Carta, A. Ferreira, A. S. Poddo, D. R. Recupero and A. Sanna, “Multi-DQN: An ensemble of deep q-learning agents for stock market forcasting,” *Expert Systems with Applications*, vol. 164, 2021.

[4] S. Arslankaya and Ş. Toprak, “Makine öğrenmesi ve derin öğrenme algoritmalarını kullanarak hisse senedi fiyat tahmini,” *Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi*, vol. 13, no. 1, pp. 178-192, 2021.

[5] D. Wei, “Prediction of Stock Price Based on LSTM Neural Network,” *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, 2019, pp. 544-547.

[6] Ş. Sakarya and Ü. Yılmaz, “Derin öğrenme mimarisi kullanarak BİST30 indeksinin tahmini,” *Europan Journal of Educational & Scocial Sciences*, vol. 4, no. 2, pp. 106-121, 2019.

[7] A. Ghosh, S. Bose, G. Maji, N. C. Debnath and S. Sen, “Stock Price Prediction Using LSTM on Indian Share Market,” *Proceedings of 32nd ONternational Conference on Computer Applications in Industry and Engineering. EPiC Series in Computing*, 2019, pp. 101-110.

[8] Z. D. Akşehir and E. Kılıç, “Makine öğrenmesi teknikleri ile banka hisse senetlerinin fiyat tahmini,” *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, vol. 12, no.2, pp. 30-39, 2019.

[9] M. Hiransha, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, “NSE stock market prediction using deep-learning models,” *Procedia Computer Science*, vol. 132, pp. 1351-1362, 2018.

[10] W. K. Liu, and M. K. P. So, “A GARCH model with artificial neural networks,” *Information*, vol. 11, no. 10, 2020.

[11] M. Vijh, D. Chandola, V. A. Tikkiwal and A. Kumar, “Stock closing price prediction using machine learning techniques,” *Procedia Computer Science*, vol. 167, pp. 599-606, 2020.

[12] Z. Tan, Z. Yan and G. Zhu, “Stock selection with forest: an exploitation of excess return in the chinese stock market,” *Heliyon*, vol. 5 no. 8, 2019.

[13] K. Kaczmarczyk and M. Hernes, “Financial decision support using the supervised learning method based on random forests,” *Procedia Computer Science*, vol. 176, pp. 2802-2811, 2020.

[14] C. Ciner, “Do industry returns predict the stock market? A reprise using the random forest,” *The Quarterly Review of Economics and Finance*, vol. 72, 2018.

[15] M. Keskin ve A. Yücel , "BIST 100 Endeksi İle Altın Fiyatları İlişkisinin Yapay Sinir Ağları Yöntemiyle Belirlenmesi (1988-2020)", MANAS Sosyal Araştırmalar Dergisi, c. 11, sayı. 2, ss. 600-611, Nis. 2022.

[16] G. Şişmanoğlu , F. Koçer , M. A. Önde and O. K. Sahingoz , "Derin öğrenme yöntemleri ile borsada fiyat tahmini", *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, c. 9, sayı. 1, ss. 434-445, 2020.

[17] İ. Dalkıran and M. Ozan , "Derin Öğrenme Teknikleri Kullanılarak Borsadaki Hisse Değerlerinin Tahmin Edilmesi", Avrupa Bilim ve Teknoloji Dergisi, no. 39, pp. 143-148, 2022.

[18] M. Sarıkoç and M. Çelik , "Boyut İndirgeme Teknikleri ve LSTM Derin Öğrenme Ağı İle BIST100 Endeksi Fiyat Tahmini", Avrupa Bilim ve Teknoloji Dergisi, no. 34, pp. 519-524, 2022.

[19] Z. D. Akşehir and E. Kılıç , "Hisse Senedi Tahmininde Karşılaşılan Veri Dengesizliği Problemi için Yeni Bir Kural Tabanlı Yaklaşım ve 2D-CNN Modeli", Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, vol. 15, no. 1, pp. 6-13, 2022.

[20] Z. D. Akşehir and E. Kiliç, "How to Handle Data Imbalance and Feature Selection Problems in CNN-Based Stock Price Forecasting," in IEEE Access, vol. 10, pp. 31297-31305, 2022.

[21] M. A. Ozbayoglu, M. U. Gudelek and O. B. Sezer, “Deep learning for financial applications: a survey.” *Applied Soft Computing Journal*, vol. 93, 2020.

[22] [Available] https://tr.investing.com/.

[23] A. Subasi, “Chapter 3-Machine learning techniques,” in Practical Machine Learning for Data Analysis Using Python, 2020. 91-202.

[24] S. Jain and M. Kain, “Prediction for Stock Marketing Using Machine Learning,” International Journal on Recent and Innovation Trends in Computing and Communication, vol. 6, no. 4, pp. 131-135, 2018.

[25] P. Wang, T. Jiang, G. Fan and C. Dan, “Prediction of Torpedo Initial Velocity Based on Random Forests Regression,” 2015 7th Intenational Conference on Intelligent Human-Machine Systems and Cybernetics, 2015, vol. 1, pp 337-339.

[26] G. Li, M. Xiao and Y. Guo, “Application of Deep Learning in Stock Market Valuation Index Forecasting,” *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 2019, pp. 551-554.

[27] A. İ. Taş , P. Gülüm and G. Tulum , "Finansal Piyasalarda Hisse Fiyatlarının Derin Öğrenme ve Yapay Sinir Ağı Yöntemleri ile Tahmin Edilmesi; S&P 500 Endeksi Örneği", Düzce Üniversitesi Bilim ve Teknoloji Dergisi, c. 9, sayı. 3, ss. 446-460, 2021.

[28] Ö. Çetin and A. H. Isık, "Monthly electricity generatıon forecast in solar power plants with LSTM", *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 9, sayı. 6, ss. 55-64, 2021.

[29] P. Ahire, H. Lad, S. Parekh and S. Kabrawala, “LSTM based stock price prediction,” *International Journal of Creative Research Thoughts*, vol. 9 no. 2, pp. 5118-5122, 2021.

[30] S. Kumar and D. Ningombam, "Short-Term Forecasting of Stock Prices Using Long Short Term Memory," *2018 International Conference on Information Technology (ICIT)*, 2018, pp. 182-186.

[31] D. Reddy, H. Babu, K. Reddy and Y. Saileela, “Stock market analysis using LSTM in deep learning,” *International Journal of Engineering and Technical Research*, vol. V9, 2020.

[32] U. Demirel, H. Çam and R. Ünlü, “Predicting stock prices using machine learning methods and deep learning algorithms: the sample of the ıstanbul stock exchange,” *Gazi University Journal of Science*, vol. 34, pp. 63-82, 2021.

[33] S. Mehtab, J. Sen and S. Dasgupta, "Robust Analysis of Stock Price Time Series Using CNN and LSTM-Based Deep Learning Models," *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2020, pp. 1481-1486.

[34] A. Gilik, A. S. Ogrenci and A. Ozmen, “Air quality prediction using a hybrid deep learning architecture,” *Environmental Science and Pollution Research*, vol. 29, pp. 11920-11938, 2022.

[35] S. Mehtab and J, Sen, “Stock Price Prediction Using Convolutional Neural Network on a Multivariate Timeseries,” *Proceedings of the 3rd National Conference on Machine Learning and Artificial Intelligence*, New Delhi, INDIA, 2020.

[36] O. B. Sezer, M. U. Gudelek and A. M. Ozbayoglu, “Financial time series forecasting with deep learning: A systematic literature review: 2005-2019,” *Applied Soft Computing Journal*, vol. 90, 2020.