



Research Article

PREDICTION OF BIST PRICE INDICES: A COMPARATIVE STUDY BETWEEN TRADITIONAL AND DEEP LEARNING METHODSÖyküm Esra YİĞİT*¹, Selçuk ALP², Ersoy ÖZ³¹*Yildiz Technical University, Department of Statistics, ISTANBUL; ORCID: 0000-0001-7805-3979*²*Yildiz Technical University, Dept. of Industrial Eng., ISTANBUL; ORCID: 0000-0002-6545-4287*³*Yildiz Technical University, Department of Statistics, ISTANBUL; ORCID: 0000-0001-9087-434X***Received: 22.04.2020 Revised: 25.08.2020 Accepted: 03.09.2020****ABSTRACT**

Financial time series prediction is a challenging task due to the noisy, non-stationary and chaotic nature series. Traditional methods, especially autoregressive integrated moving average (ARIMA) has a wide range of application. With the rapid development of information technologies in the last two decades, various deep learning methods which are inspired by human brain that consists of inter-connected neurons have been proposed in order to improve the prediction performance of time series. As the data amount increases, these methods have been seen as an alternative for traditional ones having some important limitations. The main purpose of this study is to determine whether the deep learning methods outperform than traditional ARIMA method in predicting the BIST 30, BIST 50 and BIST 100 price indices. The prediction performance of ARIMA is compared against the prediction performances of Long Short-Term Memory and Gated-Recurrent Unit for each BIST price index. According to the root mean square evaluation metric, it is found that ARIMA models have better performance in predicting BIST 30, BIST 50 and BIST 100 indices than deep learning architectures.

Keywords: Deep learning, ARIMA, long short-term memory, gated-recurrent unit.**1. INTRODUCTION**

A time series is a sequence of data points $X = \{x_1, x_2, \dots, x_t\}$ that are measured over time where x_i represents a single data point at time t . In many different areas, the real-world data shows the time series behavior and forecasting the future based on past observations becomes an important task in scientific, commercial, industrial or economic decision-making [1]. Especially in finance, this task is noticeably difficult due to the noisy, non-stationary and chaotic nature of financial time series. Mainly two approaches based on statistical and computationally methods have been commonly used [2] in capturing the time series patterns.

In financial time series, statistical based methods including Holt-Winters [3], autoregressive integrated moving average (ARIMA) [4], and regression (e.g., linear, support vector and decision tree regressions) have been widely-used as a traditional methods. Due to the statistical properties, ARIMA method is known as the most popular one among them. The method combines both autoregressive (AR) and moving average (MA) processes to construct a composite time series

* Corresponding Author: e-mail: oeyigit@yildiz.edu.tr, tel: (212) 383 44 39

model [5]. Many important studies have been in attempt to predict financial time series using ARIMA models [6-9]. However, ARIMA is in the class of linear models and have some strict assumptions that should be satisfied in the process of constructing models. The non-capability of capturing the non-linear hidden patterns [10] in complex-structured financial time series makes ARIMA prone to the errors. Therefore, some computational techniques which are in the class of non-parametric models have been recently address the challenges of time series prediction and have been preferred instead of traditional methods.

Long Short-Term Memory (LSTM) is one of the memory-based architectures in deep learning and was firstly introduced by Hochreiter and Schmidhuber [11]. LSTM is an improved version of recurrent neural network (RNN). One of the main drawback of RNN is the non-capability in storing information over a long time period and its forecasting performance can be easily affected due to the vanishing gradient or exploding gradient. The exploding problem can be relatively easily addressed by enforcing a constraint over the gradient norm [12]. On the other hand, LSTM was designed to overcome the vanishing problem caused in RNN by re-parametrizing RNN. Thus, unlike RNN, LSTM has a capable of learning long-term dependencies. In recent years, the LSTM applications in financial time series data have dramatically increased (see a systematic literature review by [13]). Another improved RNN architecture that is address the problem of vanishing gradient is the Gated-Recurrent Unit (GRU). The main difference between LSTM and GRU architectures is the number of gates. While LSTM consists of three gates namely an input gate, a forget gate and an output gate; GRU has only two gates namely a reset gate and an update gate. Since GRU uses less training parameters, it is advantaged in terms of computational cost and memory usage. On the other hand, LSTM performs better with longer sequence datasets compared to GRU. As in LSTM, GRU method has been becoming one of the flexible and efficient RNN based-architecture used for financial time series [14,15].

Although an extensive literature deal with deep learning in predicting financial time series such as stock market prices or exchange rates at international level, limited studies are available for Turkish stock market. Especially for BIST 30, BIST 50 and BIST 100 price indices, to the best of our knowledge, there has been no previous attempt in comparing the prediction performances of models that are statistical-based and memory-based. This study compares the prediction performances of ARIMA, LSTM and GRU for three BIST price indices. ARIMA model is chosen as a representative of statistical-based prediction method due to the non-stationarity behavior of time series. LSTM and GRU are considered as a representative of memory-based models in deep learning.

The rest of this study is organized as follows: Section 2 includes some related works. Section 3 provides detailed descriptions about ARIMA, LSTM and GRU models. Materials including data sets' collection and descriptions are given in Section 4. The results of each model are presented separately in Chapter 5. Comparisons of models are presented in Section 6. Finally discussion and conclusion regarding the study are presented in Section 7.

2. RELATED WORKS

ARIMA models can be seen as the standard procedure used in time series for more than 50 years and still remain the great interest among researchers. In the study of [16], it was found that the best model was ARIMA(2,1,1) in the prediction performed with the Box-Jenkins methodology for the exchange rate series in Turkey covering the period 1991-2002. In the study by [17], ARIMA was used in modeling five different stock indices of emerging countries and it was shown the forecasting accuracy rate of ARIMA(2,1,0) was approximately 72% in BIST 100 index. In their study [18], BIST 100 daily data set for the period 2006-2012 was used in modeling with ARIMA and GARCH and it was found that the ARIMA yielded the best performance. It is worth to mention that different symmetric and asymmetric conditional variance models (e.g., ARCH, GARCH and EGARCH) have been proposed as an alternative approaches of ARIMA in

explaining the behaviors of time series such as leptokurtic, volatility clustering, long memory and leverage effect. Although these models often show better performance than ARIMA, it keeps being the first step in the process of modeling time series using these alternative approaches. The results obtained from the ARIMA are used as a baseline in evaluating whether the alternative approaches are needed and whether these approaches improve the prediction performance.

Among financial literature focused on deep learning, important studies have been conducted in time series prediction. By using deep learning methods, it is aimed to define meaningful patterns from huge time series and to obtain better results from machine learning algorithms [19]. In the study of [20], S&P 500 index from 1989 to 2015 was modeled with LSTM networks. In the study of [21], the performances of ARIMA, LSTM and GRU were compared in forecasting the Bitcoin's price data and was found that ARIMA gave the best performance in terms of different evaluation metrics. Another study on Bitcoin's price data was utilized by [22]. The study focused on ARIMA RNN and LSTM. It was found that both LSTM and RNN outperformed the ARIMA. The study by [5] used different financial time series from 1985 to 2018 and their results indicated that LSTM was superior to ARIMA. In the study by [23], data set which consists of National Stock Exchange minute-wise stock price between 2014 and 2015 was used. RNN, LSTM and Convolutional Neural Network (CNN) deep learning architectures were utilized and CNN was found as the best model. A steel price forecasting model was developed using LSTM in [24]. RNN, LSTM and GRU networks were compared in terms of their forecasting performances on Google stock price movements in [25].

3. METHODS

3.1. ARIMA model

In ARIMA models, it is assumed that the future value of a data point is a linear function of multiple past data points and random errors which are assumed to follow normal distribution with mean zero and constant variance σ^2 . The model has the following form:

$$x_t = c + \sum_{i=1}^p \phi_i x_{t-i} + \varepsilon_t + \sum_{i=0}^q \theta_i \varepsilon_{t-i} \quad (1)$$

where x_t and ε_t show the observed data point at time t , respectively. Integers p , d and q are the model parameters. $\phi_i > 0$ ($i = 1, 2, \dots, p$) and $\theta_i > 0$ ($j = 1, 2, \dots, q$) in which p and q denote the orders of autoregressive and moving average processes. To express the model, ARIMA (p, d, q) is generally used and d shows the order of difference to make the series stationary. When $p = 0$, the model shown with Equation 1 reduces to an MA model with order q . When $q = 0$, the model becomes an AR model with order p . If $p = q = d = 0$, then it means that the process is a white noise.

To construct an appropriate ARIMA model which is capable to well-capture the behavior of time series, the estimation of parameters plays an important role. Box-Jenkins methodology presents a practical approach to model the time series with ARIMA [10] and has mainly three steps that are performed by iteratively: (1) the model identification (2) the parameter estimation (3) the diagnostic checking. The model identification step involves the transformation of data in order to achieve stationary in time series. The most commonly used transformation is differencing. According to the stationarity tests such as Augmented Dickey-Fuller (ADF) or sample autocorrelation (ACC) and partial autocorrelation (PACC) functions, it is decided to order of differencing. The computed ACC and PACC of transformed series are adequately describes the appropriate model parameters p and q . With the chosen parameters, model is estimated by maximum likelihood (ML) method and t-test is employed for seeking the statistical significance of each estimated coefficient of parameter. To determine whether the derived model is adequate for forecasting, the diagnostic checking step is performed by handling the residuals generated by

the model. The model is treated as adequate if there are random patterns and independencies among residuals. Otherwise, the entire process is repeated until an adequate model is found.

3.2. LSTM Model

LSTM is a special kind of RNN architecture which was firstly used in language modeling [11]. As in RNN, the network is composed of neurons layers and input is propagated through layers for the purpose of making prediction. Mainly two problems (vanishing gradients and exploding gradients) arise during the RNN training caused by not setting the hyperparameters properly. When the weigh matrix values become so small, the gradient signal gets so small as to have no effect [26]. This leads a non-capable in learning the long-term dependencies in the time series. Thus, a model shows poorer performance in terms of accuracy than it is supposed to be. This is often called as vanishing gradients problem. In addition, if the weigh matrix values become so large, the gradient signal gets so large as to result overflow [25,26], which is called as the problem of exploding gradients. Overcoming the problem of exploding gradients is relatively easier than vanishing gradients problem. Instead of conventional neuron, LSTM uses one or more memory cells which can be controlled by input, output and forget gates [27]. In a LSTM memory cell, the same roles are assigned to input and output gates as in RNN. Unlike RNN, an extra gate, forget gate, is designed in order to decide the discarding information from the cell. Thus, a memory cell is responsible to remember or to forget the current state of the unit that is used or not used in the next time step [28].

Figure 1a shows a simple LSTM architecture. As it is illustrated, LSTM network consists of an input layer, one or more hidden layers and an output layer. The number of neurons in the input layer is as much as the number of explanatory variables in the model. The output space is the number of neurons contained in the output layer. The hidden layer(s) consists of memory cell(s). The input gate in a memory cell (i_t) decides which information flows into memory cell. When i_t outputs 0, the blocking is done to prevent flowing information to the next layer. The forget gate (f_t) decides the discarding information from the cell and outputs a vector that consists of values between 0 and 1. While the 0 output refers that the information is completely discard from the cell, the output of 1 refers that the information is completely remembered. Finally, the output gate (o_t) is responsible in deciding which information from the cell is used as the output [28,29].

The mathematical expressions of LSTM working are given below:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + V_{ic}c_{t-1}) \tag{2}$$

$$\tilde{c}_t = \tanh(W_{cx}x_t + W_{ch}h_{t-1}) \tag{3}$$

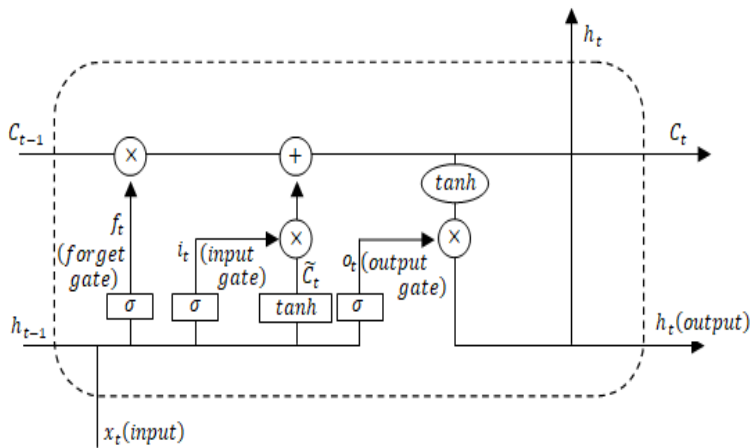
$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + V_{fc}c_{t-1}) \tag{4}$$

$$C_t = i_t * \tilde{c}_t + f_t * c_{t-1} \tag{5}$$

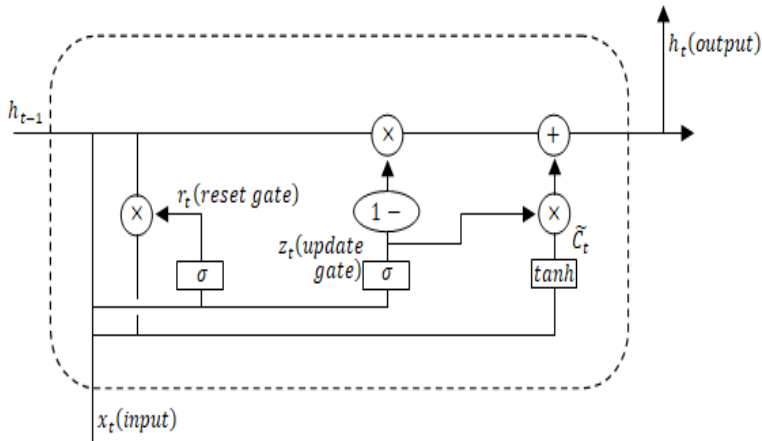
$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + V_{oc}c_t) \tag{6}$$

$$h_t = o_t \tanh(C_t) \tag{7}$$

where the input of the cell and output of the gate at time t are denoted as x_t and h_t , respectively. The terms W_{ix} , W_{ih} , W_{cx} , W_{ch} , W_{fx} , W_{fh} , W_{ox} , W_{oh} , V_{ic} , V_{fc} and V_{oc} denote corresponding weight matrices. $\sigma = 1/1 + e^x$ is the logistic sigmoid function. $*$ indicates the vectors' scalar product. By using Equation 2 and Equation 3, the values of input gate and a vector of candidate state values are calculated, respectively. The activation of the forget gate is given with Equation 4. Using the values of i_t , \tilde{c}_t and f_t , the value of memory cell is computed as in Equation 5. The output gate value is computed (given in Equation 6) after computation of memory cell state. Finally, the output gate value (h_t) is found with Equation 7. For the sake of simplicity in formulas, the bias vectors are ignored for i_t , f_t , C_t and o_t .



1a. Structure of LSTM cells



1b. Structure of GRU cells

Figure 1. Structure of LSTM and GRU NN cells

3.3. GRU model

GRU [30] is the other improved version of RNN to deal with vanishing and exploding gradients problems in RNN. The internal structure of GRU is less complicated than LSTM. Fewer computations in training phase makes GRU faster in computational time as compared with LSTM. Figure 1b shows a simple GRU architecture. GRU network consists of two gates namely a reset (r_t) and an update (z_t) and these gates are used in deciding which information will be passed to the output [21]. Specifically, the reset gate decides the amount of information that will be discarded from the cell, whereas the update gate is responsible in deciding the amount of information to be stored in the cell state. The mathematical expressions of GRU working are given below:

$$z_t = \sigma(W_{zx}x_t + W_{zh}h_{t-1}) \tag{8}$$

$$r_t = \sigma(W_{rx}x_t + W_{rh}h_{t-1}) \tag{9}$$

$$\check{c}_t = \tanh\left(W_{cx}x_t + (W_{ch}(r_t * h_{t-1}))\right) \tag{10}$$

$$h_t = (z_t * \check{c}_t) + ((1 - z_t) * h_{t-1}) \tag{11}$$

where the input of the cell and output of the gate at time t are denoted as x_t and h_t , respectively. The terms W_{zx} , W_{zh} , W_{rx} , W_{rh} , W_{cx} and W_{ch} denote corresponding weight matrices. σ is the logistic sigmoid function and $*$ indicates the vectors' scalar product. For the sake of simplicity in formulas, the bias vectors are ignored for z_t , r_t and \check{c}_t .

4. MATERIALS

4.1. Data collection

The sample consists of the historical daily BIST price indices obtained from both The Central Bank of Turkey electronic data distribution system and “Investing.com”. The data sets included BIST 30, BIST 50 and BIST 100 price indices in Borsa İstanbul (BIST). Table 1 shows the start and end dates of the indices available in the mentioned sources and used as data sets in this study.

Table 1. Periods of datasets

Index Name	Starting Date	Ending Date
BIST 30	January, 3, 1997	May, 14, 2020
BIST 50	January, 5, 2000	May, 14, 2020
BIST 100	January, 3, 1988	May, 14, 2020

4.2. Data description

Each time series dataset has a set of features namely “Open”, “High”, “Low”, “Close” and “Volume”. Since the features are highly-correlated, the “Close” feature is selected as only the feature of time series to be fed into the ARIMA, LSTM and GRU models. Some descriptive statistics of time series datasets are presented in Table 2. Based on the performed skewness [31] and kurtosis tests [32], the null hypotheses which state that the populations from which the samples are drawn are normally distributed can be rejected for all datasets. This indicates, all of three BIST indices exhibit right-skewed and platykurtic distributions.

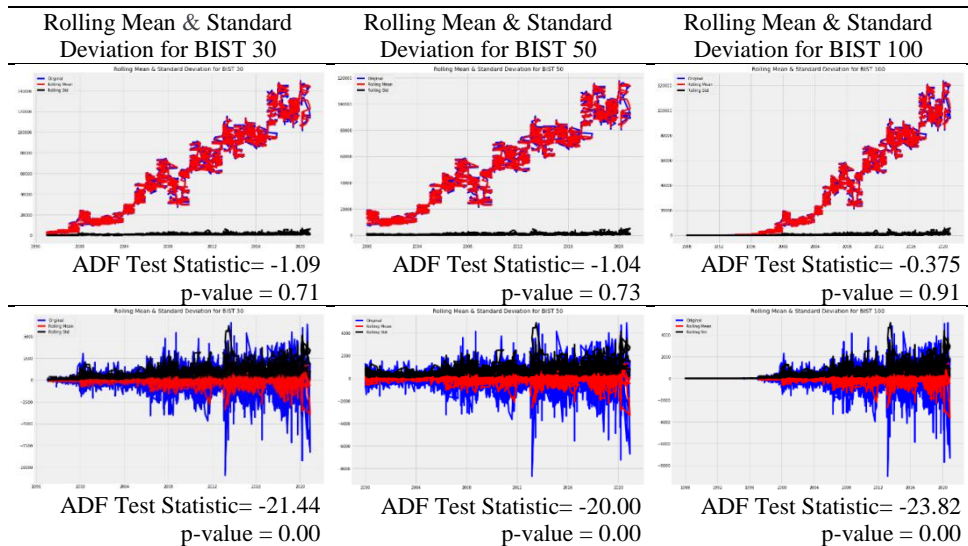
Table 2. Descriptive statistics of BIST indices

BIST Index Name	Number of Observations	Mean	Standard Deviation	Skewness	Kurtosis
BIST 30	5,843	58860.25	41441.98	0.23***	-1.18***
BIST 50	5,108	52362.64	30003.13	0.10***	-1.13***
BIST 100	8,236	33798.64	35695.63	0.68***	-0.91***

*** indicates significant at 1% significance level.

One of the models used in this study is the ARIMA which needs the stationarity of time series data. To check the stationary, two rolling statistics such as mean and standard deviation considering a window size equal to 5 are computed for each data set. Also, ADF tests are performed. In the first panel Figure 2, first, second and third columns shows the rolling statistics calculated for BIST 30, BIST 50 and BIST 100 indices, respectively. At the bottom of each figure, the results of ADF unit root test for each index series are given. As it is shown, calculated tau statistics are less than MacKinnon’s critical values in an absolute term and it can be concluded that the null hypotheses of non-stationary in the time series cannot be rejected at 1%, 5% and 10%

significance levels. To make the time series stationary, first difference is taken and then the rolling statistics and ADF tests are re-calculated (given in the second Panel of Figure 2). As it is demonstrated, after taking the first difference, the null hypothesis of unit root is rejected for each time series dataset at all significance levels. Hence, it can be concluded that all BIST price indices are first-order integrated.



Note: Mac Kinnon's critical values for 1%, 5% and 10% significance levels are -3.43, -2.86 and -2.56, respectively.

Figure 2. Rolling statistics

4.3. Evaluation metric

To compare the prediction performances of models, the root mean square error (RMSE) which is commonly used statistical-based metric is used. RMSE measures the differences between the observed values and values predicted by the model. A relatively high weight is given to large errors [33] and RMSE can be computed as follows:

$$RMSE = \sqrt{\frac{1}{N}(\sum_{i=1}^N(x_i - \hat{x}_i))^2} \tag{12}$$

where N is the total size of observations, x_i is the observed value and \hat{x}_i is the value predicted by the model. RMSE is the square root of MSE and the model which has the minimum RMSE value is considered as the best one. In evaluating the performance of a constructed model, splitting procedure in which gathered samples are divided into two sub-samples has been commonly used in machine learning applications. In this study, each time series is split into two sub-sets with a ratio of 80:20 according to the study of [20]. The training datasets are used as inputs for training the models, whereas testing datasets are used to evaluate the prediction performances of constructed models.

5. FITTING MODELS

5.1. ARIMA fitting

The best combination of hyperparameters (p, d, q) for the ARIMA models are searched by performing grid method in pre-defined ranges of parameters $(p = [0,4]; d = [0,2], q = [0,4])$. In evaluating the models with different hyperparameters, each is ranked against one another based on how the model is accurately predict the future. The metric used in predicting competitions among ARIMA models is the symmetric mean absolute percentage error (SMAPE). The SMAPE can be defined by:

$$SMAPE = \frac{1}{N} \sum_{i=1}^N \frac{|x_i - \hat{x}_i|}{(|x_i| + |\hat{x}_i|)/2} \times 100 \tag{13}$$

where x_i and \hat{x}_i denote the observed and one-step ahead forecast values, respectively. When a model fit the data perfectly, yet the SMAPE takes the value of 0%. Table 3 shows the best fitted ARIMA model by the grid search for each time series dataset.

Table 3. Best fitted ARIMA models

Index Name	Best Fitted ARIMA(p, d, q)	SMAPE
BIST 30	ARIMA(2,1,0)	16.45
BIST 50	ARIMA(0,1,3)	14.68
BIST 100	ARIMA(3,1,1)	17.80

As it is shown, the best fitted models are ARIMA(2,1,0), ARIMA(0,1,3) and ARIMA(3,1,1) for BIST 30, BIST 50 and BIST 100 indices, respectively. In addition, how well the predictions matched up with the testing datasets for the observed values are shown with Figure 3.

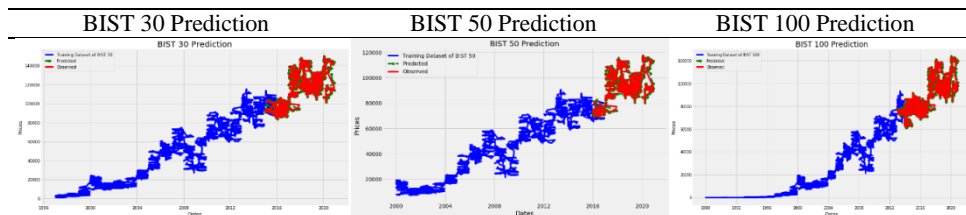


Figure 3. Prediction with best fitted ARIMA

5.2. LSTM fitting

As in ARIMA, the raw data is needed to be transformed and get ready to be modeled before performing LSTM. Data pre-processing is an important step in machine learning applications to transform the raw data into understandable format. Different methods such as normalization, aggregation and sampling can be used for different purposes (e.g., to accelerate the iteration convergence). Among these pre-processing methods, data normalization is the most commonly used in both classification and regression tasks [34]. This study focuses on min-max normalization (MMN) [35] to re-scale the data sets within the range of 0 to 1. The equation for the MMN is given as follows:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{14}$$

where $\min(\cdot)$ and $\max(\cdot)$ shows the maximum and minimum values of observed values in a dataset x , respectively. z is the normalized value.

Each dataset is transformed by using MMN. LSTM network implementation is performed using Tensorflow library in Python which is an open-source platform designed by Google [36]. In the LSTM network, the number of input layer is set at 1 since each dataset has a “Close” feature. Four hidden layers of LSTM with size of 50 units are used. A sliding window of 60 time steps is used for predictions since the LSTM has the ability of store long term memory state. Thus, the number of previous training set elements for each training set element is 60. Adam optimization algorithm [37] which is an extended version of stochastic gradient descent (SGD) and mean square error (MSE) loss function are used in compiling the LSTM model. Learning rate of the Adam algorithm is set at 0.01. The model is trained on for 50 epochs having 32 batch sizes. Figure 4 illustrates how well the predictions matched up with the testing datasets of BIST 30, BIST 50 and BIST 100 price indices.

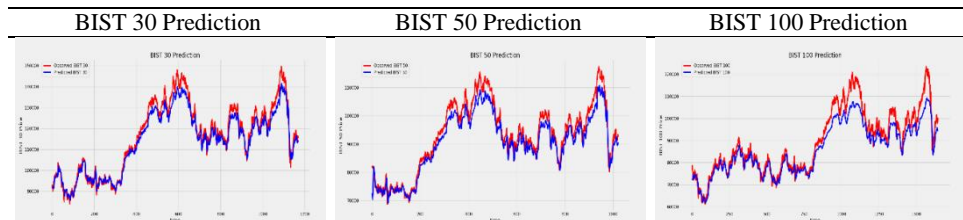


Figure 4. Prediction with LSTM

5.3. GRU fitting

When modeling the BIST price indices with GRU, each dataset is transformed by using MMN as in LSTM. Also, the GRU network implementation is performed by Python/Tensorflow library. The number of input layers, hidden layers, time steps, epochs and batch sizes are chosen as the same values as those given in LSTM. Adam algorithm with learning rate 0.01 is used in compiling the GRU model. In addition, MSE loss function is used as the loss function of output layer. Figure 5 illustrates how well the predictions matched up with the testing datasets of BIST 30, BIST 50 and BIST 100 price indices.

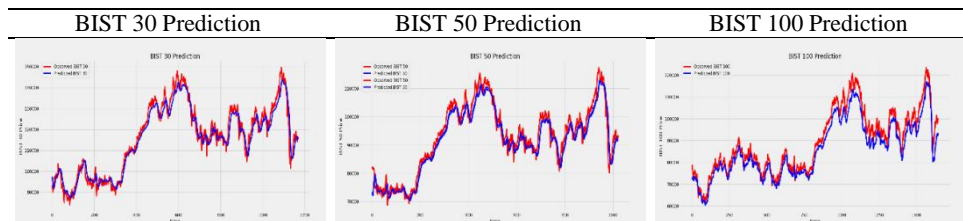


Figure 5. Prediction with GRU

6. MODELS COMPARISON RESULTS

Table 4 present the RMSE values for ARIMA, LSTM and GRU models fitting. The last column of Table 4 gives the RMSE-difference between the best and the worst model in terms of RMSE for each BIST index. As it is shown, the traditional approach, ARIMA, gives better results when compared with deep-learning based methods, LSTM and GRU. Among deep learning architectures, while the performance of GRU is better than LSTM for BIST 50 and BIST 100; the LSTM is favored than GRU for BIST 30.

Table 4. Summary of the results

BIST Index Name	ARIMA	LSTM	GRU	RMSE-Difference
BIST 30	1626.36	3518.01	3660.06	2033.7
BIST 50	1275.54	3308.53	3229.17	2032.99
BIST 100	1218.15	4891.02	4288.69	3672.87

7. DISCUSSION AND CONCLUSION

The stock investment is known as one of the main investment tools in the financial markets. Different indices in Borsa İstanbul have been calculated for a long time in order to be useful guides to the investors who carefully follow market movements. These indices make easier to understand the current status of the market. Also, they are used as benchmarks by the financial industry. In general, investors desire to face minimum risk while making their decisions. However, more risky investment tools might be more profitable since the higher risk may lead to higher profit. Thus, the main research question in making investing decision is how risk can be reduced and how profit can be increased? It is not possible for this question to be answered precisely. However, different traditional and machine learning methods can effectively improve the prediction precision and the outputs generated from these methods can help for investors in avoiding risks and improving benefit [38].

The main purpose of this study is to investigate the performances of traditional and deep learning methods in BIST 30, BIST 50 and BIST 100 price indices prediction. The prediction performance of ARIMA, as a traditional method, is compared against the LSTM and GRU, as deep learning methods. Obtained results show that ARIMA outperforms LSTM and GRU in terms of RMSE. This result may have different causes. First of all, this study employs an ARIMA model for one-step ahead forecasting on univariate datasets. Secondly, the amount of data might be not enough being in the category of "massive". As stated in [39,40], ARIMA generate better performance than deep learning methods specially when the researchers deal with univariate time series. Deep learning methods, on the other hand, have the capability of coping with multivariate time series. Also, deep learning methods deliver on their promise for massive datasets. In the future studies, it is planned to study with massive datasets (e.g., minute-wise index price). In addition, future studies are planned to be focused on extracting various uncorrelated features from the raw data and hence studying with multivariate structure.

Acknowledgements

The authors would like to thank anonymous reviewers for their valuable comments and suggestions to improve the paper.

REFERENCES

- [1] Cortez, P., and Donate, J. P. (2014) Global and decomposition evolutionary support vector machine approaches for time series forecasting, *Neural Computing and Applications* 25(5), 1053-1062.
- [2] Jothimani, D., and Yadav, S. S. (2019) Stock trading decisions using ensemble-based forecasting models: a study of the Indian stock market, *Journal of Banking and Financial Technology* 3(2), 113-129.
- [3] Winters, P. R. (1960) Forecasting sales by exponentially weighted moving averages, *Management Science* 6(3), 324-342.
- [4] Box, G.E.P, and Jenkins, G. (1976) Time series analysis: forecasting and control. *Holden Day*, San Francisco, USA.

- [5] Siami-Namini, S., and Namin, A. S. (2018) Forecasting economics and financial time series: ARIMA vs. LSTM. *arXiv preprint arXiv:1803.06386*.
- [6] Meyler, A., Kenny, G., and Quinn, T. (1998) Forecasting Irish inflation using ARIMA models, MPRA Paper No 11359.
- [7] Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. (2014, March). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation* (pp. 106-112). IEEE.
- [8] Junior, P. R., Salomon, F. L. R., and de Oliveira Pamplona, E. (2014) ARIMA: An applied time series forecasting model for the Bovespa stock index, *Applied Mathematics* 5(21), 3383-3391.
- [9] Gay Jr, R. D. (2016) Effect of macroeconomic variables on stock market returns for four emerging economies: Brazil, Russia, India, and China, *International Business & Economics Research Journal (IBER)* 15(3), 119-126.
- [10] Zhang, G. P. (2003) Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing* 50, 159-175.
- [11] Hochreiter, S., and Schmidhuber, J. (1997) Long short-term memory, *Neural Computation* 9(8), 1735-1780.
- [12] Pascanu, R., Mikolov, T., and Bengio, Y. (2013, February). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310-1318).
- [13] Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. (2020) Financial time series forecasting with deep learning: A systematic literature review: 2005–2019, *Applied Soft Computing* 90, 106181.
- [14] Di Persio, L., and Honchar, O. (2016) Artificial neural networks architectures for stock price prediction: Comparisons and applications, *International journal of circuits, systems and signal processing* 10(2016), 403-413.
- [15] Lee, S. I., and Yoo, S. J. (2018) Threshold-based portfolio: the role of the threshold and its applications, *The Journal of Supercomputing* 2018, 1-18.
- [16] Bircan, H., and Karagöz, Y. (2003) Box-Jenkins Modelleri ile Aylık Döviz Kuru Tahmini Üzerine Bir Uygulama, *Kocaeli Üniversitesi Sosyal Bilimler Enstitüsü Dergisi* (6) 2003/2, 49-62.
- [17] Yenice, S., and Tekindal, M. A. (2015) Forecasting the stock indexes of fragile five countries through Box-Jenkins methods, *International Journal of Business and Social Science* 6(8), 180-191.
- [18] Sekreter, A., and GURSOY, A. (2014) Combining forecasting method vs. individual forecasting methods: Evidence from Istanbul Stock Exchange National 100 Index, *The Empirical Economics Letters* 13(7), 735-743.
- [19] Sezer, Ö. B. (2018) Zaman serisi verilerinin derin yapay sinir ağları ile analizi ve eniyilemesi: Finansal tahmin algoritmaları. PhD Thesis, *Graduate School of Engineering and Science, TOBB Ekonomi ve Teknoloji Üniversitesi, Ankara, Turkey*.
- [20] Fischer, T., and Krauss, C. (2018) Deep learning with long short-term memory networks for financial market predictions, *European Journal of Operational Research* 270(2), 654-669.
- [21] Yamak, P. T., Yujian, L., and Gadosey, P. K. (2019, December). A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence* (pp. 49-55).
- [22] McNally, S., Roche, J., and Caton, S. (2018, March). Predicting the price of bitcoin using machine learning. In *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)* (pp. 339-343). IEEE.

- [23] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., and Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 1643-1647). IEEE.
- [24] Çetin, K., Aksoy, S., and İşeri, İ. (2019, September). Steel Price Forecasting Using Long Short-Term Memory Network Model. In *2019 4th International Conference on Computer Science and Engineering (UBMK)* (pp. 612-617). IEEE.
- [25] Di Persio, L., and Honchar, O. (2017). Recurrent neural networks approach to the financial forecast of Google assets. *International journal of Mathematics and Computers in simulation*, 11.
- [26] Brownlee, J. (2017). *Long Short-term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning*. Machine Learning Mastery.
- [27] Bsir, B., and Zrigui, M. (2018, September). Bidirectional LSTM for author gender identification. In *International Conference on Computational Collective Intelligence* (pp. 393-402). Springer, Cham.
- [28] Yu, L., Chen, J., Ding, G., Tu, Y., Yang, J., and Sun, J. (2018). Spectrum prediction based on Taguchi method in deep learning with long short-term memory. *IEEE Access*, 6, 45923-45933.
- [29] Soutner, D., and Müller, L. (2013, September). Application of LSTM neural networks in language modelling. In *International Conference on Text, Speech and Dialogue* (pp. 105-112). Springer, Berlin, Heidelberg.
- [30] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- [31] R. B. D'Agostino, A. J. Belanger and R. B. D'Agostino Jr., "A suggestion for using powerful and informative tests of normality", *American Statistician* 44, pp. 316-321, 1990.
- [32] Anscombe, F. J., and Glynn, W. J. (1983) Distribution of the kurtosis statistic b_2 for normal samples, *Biometrika* 70(1), 227-234.
- [33] Saigal, S., and Mehrotra, D. (2012) Performance comparison of time series data using predictive data mining techniques, *Advances in Information Mining* 4(1), 57-66.
- [34] Jain, S., Shukla, S., and Wadhvani, R. (2018) Dynamic selection of normalization techniques using data complexity measures, *Expert Systems with Applications* 106, 252-262.
- [35] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- [36] Abadi, Martin, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "TensorFlow: A System for LargeScale Machine Learning." In OSDI, vol. 16, pp. 265-283. 2016.
- [37] Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [38] Luo, H., and Wang, S. (2016) Based on the PCA-ARIMA-BP hybrid model of stock price prediction research, *ANZIAM Journal* 58, 162-178.
- [39] Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018), Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3).
- [40] Jiang, L., Rollins, K. M., Ludlow, M., and Sadler, B. (2020). Demand Forecasting for Alcoholic Beverage Distribution. *SMU Data Science Review*, 3(1), 5.