Research Article / Araştırma Makalesi

Ekonomik ve Sosyal Araştırmalar Dergisi

The International Journal of Economic and Social Research

2024, 20(2)

Evaluation of BIST100 Index Prediction Performance of Deep and Machine Learning Algorithms

Derin ve Makine Öğrenme Algoritmalarının BIST100 Endeksi Tahmin Performanslarının Değerlendirilmesi

Yunus Emre GÜR[1]

Geliş Tarihi (*Received*): 25.09.2023 **Kabul Tarihi (*Accepted*):** 28.06.2024 **Yayın Tarihi (*Published*):** 30.12.2024

Abstract: This study investigates the possibility of forecasting the Borsa Istanbul BIST 100 index using machine learning and deep learning techniques. The study uses the BIST 100 index as the dependent variable. In addition, gram gold price, daily dollar exchange rate (in TL), daily euro exchange rate (in TL), BIST trading volume, daily Brent oil prices, BIST trading volume, BIST overnight repo rates, and BIST Industrial Index (XUSIN) data are used as independent variables. The Central Bank of the Republic of Turkey provides daily statistics on these variables. The performance of several deep learning recurrent neural networks (RNN) and machine learning network structures—including Random Forest, K-Nearest Neighbors, Multilayer Perceptron, Radial Basis Function, and Support Vector Machine—for predicting the BIST 100 index is tested and compared in this study. The results indicate that the CNN model outperforms the other models in terms of prediction accuracy, with the lowest RMSE and MSE values, and the highest R^2 value. This suggests that CNN is a robust model for financial forecasting. The relevant literature is summarized in this context in the first portion of the study, after which the methods and results are described. Then the obtained comparative prediction values are presented. Finally, the study is concluded by presenting the interpretations of the results and recommendations.
Keywords: Machine Learning, Deep Learning, Financial Forecasting, BIST100 Index, CNN Algorithm

&

Özet: Bu çalışma, Borsa İstanbul BIST 100 endeksinin makine öğrenmesi ve derin öğrenme teknikleri kullanılarak tahmin edilme olasılığını araştırmaktadır. Çalışmada bağımlı değişken olarak BIST 100 endeksi kullanılmıştır. Ayrıca gram altın fiyatı, günlük dolar kuru (TL cinsinden), günlük euro kuru (TL cinsinden), BIST işlem hacmi, günlük Brent petrol fiyatları, BIST işlem hacmi, BIST gecelik repo faizleri ve BIST Sanayi Endeksi (XUSIN) verileri bağımsız değişkenler olarak kullanılmıştır. Türkiye Cumhuriyet Merkez Bankası bu değişkenlere ilişkin günlük istatistikleri sağlamaktadır. Bu çalışmada, Rastgele Orman, K-En Yakın Komşular, Çok Katmanlı Algılayıcı, Radyal Taban Fonksiyonu ve Destek Vektör Makinesi dahil olmak üzere çeşitli derin öğrenme tekrarlayan sinir ağları (RNN) ve makine öğrenimi ağ yapılarının BIST 100 endeksini tahmin etme performansı test edilmiş ve karşılaştırılmıştır. Sonuçlar, CNN modelinin en düşük RMSE ve MSE değerleri ve en yüksek R^2 değeri ile tahmin doğruluğu açısından diğer modellerden daha iyi performans gösterdiğini ortaya koymaktadır. Bu da CNN'in finansal tahmin için sağlam bir model olduğunu göstermektedir. Çalışmanın ilk bölümünde bu bağlamda ilgili literatür özetlenmiş, ardından yöntem ve sonuçlar açıklanmıştır. Daha sonra elde edilen karşılaştırmalı tahmin değerleri sunulmuştur. Son olarak, sonuçlara ilişkin yorumlar ve öneriler sunularak çalışma sonlandırılmıştır.
Anahtar Kelimeler: Makine Öğrenimi, Derin Öğrenme, Finansal Tahmin, BIST100 Endeksi, CNN Algoritması

Atıf/Cite as: Gür, Y. E. (2024). Evaluation of BIST100 Index Prediction Performance of Deep and Machine Learning Algorithms. *Ekonomik ve Sosyal Araştırmalar Dergisi*, 20(2), 394- 408.

İntihal-Plagiarism/Etik-Ethic: Bu makale, en az iki hakem tarafından incelenmiş ve intihal içermediği, araştırma ve yayın etiğine uyulduğu teyit edilmiştir. / This article has been reviewed by at least two referees and it has been confirmed that it is plagiarism-free and complies with research and publication ethics. https://dergipark.org.tr/tr/pub/ijaws

[1] Dr. Öğretim Üyesi, Yunus Emre Gür, Fırat Üniversitesi, Yönetim Bilişim Sistemleri Bölümü, yegur@firat.edu.tr (Sorumlu Yazar), https://orcid.org/0000-0001-6530-0598

1. Introduction

Nowadays, the rapid access to information about financial markets and the development of the instruments used in them attract the attention of many people. However, the complexity and unpredictable nature of financial markets can carry great risks for investors without financial experience, which emphasizes the importance of decision support systems in financial markets. Therefore, the analysis of financial data has become a popular research topic (Bulut, 2024). Its main purpose is to predict the effects of market conditions and their future directions, especially the forecasting of financial time series, which forms the basis for making decisions in the financial asset market, and this has attracted the interest of many researchers (Sarıkoç and Çelik, 2022: 520).

Forecasting indices in capital markets can help create more effective investment strategies. However, the complex dynamics of stock market indices make it difficult to predict their return values or direction of movement because these variables depend on many factors (Simsek, 2024b). Unexpected events are one of the factors that can lead to fluctuations in stock market indices (Alaca and Güran, 2022: 380). Individual and institutional investors want to obtain information about the securities and other investment instruments they own or are interested in for future planning. Therefore, it is of great importance to predict the future performance of these investment instruments.

However, since the BIST100 index is closely associated with Turkey's economic performance, forecasting its future price movements has always been an attractive topic in the financial literature. Many models have been developed for the forecasting process. While traditional methods such as simple regression models, Box-Jenkins (ARMA) models, and VAR (vector auto-regressive) models based on with the development of technology, flexible forecasting methods like artificial neural networks, fuzzy logic, and genetic algorithms are now often employed (Kantar, 2020: 122). In recent years, beyond traditional statistical methods, analysis and forecasting models based on deep learning and machine learning have achieved impressive success (Simsek, 2024a). These models have shown significant success in time series and similar sequential learning tasks (Sarıkoç and Çelik, 2022: 520).

In this study, a total of 425 daily Borsa Istanbul BIST 100 index data between January 2022 and September 2023 was forecasted using deep learning and machine learning algorithms. In this direction, it is trying to reveal which system is successful in the forecasting process. For this purpose, the independent variables to be used as input in the algorithms for the prediction process were first determined. The scholarly literature on the elements influencing the BIST 100 index was examined to identify the independent variables. The research revealed that there is a significant correlation between gold prices, BIST Industrial Index (XUSIN), exchange rate, oil price, BIST Trading Volume, Trading Amount, and interest rates and BIST 100 index (Karcıoğlu and Özer, 2014; Sevinç, 2014; Aydın and Çavdar, 2015; Alper and Kara, 2017; Koyuncu, 2018). Daily data on these factors is obtained from the Central Bank of the Republic of Turkey and included in the study as independent variables.

Within the study, the BIST100 index is used as an independent variable. In addition, gram gold price, daily dollar exchange rate (in TL), daily euro exchange rate (in TL), BIST trading volume, daily Brent oil prices, BIST trading quantity, BIST overnight repo rates, and data from the BIST Industrial Index (XUSIN) were used as the independent variables. In the research process, different deep learning recurrent neural networks (RNN) (Long and Short Term Memory-LSTM, Gated Recurrent Unit-GRU, Convolutional Neural Network-CNN) and machine learning (Random Forest-RF, K-Nearest Neighbors-KNN, Multilayer Perceptron-MLP, Radial Basis Function-RBF, and Support Vector Machine-SVM) the accuracy of various network architectures in foretelling the BIST 100 index was examined and compared. In this framework, in the first stage of the study, after summarizing the related literature, the methodology and the data used are explained. Then, the obtained comparative forecasting values are presented. Finally, the study is completed by presenting the interpretation of the results and recommendations.

2. Literature Review

In this section, some of the studies using LSTM, GRU, CNN, RF, KNN, MLP, RBF, and SVM algorithms as methods for model identification and estimation are summarized below.

Singular Spectrum Analysis (SSA) was utilized by Fenghua et al. (2014) to evaluate stock prices with the goal of breaking down trends, market volatility, and noise into various economic aspects and time periods. They then include these traits using the support vector machine technique to forecast pricing. The experimental findings showed that combination forecasting approaches (EEMD-SVM and SSA-SVM) performed better than SVM without incorporating these pricing variables, while SSA-SVM produced the best predicting outcomes.

Heo and Yang (2016) conducted their research using SVM, a machine learning method. One sort of fundamental analysis used in the study is to forecast stock prices using the company's intrinsic value. The company's financial statements served as the SVM's input. The stock's future direction was forecasted based on these findings. In comparison to expert forecasts, the findings demonstrated that SVM had better predicting abilities.

For the Standard & Poor's 500 (S&P 500) Index, Sethia and Raut (2019) sought to create the best model that can forecast stock prices five days in advance. The best model was selected after the performance of the ANN, GRU, LSTM, and SVM models were compared. The forecasting and training data were gathered from 2000 to 2017 during a 12-year span. The open, high, low, close, and volume (OHLCV) data for the S&P 500 were also provided, along with 50 technical indicator-based features. The Independent Component Analysis (ICA) method used Minimax to decrease and scale each dimension after converting each feature value into a relative standard score. The effectiveness of the various models on this dataset was contrasted using clearly specified metrics. Conclusions demonstrate that the LSTM model outperforms the competition and has an R^2 value of 0.9486, 400% more than the hold-and-wait strategy's rate of return.

iShares MSCI UK exchange-traded fund daily closing price data between January 2015 and June 2018 were utilized by Nikou et al. (2019), who sought to assess the stock market's capacity for predicting. Using models with four machine learning algorithms, the forecasting procedure was conducted. The outcomes demonstrated that the support vector regression approach ranks second with less error than the neural network and random forest methods, and that the deep learning method has higher predicting capacity than the other methods.

Using monthly data for the years 2009–2019, Kantar (2020) undertook a study to link and forecast the BIST100 index with macroeconomic indicators. Both the ARMA (1,1) model and the artificial neural network model were used in the study to achieve this. The research included a variety of input variables, including gold, a basket of currency rates, deposit rates, emission rates, direct capital investments, portfolio investments, and an index of industrial production. The results of the study showed that, in terms of performance prediction, the Artificial Neural Networks model beat the ARMA (1,1) model. Additionally, according to the study's findings, the artificial neural network model is more effective at predicting how the BIST100 index would perform in the future.

According to a research by Vijh et al. (2020), it is challenging to estimate stock returns because of how unpredictable and non-linear stock markets are. Five distinct firms from various industries were employed in the study, and the closing prices of those companies were predicted using Artificial Neural Network and Random Forest methodologies. Stock financial data, such as the Opening, High, Low, and Closing prices, are utilized to generate new variables, which are then fed into the models as input. The effectiveness of the models is evaluated using common strategic metrics like RMSE and MAPE. The results show that artificial neural networks are more accurate in predicting stock prices.

Mehtab et al. (2020) illustrated regression models based on deep learning using historical stock prices of a well-known company listed on the National Stock Exchange of India. The study employed a timeframe (December 31, 2012-January 9, 2015) with data captured at five-minute time intervals and used extremely comprehensive stock price data. With the use of this incredibly thorough stock price data, five deep learning models were developed, four of which used convolutional neural networks (CNNs), while the other two relied on long- and short-term memory (LSTM). The results of the investigation demonstrated that in terms of prediction precision and execution speed, CNN-based models performed better than their LSTM equivalents.

Kemalbay and Alkış (2020) conducted research to predict the direction of the BIST100 index's daily movement. Models for multiple logistic regression and the K-nearest neighbor method were constructed using statistically significant independent variables. Out-of-sample forecasts were produced to assess the outcomes, and the forecasting performance was assessed using the accuracy rate and other statistical measures. The results show that logistic regression analysis surpasses the K-nearest neighbor approach in terms of predicting ability, with an accuracy rate of 81%, using BIST100 data during the defined time period.

Mehtab and Sen (2020) introduced a deep learning-based model based on historical NIFTY 50 index data posted on the National Stock Exchange of India. The recommended strategy included two regression models based on convolutional neural networks (CNN), two forecasting models based on CNN, and three forecasting models based on long- and short-term memory (LSTM). A multi-step forecasting methodology was used to predict the NIFTY 50 index opening values. Using this technique, forecasts were made for the NIFTY 50 index's opening values for a week, the model was retrained after that week, the actual index values were added to the training set, and forecasts were then made for the next week. It can be said that all of the offered models are excellent in predicting NIFTY 50 opening prices. The univariate encoder-decoder convolutional LSTM model was more accurate than the other models, although not by much. It was also found that the univariate CNN model handled data the quickest.

The use of deep learning techniques in financial markets was studied by Sarıkoç and Çelik in 2022. To forecast future values of the Borsa Istanbul 100 (BIST100) index, they specifically deployed a deep learning network. Additionally, they investigated the effects of employing principal component analysis, independent component analysis, and factor analysis during the data preprocessing stage on the performance of this forecasting model. The study assessed the models' capacities to forecast the index's price five days in advance using these various dimensionality reduction techniques. The data analysis showed that the PCA+LSTM hybrid model performed better than the other methods and improved the R^2 and RMSE by 4.60 and 13.35 percent, respectively.

Wenjie et al. (2022) proposed a unique composite forecasting model to predict the closing price of the following trading day. This innovative model is evaluated by comparing it to well-known models like CNN-Attention-GRU, CNN-Attention, CNN-GRU-Attention, and LSTM-Attention. The outcomes of the experiments demonstrate that this model surpasses the competition and excels in assessment criteria including MAE, RMSE, and R^2.

Kurani et al. studied the two main stock forecasting methods, ANN and SVM, in their work from 2023. The study investigated numerous stock market forecasting technologies and assessed the use of techniques including moving average algorithms, decision trees, sentiment analysis, and data mining in various forecasting situations. Additionally, new research utilizing ANN and SVM is discussed, and it is observed that these algorithms perform better with hybrid models (such as ANN-MLP and GARCH-MLP).

Mukherjee et al. (2023) used CNN and Deep Feed Forward Neural Networks, two well-known network models, to predict stock market prices. The models were used to predict the values of the data for the forthcoming days based on the data from the previous few days. Deep learning was used in an effort to optimize these predictions, and substantial outcomes were attained. The CNN model has a 98.92% accuracy rating compared to the ANN model's 97.66%.

A clustering-assisted deep learning architecture was proposed by Li et al. (2023) to increase the accuracy of stock price prediction. Three well-known deep learning prediction models—GRU, RNN, and LSTM—were merged into the recommended architecture. The study's pre-prediction process of clustering improved the quality of the training models. The Logistic Weighted Dynamic Time Warping similarity metric is suggested for effective clustering. This metric, which reflects the relative significance of return observations for creating distance matrices, is produced by expanding the Weighted Dynamic Time Warping approach. A logistic probability density distribution function, which is exactly based on stock return distributions, is used in place of WDTW's cost weighting function. On the three deep learning models mentioned earlier, the clustering-based forecasting method was used. The recommended method, which combines logistic WDTW clustering and the LSTM model, performs well in forecasting using five distinct assessment measures, according to extensive research on daily US stock price datasets.

In his research, Yan (2023) used a deep learning model with AdaBoost feature selection to forecast the values of stock index futures. The proposed hybrid model constructs a two-layer long short-term memory-based predictor and employs an AdaBoost regressor wrapped in sklearn for feature selection, to be more exact. According to performance measurements, the suggested model regularly outperforms other well-known prediction models including random forests, multilayer perceptrons, gated recurrent units, deep belief networks, and overlapping noise-reducing autocoders.

3. Data and Methodology

This study uses 425 days of data from January 2022 to June 2023, including the BIST 100 index, the dollar/Turkish Lira exchange rate, the euro/Turkish Lira exchange rate, the oil price, the BIST trading volume, the BIST overnight repo rates, BIST trading quantity, and the BIST Industrial Index (XUSIN). The data were obtained through the Central Bank's EDS system. The seven independent variables described above are used to forecast the BIST 100 index. Table 1 also includes a portion of the data set utilized in this investigation.

Table 1: Part of the Dataset Used in the Study

Date	BIST100	Daily Dollar (TL)	Gold	Oil Price	BIST Trading Volume	XUSIN	BIST Trading Amount	Daily Euro (TL)	BIST Overnight Repo Rates
03-01-2022	1926.66	13.33	1800.1	78.25	27711632.40	3525.02	3560861.11	15.09	14.07
04-01-2022	1963.59	13.42	1814.6	79.39	40062590.14	3587.93	5001149.09	15.23	14.02
05-01-2022	2005.29	13.19	1825.1	80.60	34742043.89	3657.45	4011892.90	14.89	13.61
06-01-2022	2007.55	13.37	1789.2	81.99	39155757.21	3648.74	4538272.41	15.11	13.92
07-01-2022	2033.32	13.64	1797.4	82.28	40606641.73	3676.23	4857956.33	15.41	13.94
10-01-2022	2045.16	13.75	1798.8	81.56	40757346.11	3690.29	4649894.16	15.55	14.45
11-01-2022	2046.03	13.72	1818.5	84.98	48669488.56	3706.20	5446042.37	15.54	14.78
…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…

…	…	…	…	…	…	…	…	…	…
12-09-2023	8159.90	26.84	1948.9	92.97	202947877.18	13201.71	9874701.85	28.87	26.42
13-09-2023	8013.89	26.86	1946.6	93.97	194411620.34	12927.44	9296353.57	28.89	26.35
14-09-2023	8118.75	26.90	1944	94.97	191931859.36	13047.72	8856907.15	28.88	26.30
15-09-2023	7961.98	26.91	1935.2	94.27	188056307.48	12951.40	8435713.12	28.80	26.41

Resources: evds2.tcmb.gov.tr

Eight different methodologies were used in the investigation: LSTM, GRU, RF, CNN, KNN, MLP, RBF, and SVM models. Indicators of estimated statistical significance such as coefficient of determination (R^2), mean square error (MSE), mean absolute percentage error (MAPE), mean absolute error (MAE), and mean absolute percent error (MAPE) were employed to assess the performance of the models. Equations 1, 2, 3, 4, and 5 may be utilized to calculate these statistical data. However, the hyperparameter adjustments of all models used in the study are shown in Table 2.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)}{N}} \quad (1)$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (2)$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \quad (3)$$

$$MAPE = \frac{\sum_{t=1}^{n}\frac{u_t}{Y_t}*100}{n} \quad (4)$$

$$R^2 = 1 - \frac{\sum_i(y_i - \hat{y}_i)^2}{\sum_i(y_i - \mu)^2} \quad (5)$$

Table 2: Hyperparameter Settings of All Models

LSTM Model	**GRU Model**	**RF Model**	**CNN Model**	**KNN Model**	**MLP Model**	**RBF Model**	**SVM Model**
LSTM Layer: Units: 50 Activation : ReLU Dense Layer: Units: 1 Optimizer : Adam Loss Function: Mean Squared Error (MSE) random state: 42 Model Training: 100 epochs	GRU Layer: Units: 50 Activation : ReLU Dense Layer: Units: 1 Optimizer : Adam Loss Function: Mean Squared Error (MSE) random state: 42 Model Training: 100 epochs	n_estimators : 200 max depth: 10 min samples split: 5 min samples leaf: 2 max featutes: “sqrt” bootstrap: True random state: 42 Model Training: Train Size: 80% Test Size: 20%	Conv1D Layer: Filters: 64 Kernel size: 3 Activation: ReLU MaxPooling1D Layer: Pool size: 2 Flatten Layer Dense Layer (1): Units: 50 ReLU Activation Dense Layer (2): Units: 1 Optimizer: Adam Loss Fucntion: Mean Squared	n_neighbors: 5 weights: “uniform” algorithm: “auto” leaf size: 30 p: 2 metric: “minkowski” metric params: {'p': 2} n_jobs: 1 random state: 42 Model Training: Train Size: 80% Test Size:	Dense Layer 1: Units: 100 Activation : Sigmoid Dense Layer 2: Units: 50 Activation : ReLU Dense Layer 3: Units: 25 Activation : ReLU Dense Layer (Output): Units: 1 Activation : Linear Optimizer	n_clusters: 200 random state: 42 RBF Layer: Units: 200 Activation : ReLU İnput_dim : n_clusters Dense Layer (output): Units: 1 Activation : Linear Optimizer: Adam Loss Function:	Kernel: Linear Degree: 3 (polynomial) Gamma: “scale” Coef0: 0.0 Tol: 0.001 C: 1.0 Epsilon: 0.1 Shrinking: True Cache size: 200 Verbose: false Max iterations: 1 random state: 42 Model

32 batch size Verbose 1 Train Size: 80% Test Size: 20%	32 batch size Verbose 1 Train Size: 80% Test Size: 20%		Error (MSE) random state: 42 Model Training: 100 epochs 32 batch size Verbose 1 Train Size: 80% Test Size: 20%	20%	: Adam Loss Function: Mean Squared Error (MSE) random state: 42 Model Training: 100 epochs 32 batch size Verbose 1 Train Size: 80% Test Size: 20%	Mean Squared Error (MSE) random state: 42 Model Training: 100 epochs 32 batch size Verbose 1 Train Size: 80% Test Size: 20%	Training: Train Size: 80% Test Size: 20%

3.1. Long and Short Term Memory (LSTM) Model

A modern neural network architecture called Long Short-Term Memory (LSTM) is renowned for excelling at sequential data categorization. An integral feature distinguishing LSTM from traditional neural networks is a memory gate that facilitates the retention of critical information. Additionally, LSTM incorporates a forgetting gate, allowing it to discard irrelevant data (Alshaikhdeeb ve Cheah, 2023: 546).

In order to create a consistent range of values between 0 and 1, the min-max method was used to standardize the data sets, as shown in Equation (6).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6}$$

The LSTM mechanism's forget gate specifically functions to eliminate the cell state data from the preceding sequence. The time series' current input is denoted by x_t, while its previous hidden state is symbolized by h_{t-1}. Both of these values are processed by the activation function σ_g, resulting in the generation of the output vector f_t, which is linked to the forget gate. This relationship can be expressed using Equation (7), wherein the bias coefficient is denoted as b_f, the forget gates are represented as W_f and U_f, and the activation function is symbolized as σ_g.

$$f_t = \sigma_g \left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{7}$$

Equations 8 and 9 use the current data point in the time series input, indicated as x_t, and the hidden state from the previous time step, written as h_{t-1}, in order to determine the coefficients i_t and C'_t within this specific gate. These coefficients are computed through the activation function. The weight coefficients are symbolized by W_i, U_i, W_c, and U_c,while the activation function is represented by the abbreviations σ_g and σ_c.

$$i_t = \sigma_g \left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{8}$$

$$C'_t = \sigma_c \left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{9}$$

The update mechanism in Equation 10 involves the cell state, denoted as C_t. The output of the forget gate, f_t, is multiplied by the previous cell state, C_{t-1}, and added to the output of the input gate, i_t, along with the cell candidate data, C'_t. The changed cell state, C_t, is described by this calculation.

$$C_t = f_t \, x \, C_{t-1} + i_t \, x \, C'_t \tag{10}$$

Equation (11) illustrates the generation of the output vector o_t, which is achieved by applying the activation function $\sigma_g$to the input vectors $h_{t-1}$and x_t. The input gate is associated with bias coefficient b_0 and weight coefficients for the cell state, W_o and U_o.

$$o_t = \sigma_g (W_o x_t + U_o h_{t-1} + b_o) \tag{11}$$

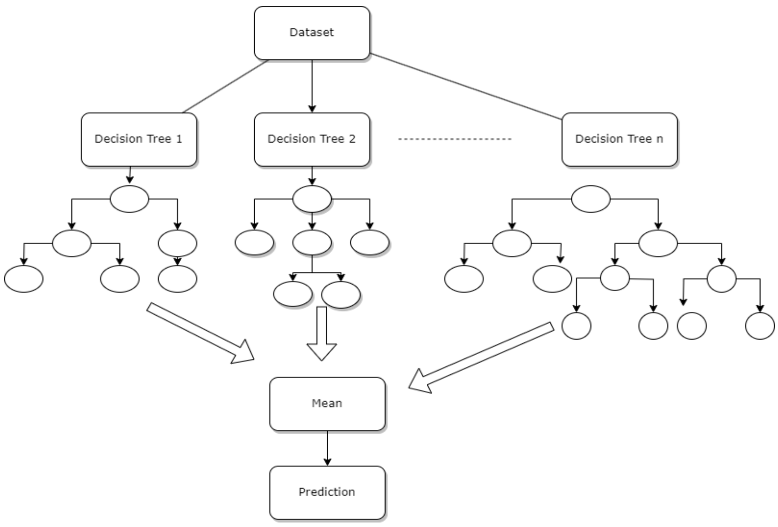
$$h_t = o_t \text{ x } tan\, h\, (C_t) \tag{12}$$

Once generated, the output value o_t is subjected to multiplication with the current sequential cell state C_t. Equation (12) demonstrates the generation of the hidden layer's output through the activation function tanh.

3.2. Random Forest (RF) Model

Classification and Regression Tree (CART), the precursor of Random Forest (RF), was proposed by L. Breman in 1984. In 1996, Breiman developed bagging, a significant RF method. RF is a classification technique based on collective learning. It is constructed using the CART and Bagging protocols. The CART uses a tree-structured classification model to link observations about an item with classification decisions about the object's class. Fig. 1 illustrates CART in terms of the division of a variable as it descends to the exit node, which influences decisions made at each node. For the random forest approach, Figure 1 depicts its organizational structure. The prediction accuracy of CART isn't extremely good, though. Alternatively said, CART has a low bias but a high variance. To solve this issue, RF integrates the Bagging strategy as an extension of CART. This suggests that 1) RF fits many CARTs to bootstrap sets sampled from the first training set, and 2) RF predicts using the mode of the predictions provided by the fitted CARTs. Bagging will reduce the variation of CART while maintaining a low bias. Randomized node optimization is a method that RF employs to further reduce the CART variance. The RF modifications to CART discussed above eliminate its flaws and show incredibly strong performance (Mei vd., 2014).

Figure 1: Random Forest Algorithm Structure



Similar to supervised learning, the Random Forest approach begins with the generation of training data. The remaining test data is processed using this model, which was created from the RF training data. As the name suggests, the learning process entails a machine integrating information and skills, basically assisting the system's knowledge development. The dataset that was presented to the algorithm throughout this process and contained the crucial variables for model formation is sometimes referred to as the "training data." "Test data" refers to data different from this "training data" that were ignored during model training. The effectiveness of the algorithm is evaluated using this test data. The results from each individual tree are meticulously evaluated using the RF approach, which makes use of this particular data (Sevgen and Aliefendioglu, 2020).

3.3. Gated Recurrent Unit (GRU) Model

Cho et al. (2014) created the Recurrent Neural Network (RNN) variant known as the GRU for the first time. Recurrent neural networks (RNNs) have difficulty properly capturing and processing long-range input; the addition of a gating component resolves this issue. In contrast to LSTM, which has a more sophisticated structure, GRU simply has the update gate (z_t) and reset gate (r_t). A Gated Recurrent Unit (GRU)'s update gate, sometimes referred to as the input gate, is crucial in determining how much of the input (x_t) and the previous output (h_{t-1}) should be sent to the cell after it. The reset gate, on the other hand, establishes the extent to which the earlier information should be discounted. As required by the weight W, the current memory content makes it easier to transmit only relevant data to the following iteration (Gao et al., 2021). Equations 13 and 14 establish the main activities of the Gated Recurrent Unit (GRU).

Update Gate:

$$z_t = \sigma(W_z * [h_{t-1}, x_t)] \tag{13}$$

Reset Gate:

$$r_t = \sigma(W_r * [h_{t-1}, x_t)] \tag{14}$$

3.4. Convolutional Neural Networks (CNN) Model

The adaptive moment estimation (Adam) form of the stochastic gradient descent optimization approach is used to train the CNN model, according to Hsieh et al. (2020). By minimizing the loss function, which gauges the discrepancy between network prediction and actual data, the Adam train the neural network's parameter set (Togacar et al., 2019). The cross-entropy loss function, $\mathrm{J(x, y, \theta)}$ which is determined by Equation (15), is the one we have chosen to employ in our CNN. The network learns by decreasing $\mathrm{J(x, y, \theta)}$in respect to the set of network parameters.

$$J(x, y, \theta) = -\sum_{i=1}^{K} x_i log y_i, \tag{15}$$

If K is the minibatch size, then x and y, respectively, represent the ground truth and the CNN's anticipated output. The CNN's parameter set is obtained through backpropagation training using iterative updating and is as follows:

$$\theta_t \leftarrow \theta_{t-1} - \frac{\eta \hat{m}_t}{\sqrt{\theta_t} + \varepsilon} \tag{16}$$

First-moment estimate with bias correction and second-moment computation with bias correction are represented by $\hat{m}_t$ and θ_t, respectively, and are a very tiny constant.

The Adam approach for calculating adaptive learning rates produces unique learning rates for various inputs. The learning rate is set at 0.001 by default. With a high learning rate and a low learning rate, a model will often learn more fast while learning more gradually. The 1D CNN additionally employs the dropout layer between the two convolutional blocks to prevent overfitting. During training, a predetermined percentage of neurons will be randomly selected by the dropout layer, and only their weights will be updated. In other words, only half of the neurons will get updates because the dropout parameter is set at 0.5.

3.5. K-Nearest Neighbors (KNN) Model

The machine learning method known as K-nearest Neighbor is considered to be user-friendly. The stock prediction problem may be mapped using a technique based on similarity. By combining test data with historical stock data, a set of vectors is produced. Each vector in an N-dimensional space corresponds to a stock feature. A similarity metric, such as Euclidean distance, is then established in order to reach a conclusion. This section provides an explanation of kNN. The lazy learning method known as kNN is used to generate the k records in the training data set that, in terms of k values, are closest to the test (i.e., query record). Following that, each of the selected k records votes on the class label for the query record (Alkhatib et al., 2013). The following stock market closing price forecast is produced using kNN: From step a, find the k closest neighbors. b) Determine the distance between the training samples and the query record. c) Place each training data record in distance order. d) Assign the prediction value for the query record to the class labels of the k closest neighbors with the most votes.

3.6. Support Vector Machine (SVM) Model

Vapnik created the H-support vector machine in 1998 to address prediction issues. The -SVM is mostly introduced here. Typical examples of support vector regression are as follows: Given a collection of data points, $\{(x_1, y_1), \dots, (x_i, y_i)\}$, where $x_i \; \varepsilon \; R$ is an input and $y_i \; \varepsilon \; R$ is a target output (Fenghua et al., 2014):

$$\min_{\omega, b, \xi, \xi^*} \frac{\omega^T \omega}{2} + c\sum_{i=1}^{n} \xi_i + c\sum_{i=1}^{n} \xi^*$$

$$s.t{:}\ \omega^T \phi\,(x_i) + b - y_i \leq \varepsilon + \xi_i \tag{17}$$

$$\omega^T \phi\,(x_i) + \mathrm{b} - y_i \leq \ \varepsilon + \xi_i{}^*$$

$$\xi_i, \xi_i{}^* \ \geq 0, i = 1, 2 \dots n$$

The saddle point condition and duality allow us to get the dual form, which is needed to solve the aforementioned problem:

$$\min_{\partial,\partial^*} \frac{1}{2}\,(\partial - \partial^*)^T\, Q(\partial - \partial^*) + \mathcal{E}\sum_{i=1}^{n}(\partial_i + \partial^*{}_i) + \mathcal{E}\sum_{i=1}^{n} y_i(\,\partial_i - \ \partial^*{}_i)$$

$$s.t{:}\ \textstyle\sum_{i=1}^{n}(\partial_i - \ \partial^*{}_i) = 0, 0 \ \leq \partial_i, \partial^*{}_i \leq c, i = 1, 2 \dots n \tag{18}$$

$$\omega = \ \textstyle\sum_{i=1}^{n}(\partial^*{}_i - \partial_i) \tag{19}$$

The kernel functions $Q = k\big(x_i, x_j\big) \equiv \ \phi(x_i)^T \phi\,\big(x_j\big),\ k(x_i, x_j)$ exist. A kernel function can be any function that meets the Mercer condition. The main kernel operations are as follows:

(a) Linear kernel $\mathrm{k}(\mathrm{x}, \mathrm{x_i}) = \mathrm{x^T x_i}$;

(b) Polymonial kernel $\mathrm{k}(\mathrm{x}, \mathrm{x_i}) = (\mathrm{yx^T x_i} + \mathrm{r})^{\mathrm{p}}$;

(c) RBF kernel $\mathrm{k}(\mathrm{x}, \mathrm{x_i}) = \exp(-\mathrm{y}\|\mathrm{x} - \mathrm{x_i}\|^2)$;

(d) Sigmoid kernel $\mathrm{k}(\mathrm{x}, \mathrm{x_i}) = \tanh(\mathrm{yx^T x_i} + \mathrm{r})$.

The best classifier can be found from the analysis above:

$$f(x) = \textstyle\sum_{i=1}^{n}(\partial^*{}_i - \partial_i)\ k(x_i, x) + b \tag{20}$$

3.7. Multilayer Perceptron (MLP) Model

Neural networks have been used for tasks including predicting stock price, modeling option pricing, and estimating currency exchange rates. A learning tool called an artificial neural network (ANN) was developed to mimic how the brain does particular tasks. The MLP is a common artificial neural network (ANN) composed of a number of linked neurons organized in layers. Each neuron's activation function converts a linear mixture of input signals into an output. According to Ince and Trafalis (2008), the weights of this linear combination are coupled to the synaptic weights that connect this neuron to every other neuron in the top layer.

$$y_j = f(\sum_{i=1}^{m} w_{ji} x_i - b) \quad (21)$$

When f is the activation function, w_{ji} is the synaptic weight connected to the connection between the generic neurons found in layers j and i , respectively, and b is the bias term (another neuron weight). The output of the layer j generic neuron is identified as y_j.

The weight update strategy that is most typically used for training MLP networks is known as "error backpropagation" (EBP). The back propagation learning approach's main idea is to repeatedly apply the formula for determining the impact of each network weight on a certain error function. The EBP method is more thoroughly explained in Haykin (1994) (Ince and Trafalis, 2008).

3.8. Radial Basis Function (RBF) Model

RBF kernels, sometimes referred to as radial basis function kernels, are extensively used in a variety of machine learning kernelized learning approaches. It is most typically used in classification using support vector machines. The value of a radial basis function is a real-valued function that exclusively depends on either the distance from the origin or, alternatively, the distance from a different point called the center. Any function that satisfies the requirements is a radial function. The RBF Kernel, which is only a low-band pass filter, is frequently used in signal processing to blur images. Reddy (2018) claims that the RBF Kernel acts as the prior that selects smooth solutions. The Gaussian kernel is a kernel that assumes the shape of a Gaussian function rather than a radial basis function. It is often referred to as the Radial basis function kernel or the RBF kernel. A description of the RBF kernel is provided below:

$$K_{RBF}(x, x') = \exp[-y\|x - x'\|^2] \quad (22)$$

where the kernel's "spread" is set by the parameter y.

A fresh foundation for the output function's synthesis is provided by the RBF units. The radial basis functions are overcomplete and non-orthogonal.

4. Results and Discussion

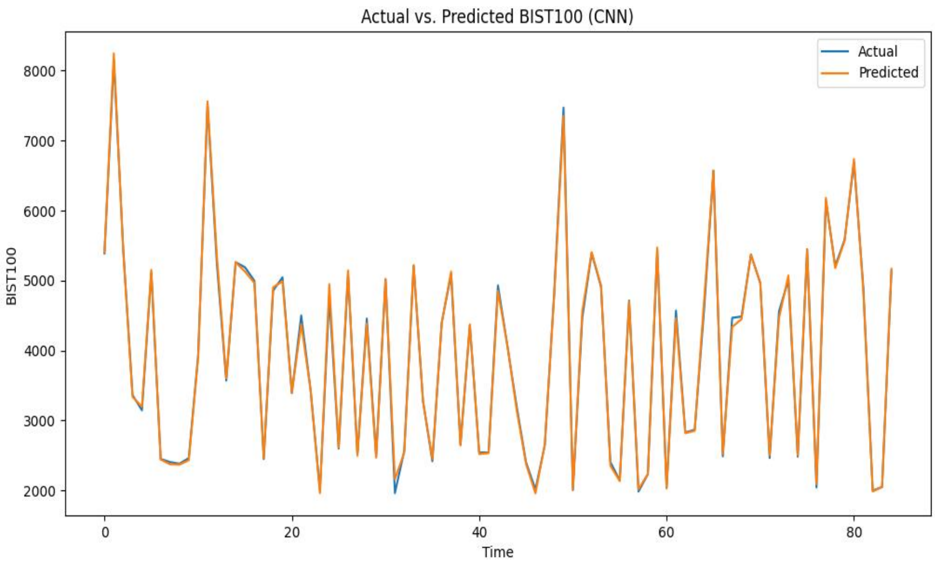
In this study, LSTM, CNN, SVM, RBF, MLP, GRU, RF, and KNN models are used to predict the BIST100 index. Many factors, such as the dollar/Turkish Lira exchange rate, the euro/Turkish Lira exchange rate, the oil price, BIST trading volume, BIST overnight repo rates, BIST trading quantity, and the BIST Industrial Index (XUSIN), are used in the models. Data are collected from the evds.mb.gov.tr website, which serves as the official portal of the Central Bank of the Republic of Turkey. The data set has a time span of 425 days, starting in January 2022 and ending in June 2023. A total of five different error statistics are used in the study. These error coefficients are RMSE, MSE, MAE, MAPE, and R^2. Each model is developed, trained, and then evaluated using a specific data set. Table 3 shows the error coefficients calculated from the model tes results.

Table 3: Test Results of LSTM, CNN, SVM, RBF, MLP, GRU, RF, and KNN Algorithms

	LSTM	CNN	SVM	RBF	MLP	GRU	RF	KNN
RMSE	80.414	55.399	434.908	129.702	82.792	66.598	63.186	124.461
MSE	6466.48	3069.05	189145.54	16822.71	6854.55	4435.38	3992.58	15490.67
MAE	58.629	38.733	390.606	98.080	67.246	53.321	39.986	90.767
MAPE	0.0159	0.0104	0.1282	0.0254	0.0215	0.0152	0.0098	0.0221
R^2	0.9981	0.9986	0.9348	0.9952	0.9975	0.9984	0.9988	0.9941

It is clear from the statistics in Table 3 that the CNN model outperforms the other models. The statistics are used to measure the extent to which the predictions deviate from the true values using the RMSE and MSE measures. Decreasing RMSE and MSE values indicate superior prediction accuracy. As a result, it can be seen that the CNN model performs better in terms of RMSE and MSE. MAE evaluates how close the actual facts and predictions are to each other. A smaller MAE score is indicative of more accurate predictions. In this context, it can be seen that the GRU model exhibits a relatively lower MAE value compared to other models. Mean Absolute Percentage Error (MAPE) measures the level of error in percentage terms. A lower MAPE number is indicative of more accurate predictions. The CNN model shows superior performance compared to other models when MAPE is taken into account. The extent to which the model takes into account the variability of the data is measured by the R^2 statistic. A larger R^2 value indicates a stronger correlation between the model and the observed data. The CNN model exhibits a superior R^2 value compared to alternative models. It only underperforms the LSTM model by a very small margin. As a result, the CNN model shows superior performance in terms of error measures such as RMSE, MSE, MAE as well as explanatory power measured by the coefficient of determination R^2. Figure 2 is a graph showing the actual and predicted values of the CNN model.

Figure 2: Actual and Predicted Values of CNN Algorithm



In this study, the variables of the dollar/Turkish Lira exchange rate, the euro/Turkish Lira exchange rate, the oil price, BIST trading volume, BIST overnight repo rates, BIST trading quantity, and the BIST Industrial Index (XUSIN) were employed to estimate the BIST100 index. The findings indicate that the CNN model outperforms the LSTM, GRU, RF, KNN, MLP, RBF, and SVM algorithms in terms of prediction accuracy.

This study demonstrated that the CNN model significantly outperforms other machine learning and deep learning models in predicting the BIST 100 index. Comparing our results with the existing literature, Fenghua et al. (2014) showed that hybrid models like SSA-SVM perform better than standalone SVM models, aligning with our finding that complex models, such as CNN, offer superior performance. Heo and Yang (2016) and Sethia and Raut (2019) found SVM and LSTM models effective, respectively, yet our study emphasizes CNN's superior predictive capability. Nikou et al. (2019) and Kantar (2020) also highlighted the effectiveness of deep learning methods over traditional models, consistent with our results. Mehtab et al. (2020) and Mukherjee et al. (2023) further supported CNN's robustness in financial forecasting, reinforcing our findings. Li et al. (2023) demonstrated that model enhancements improve accuracy, which parallels our study's emphasis on advanced preprocessing and model sophistication. This collective evidence underscores the reliability of CNN models in financial forecasting. Future research should consider incorporating additional financial variables, extending the time frame, and examining the influence of external factors to further validate and enhance prediction accuracy.

These findings can be validated in other research by reproducing them with more data and over longer time periods. Analyzing performance over time can help determine how stable a model is. Although the variables utilized in this study appear to increase forecast accuracy, the effects of additional financial or economic variables may also be studied. To enhance prediction accuracy, extra data may be good to collect. Additionally, frequent updates of this data might support the model's ability to constantly generate current projections with high accuracy. Future research may concentrate on financial decisions like risk management and portfolio optimization in addition to financial projections. This could aid investors in creating more sensible and successful investment plans. Finally, greater research on how external elements, such as economic or political events, affect the model's outcomes, may aid in producing forecasts that are more reliable and comprehensive.

References

Alaca, M., and Güran, A. (2022). Trend Forecasting of BIST 100 Index Using Sentiment Scores and Technical Indicators During the COVID-19 Pandemic. *Journal of Information Technologies*, 15(4), 379-388.

Alkhatib, K., Najadat, H., Hmeidi, I., and Shatnawi, M,K,A. (2013). Stock price prediction using k-nearest neighbor (kNN) algorithm. International Journal of Business, *Humanities and Technology*, 3(3), 32-44.

Alper, D., and Kara, E. (2017). Macroeconomic Factors Affecting Stock Returns in Borsa Istanbul: A Research on BIST Industrial Index. *Journal of Süleyman Demirel University Faculty of Economics and Administrative Sciences*, 22(3), 713-730.

Alshaikhdeeb, A. J., and Cheah, Y. N. (2023). Utilizing Word Index Approach with LSTM Architecture for Extracting Adverse Drug Reaction from Medical Reviews. *Journal of Advances in Information Technology*, 14(3).

Aydin, A. D., and Cavdar, S. C. (2015). Comparison Of Prediction Performances Of Artificial Neural Network (ANN) And Vector Autoregressive (VAR) Models By Using The Macroeconomic Variables Of Gold Prices, Borsa Istanbul (BIST) 100 Index And US Dollar-Turkish Lira (USD/TRY) Exchange Rates. *Procedia Economics and Finance*, 30, 3-14.

Bulut, E. (2024). Market Volatility and Models for Forecasting Volatility. In *Business Continuity Management and Resilience: Theories, Models, and Processes* (pp. 220-248). IGI Global.

Cho K., Van Merrienboer B., Gulcehre C. et al., (2014). Learning Phrase Representations Using RNN Encoder-Decoder For Statistical Machine Translation, *arXiv preprint arXiv*:1406.1078, 2014.

Fenghua, W. E. N., Jihong, X. I. A. O., Zhifang, H. E., and Xu, G. O. N. G. (2014). Stock price prediction based on SSA and SVM. *Procedia Computer Science*, 31, 625-631.

Gao, Y., Wang, R. and Zhou, E. (2021). Stock Prediction Based on Optimized LSTM and GRU Models. *Scientific Programming*, 2021. https://doi.org/10.1155/2021/4055281

Heo, J., and Yang, J. Y. (2016). Stock Price Prediction Based On Financial Statements Using SVM. *International Journal of Hybrid Information Technology*, 9(2), 57-66.

Hsieh, C. H., Li, Y. S., Hwang, B. J., and Hsiao, C. H. (2020). Detection of atrial fibrillation using 1D convolutional neural network. *Sensors*, 20(7), 2136.

Ince, H., and Trafalis, T. B. (2008). Short Term Forecasting With Support Vector Machines And Application To Stock Price Prediction. *International Journal of General Systems*, 37(6), 677-687.

Kantar, L. (2020). Forecasting The Bıst 100 Index With Artificıial Neural Networks And Arma Model. *Journal of Accounting and Finance Review*, 3(2), 121-131.

Karcıoğlu, R., and Özer, A. (2014). Determination of Factors Affecting Stock Returns in BIST: Static and Dynamic Panel Data Analysis. *Journal of Uludag University Faculty of Economics and Administrative Sciences*, 33(1), 43-70.

Kemalbay, G., and Alkış, B. N. (2020). Prediction of Stock Market Index Movement Direction with Multiple Logistic Regression and K-Nearest Neighbor Algorithm. *Pamukkale University Journal of Engineering Sciences*, 27(4), 556-569.

Koyuncu, T. (2018). The Relationship of BIST-100 Index with Macroeconomic Variables: An Empirical Study. *Journal of Finance, Economics and Social Research*, 3(3), 615-624.

Kurani, A., Doshi, P., Vakharia, A., and Shah, M. (2023). A Comprehensive Comparative Study Of Artificial Neural Network (ANN) And Support Vector Machines (SVM) On Stock Forecasting. Annals of Data Science, 10(1), 183-208.

Li, M., Zhu, Y., Shen, Y., and Angelova, M. (2023). Clustering-Enhanced Stock Price Prediction Using Deep Learning. *World Wide Web*, 26(1), 207-232.

Mehtab, S., and Sen, J. (2020, November). Stock Price Prediction Using CNN And LSTM-Based Deep Learning Models. *In 2020 International Conference on Decision Aid Sciences and Application* (DASA) (pp. 447-453). IEEE.

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

Mei, J., He, D., Harley, R., Habetler, T., and Qu, G. (2014). A Random Forest Method For Real-Time Price Forecasting İn New York Electricity Market. *In 2014 IEEE PES General Meeting| Conference & Exposition* (pp. 1-5). IEEE.

Mukherjee, S., Sadhukhan, B., Sarkar, N., Roy, D., and De, S. (2023). Stock Market Prediction Using Deep Learning Algorithms. *CAAI Transactions on Intelligence Technology*, 8(1), 82-94.

Nikou, M., Mansourfar, G., and Bagherzadeh, J. (2019). Stock Price Prediction Using DEEP Learning Algorithm And İts Comparison With Machine Learning Algorithms. Intelligent Systems In Accounting, *Finance and Management*, 26(4), 164-174.

Reddy, V. K. S. (2018). Stock Market Prediction Using Machine Learning. *International Research Journal of Engineering and Technology (IRJET)*, 5(10), 1033-1035.

Sarıkoç, M., and Çelik, M. (2022). BIST100 Index Price Prediction with Dimension Reduction Techniques and LSTM Deep Learning Network. *European Journal of Science and Technology*, (34), 519-524.

Sethia, A., and Raut, P. (2019). Application of LSTM, GRU and ICA for stock price prediction. *In Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS* 2018, Volume 2 (pp. 479-487). Springer Singapore.

Sevinç, E. (2014). Determination of the Effects of Macroeconomic Variables on Stock Returns Traded in BÌST-30 Index Using Arbitrage Pricing Model. *Istanbul University Journal of the School of Business Administration*, 43(2), 271-292.

Simsek, A. I. (2024a). Using Machine Learning and Deep Learning Methods in Predicting the Islamic Index Price. In *Fintech Applications in Islamic Finance: AI, Machine Learning, and Blockchain Techniques* (pp. 272-285). IGI Global.

Simsek, A. I. (2024b). Improving the Performance of Stock Price Prediction: A Comparative Study of Random Forest, XGBoost, and Stacked Generalization Approaches. In *Revolutionizing the Global Stock Market: Harnessing Blockchain for Enhanced Adaptability* (pp. 83-99). IGI Global.

Togacar, M., Ergen, B., and Sertkaya, M. E. (2019). Subclass separation of white blood cell images using convolutional neural network models. *Elektronika ir Elektrotechnika*, *25*(5), 63-68.

Vijh, M., Chandola, D., Tikkiwal, V. A., and Kumar, A. (2020). Stock Closing Price Prediction Using Machine Learning Techniques. *Procedia Computer Science*, 167, 599-606.

Wenjie, L. U., Jiazheng, L. I., Jingyang, W. A. N. G., and Shaowen, W. U. (2022). A Novel Model For Stock Closıng Prıce Predıctıon Usıng Cnn-Attentıon-Gru-Attentıon. *Economic Computation & Economic Cybernetics Studies & Research*, 56(3).

Yan, W. L. (2023). Stock Index Futures Price Prediction Using Feature Selection and Deep Learning. *The North American Journal of Economics and Finance*, 64, 101867.