


<https://dergipark.org.tr/tr/pub/khosbd>

Predicting Bitcoin Price: Comparative Analysis of Machine Learning and Deep Learning Models

Bitcoin Fiyat Tahmini: Makine Öğrenmesi ve Derin Öğrenme Yöntemlerine İlişkin Karşılaştırmalı Bir Analiz

Ahmed İhsan ŞİMŞEK 1* 

¹Fırat Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, İşletme Bölümü, Elazığ, Türkiye

Makale Bilgisi

Araştırma makalesi
Başvuru: 22.11.2023
Düzeltilme: 31.01.2024
Kabul: 18.04.2024

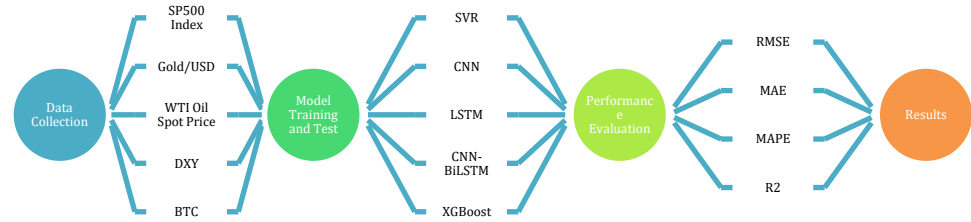
Highlights

SVR, CNN, LSTM, CNN-BiLSTM and XGBoost models were comprehensively compared for Bitcoin price prediction. S&P500, Gold/Dollar, Oil and Dollar Index were used in Bitcoin prediction, increasing the prediction accuracy. XGBoost's superior success compared to other models shows the power of this model in financial predictions.

Keywords

Bitcoin
Time Series
Deep Learning
Machine Learning
Price Prediction

Graphical Abstract



Anahtar Kelimeler

Bitcoin
Zaman Serisi
Derin Öğrenme
Makine Öğrenmesi
Fiyat Tahmini

Abstract

In recent years, Bitcoin has become an important financial instrument that has attracted increasing attention as a digital currency. Accurately predicting the value of a financial asset is of great importance for both individual and institutional investors. The aim of this study is to evaluate and compare the predictive power of different models (Support Vector Regression (SVR), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Hybrid model, which is a combination of CNN and Bidirectional LSTM (CNN-BiLSTM), and XGBoost) in predicting the Bitcoin price. The main objective of the study is to determine the most effective algorithm in predicting the Bitcoin price. In the study, external factors such as S&P500 index, Gold/Dollar exchange rate, West Texas Intermediate Oil Price and Dollar Index were used to predict the Bitcoin price. The dataset covers 2191 days of data between January 1, 2015 and September 18, 2023. The models went through a two-stage process consisting of training and testing stages. The performance of the models is evaluated using various statistical metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-squared (R²). The results show that the XGBoost algorithm gives the best results in all performance metrics. The XGBoost model is followed by CNN-BiLSTM, CNN and LSTM models, respectively. The SVR model exhibited the lowest performance.

Özet

Son yıllarda Bitcoin, dijital bir para birimi olarak giderek daha fazla ilgi gören önemli bir finansal araç haline gelmiştir. Bir finansal varlığın değerini doğru bir şekilde tahmin etmek, hem bireysel hem de kurumsal yatırımcılar için büyük önem taşımaktadır. Bu çalışmanın amacı, Bitcoin fiyatını tahmin etmede farklı modellerin (Destek Vektör Regresyonu (SVR), Konvolüsyonel Sinir Ağı (CNN), Uzun Kısa Süreli Hafıza (LSTM), CNN ve Çift Yönlü LSTM'nin (CNN-BiLSTM) birleşimi olan hibrit model ve XGBoost) tahmin gücünü değerlendirmek ve karşılaştırmaktır. Çalışmanın temel hedefi, Bitcoin fiyatını tahmin etmekte en etkili algoritmayı belirlemektir. Çalışmada Bitcoin fiyatını tahmin etmek için S&P500 endeksi, Altın/Dolar kuru, West Texas Petrol Fiyatı ve Dolar Endeksi gibi dışsal faktörler kullanılmıştır. Veri seti, 1 Ocak 2015 ile 18 Eylül 2023 tarihleri arasındaki 2191 günlük veriyi kapsamaktadır. Modeller, eğitim ve test aşamalarından oluşan iki aşamalı bir süreçten geçmiştir. Modellerin performansı, Kök Ortalama Kare Hatası (RMSE), Ortalama Mutlak Hata (MAE), Ortalama Mutlak Yüzde Hatası (MAPE) ve R-kare (R²) gibi çeşitli istatistiksel ölçütler kullanılarak değerlendirilmiştir. Sonuçlar, XGBoost algoritmasının tüm performans ölçütlerinde en iyi sonuçları verdiğini göstermektedir. XGBoost modelini sırasıyla CNN-BiLSTM, CNN ve LSTM modelleri takip etmiştir. SVR modeli ise en düşük performansı sergilemiştir.

* Corresponding author, e-mail: aismsek@firat.edu.tr

1. INTRODUCTION

The debut of Bitcoin occurred in 2008 under the alias Satoshi Nakamoto. Bitcoin was conceptualized as a novel payment system that facilitates instantaneous electronic currency transactions [1]. When considering Bitcoin, the primary concept that arises is that of blockchain technology. The utilization of blockchain technology ensures that all transactions conducted within the network undergo encryption and subsequent recording. These records of transactions are then saved within blocks. The recent surge in Bitcoin prices has generated heightened interest among investors and regulatory bodies alike. Investors began perceiving Bitcoin as a viable financial vehicle, thereby allocating a portion of their investment

Upon analysis of Figure 1, notable fluctuations are detected in specific time intervals, characterized by significant increases and declines. As an illustration, the value of the asset increased from around \$2,800 in August 2017 to \$18,000 in December 2017, representing a period of approximately four months. Subsequently, the value of the asset saw a decline, reaching a level of 6900 dollars in the month of February in the year 2018. In a comparable vein, the value of the subject in question experienced an increase from approximately \$5000 in early 2020 to reach a peak of \$69000 in the middle of 2021. The significant level of volatility exhibited by this phenomenon garners the interest of investors. Currently, the task of forecasting the price of Bitcoin holds significant importance.

Accurately estimating the valuation of a financial asset holds significant importance for investors. There exist a multitude of techniques employed

portfolio towards Bitcoin. Consequently, the Bitcoin price experienced substantial manipulation, resulting in a notable escalation in fluctuation. Figure 1 depicts the temporal progression of Bitcoin's daily price fluctuations as a time series commencing from the year 2015.

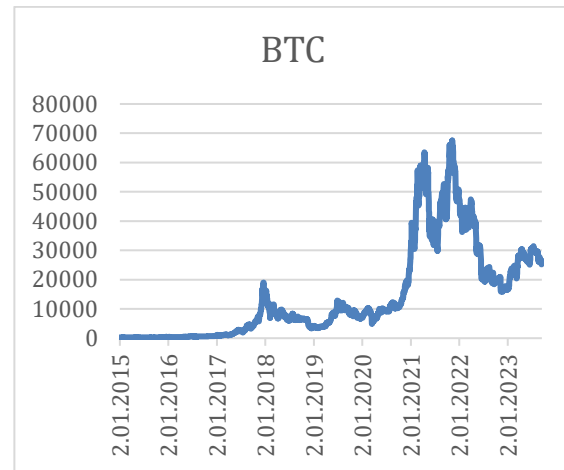


Figure 1: Daily Bitcoin price.

for the purpose of forecasting the value of a commodity. The aforementioned methodologies encompass time series analysis, machine learning, and deep learning techniques. A variety of approaches have been utilized in the existing body of literature to predict the price of Bitcoin. For example, ANN [2, 3], LSTM [4, 5]; SVR [6], CNN [7-9], XGBoost [10, 11] and CNN-BiLSTM Hybrid Model [12-15]. This is the first study that employed gold, stock market, oil and dollar index to predict Bitcoin price. Also in this study 5 different methodologies is compared together for the first time to predict Bitcoin price. In contrast to previous studies, the data set was enlarged and data was collected on 01.02.2015. No study encompassing a more extensive dataset has been located in the existing literature. The use of a substantial data collection facilitated the models to acquire enhanced learning capabilities

and generate superior outcomes in comparison to studies employing smaller data sets.

The objective of this research is to conduct a comparative analysis of various predictive models, namely SVR, CNN, LSTM, XGBoost, and the hybrid model CNN-BiLSTM, in the context of Bitcoin price prediction. Several studies in the literature exclusively utilized Bitcoin data for the purpose of predicting Bitcoin prices [2,12,16]. Furthermore, the literature has research that employ several variables to forecast the price of Bitcoin. The SP500 index price and oil price have a significant impact on Bitcoin price changes [17]. Also the gold price has a role in Bitcoin price [8]. Additionally, some studies emphasize the importance of the Dollar Index (DXY) in relation to Bitcoin price fluctuations [18]. This study incorporated the macroeconomic indicators commonly utilized in the literature to forecast the price of Bitcoin. This study utilized the variables of the S&P500 Index, Gold/USD (ounces), US Dollar Index (DXY), and West Texas Oil Spot Price to make predictions about the price of Bitcoin. The analysis incorporates data from the variables spanning from January 1, 2015, to September 18, 2023. The data pertaining to the variables utilized in the study were collected on a daily basis from the website investing.com. The numpy library was employed in the Python program to assure temporal synchronization of the variables. In order to conduct time series analysis, it is necessary for the data to be temporally synchronized. Specifically, the daily data of the SP500 index, which is one of the variables utilized, distinguishes itself from other variables due to the absence of weekend data in this index. The

synchronization procedure can be performed manually in Excel, or it can be automated within the software by implementing the required coding. To resolve this issue and achieve data parity, the requisite coding was implemented using the Python programming language, resulting in data synchronization. Initially, data was extracted from individual Excel files for each variable. The dates of each variable were transformed into a datetime object using the "pd.to_datetime" function. Additionally, the "Date" column in the variable file was designated as an index using the "set_index" function. Subsequently, the collection of data was consolidated by employing the "pd.concat" function. Subsequently, the data set's time range was established to generate the missing dates, which were then constructed using the "reindex" function, resulting in the data set being indexed based on these dates. Ultimately, all missing values were replaced with the numerical value of "0" through the utilization of the "fillna(0)" method. Once the dates with complete data were identified, a fresh dataset was generated and saved in a new Excel file using the "to_excel" function. Ultimately, the temporal synchronization procedure was finalized by removing the dates that lacked data for any variable in Excel. Following the temporal synchronization of the daily data pertaining to the variables, a total of 2191 days of data spanning from 02.01.2015 to 18.09.2023 were incorporated into the study. Subsequently, the SVR, LSTM, CNN, CNN-BiLSTM, and XGBoost methodologies were implemented in Python for the purpose of this investigation. The respective models were constructed and trained

accordingly. After the conclusion of the training phase, the testing phase was carried out to assess the efficacy of the models. Statistical metrics, including RMSE, MAE, MAPE, and R2 values, were subsequently calculated to assess the models' accuracy. Ultimately, an assessment was conducted on the statistical parameters of the models, leading to the identification of the approach that exhibited the most optimal performance.

The initial phase of the study involves presenting an introduction followed by a brief overview of the relevant literature. Subsequently, the data collection and methodologies employed in the study are presented. During the last section of the study, an assessment is conducted on the performance indicators derived from the models, followed by a comparison of the performances exhibited by these models.

2. LITERATURE REVIEW

The growing curiosity with Bitcoin has led to a surge in research pertaining to the prediction of Bitcoin prices. Based on an analysis of Google Scholar data, it was observed that a mere 11 studies were retrieved while conducting a search using the specified keywords "Bitcoin Price Prediction" within the timeframe of 2011-2015. Nevertheless, a notable rise in the quantity of research studies was seen during subsequent periods, with 403 studies identified between 2016-2020 and a notable surge to 1470 studies between 2021-2023. In the subsequent section, we present a brief overview of various research findings relevant to the prediction of Bitcoin values.

The study conducted by Yiying and Yeze (2019) utilizes advanced artificial intelligence frameworks, specifically ANN and LSTM, to analyze the price patterns of Bitcoin, Ethereum, and Ripple. The findings indicate that s ANN tend to heavily rely on long-term historical information, whereas LSTM networks mostly utilize short-term dynamics [3].

Ferdiansyah et al., conducted a model prediction of the Bitcoin stock market using the LSTM algorithm. The evaluation of the model's performance is determined using the RMSE coefficient. The methodology employed in this research involves the utilization of several techniques and tools to forecast the behavior of Bitcoin within the stock market [4].

Aggarwal et al., conducted a study on the characteristics of gold price that impact the forecast of bitcoin price. The evaluation of this prediction was conducted using the RMSE, applying various deep learning models including CNN, LSTM, and GRU. Based on the obtained findings, it was concluded that the LSTM model generated the most optimal result [7].

In order to address the issue of Bitcoin's highly volatile and unpredictable price fluctuations, offer a novel hybrid neural network model. The proposed model integrates the advantageous features of a CNN with a LSTM models. The findings indicate that the integration of both CNN and LSTM in a hybrid model exhibits a substantial enhancement in the accuracy of both value prediction and direction prediction, in contrast to the use of a singular neural network architecture [14].

Hamayel & Owda, provide an analysis of three distinct recurrent neural network RNN algorithms that are employed for the purpose of price prediction in the context of three specific cryptocurrencies, namely Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH). The algorithms demonstrate exceptional predictive capabilities as indicated by the MAPE. The findings derived from the conducted experiments indicate that the GRU outperformed both the LSTM and BiLSTM models in terms of accuracy in forecasting across all digital currencies types [5].

Cavalli & Amoretti, presents an innovative methodology for forecasting bitcoin trends, utilizing the One-Dimensional (1D CNN) technique. The researchers employed the Gold, Euro, and AAPL Stock Index as variables in their prediction of the price of Bitcoin. Based on the results obtained from implementation of the (1D CNN) model, which was trained, validated, and tested using the aforementioned datasets, it is apparent that this model demonstrates higher predictive power in anticipating bitcoin trends in comparison to LSTM models [8].

Livieris et al., propose a CNN-LSTM model for the purpose of predicting the price and movement of bitcoin. The forecasting model being suggested utilizes various inputs of cryptocurrency data and handles them as discrete entities. This approach allows for the extraction of significant information from each individual cryptocurrency. This study involved a thorough empirical examination of three consecutive years of cryptocurrency data from three cryptocurrencies with the highest market capitalization, including Bitcoin (BTC), Ethereum (ETH), and Ripple

(XRP). The coefficient of determination R^2 for the proposed models is 0.95 [15].

Tiwari et al., suggests employing ARIMA, Facebook Prophet, and XGBoost methodologies for the purpose of forecasting Bitcoin price. The ARIMA model demonstrates a satisfactory level of accuracy in forecasting the price of bitcoin based on historical data up until the day prior [19].

Murugesan et al., employed the Interval Graph (IG) technique to convert the initial dataset into a format suitable for the application of an s (ANN) model. This model was utilized to forecast the price of Bitcoin, and the accuracy of the predictions was assessed using the MAPE, RMSE, and Dstat metrics for assessment. The study has unequivocally illustrated the promising performance and efficacy of the IG-ANN. The evaluation of the performance of the suggested IG-ANN model is conducted by a comparative analysis with conventional ANN approaches using bitcoin time-series data from 2013 to 2019. The results indicate that IG-ANN exhibits higher accuracy relative to all of the other approaches [2].

Erfanian et al., utilizes various comparative methodologies, such as ordinary least squares (OLS), ensemble learning, SVR, and MLP, to examine the predictive capacity of macroeconomic, microeconomic, technical, and blockchain indicators derived from economic theories in relation to the price of BTC. Based on the findings, it has been shown that SVR has a higher level of performance compared to alternative machine learning and conventional models[20].

Dong presents a novel approach called the segmented integrated learning (ensemble-SVR) method, which is founded on the SVR methodology. In this investigation, the RMSE was employed as a performance parameter. The simulation results demonstrate that our system exhibits notable advantages in predicting virtual currency prices when compared to SVR and other widely-used machine learning techniques [6].

The study conducted by Hasan et al., presents an enhanced methodology that utilizes a deep learning algorithm, specifically a CNN, to forecast cryptocurrency prices. The present methodology is primarily utilized for the purpose of forecasting the price fluctuations of four prominent cryptocurrencies, namely Litecoin, Monero, Bitcoin, and Ethereum. The results of the analysis suggest that the suggested method has a statistically significant level of precision in forecasting prices, around 98.75% [9].

Sekhar et al., conducted a comparative analysis of the performance of LSTM and XGBoost models in predicting Bitcoin price. The dataset comprises Bitcoin statistics spanning the years 2018 to 2021. The performance indicators of the models under consideration were assessed using MAE and R-squared R^2 . Based on the findings, the LSTM model exhibits a MAE value of 0.1073, whereas the XGBoost model demonstrates an MAE value of 0.023. The R^2 values were calculated to be 0.99 and 0.868, respectively [16].

Kazeminia et al., utilized historical data of Bitcoin to generate forecasts for the closing price of the following day. This was accomplished by employing a novel hybrid 2D-CNNLSTM model, which was further enhanced with the application of OPTUNA hyperparameter adjustments. The

dataset utilized for training the model was acquired by an automated web scraping methodology. The model that was suggested resulted in an R^2 error of 0.98166 and a MAPE of 0.034. The model we propose undergoes evaluation in comparison to three unique models, specifically CNN, LSTM, and GRU. The findings of the study indicate that the hybrid model developed in this research demonstrates effectiveness in properly forecasting bitcoin prices and reliability in assisting investors in making informed investment choices. Furthermore, the aforementioned model has demonstrated superior performance compared to other widely utilized algorithms, specifically CNN, LSTM, and GRU, in terms of R^2 and MAPE metrics [13].

Chen presented a mathematical framework that demonstrates a notable level of predictive accuracy in forecasting the price of Bitcoin for the following day. This model was developed using a combination of random forest regression and LSTM approaches. Additionally, the study aimed to elucidate the elements that exert effect on the price of Bitcoin. The primary focus of the research methodologies centers on the utilization of the ARMA model for time series analysis and the LSTM algorithm within the field of deep learning. While the Diebold-Mariano test does not provide conclusive evidence that random forest regression outperforms LSTM in terms of prediction accuracy, it is worth noting that random forest regression exhibits lower prediction errors, specifically in terms of RMSE and MAPE, compared to LSTM [17].

In their study, Sathiyapriya et al., undertake a comparison examination of the XGBoost and

LSTM models to forecast the future value of the digital currency Ether. This analysis is based on the use of processed and spam-filtered Twitter data. XGBoost has demonstrated considerable efficacy in addressing regression challenges associated with samples that have restricted size, such as those pertaining to weather and demand prediction. On the contrary, the LSTM method is widely acknowledged as a highly effective and straightforward deep learning technique for addressing predicting challenges. This study aims to analyze and contrast the base version of the LSTM model with the XGBoost model in terms of their effectiveness for forecasting Ether values [21].

3. DATA & METHODOLOGY

This study employed a dataset that encompassed the time period from January 1, 2015, to September 18, 2023. The dataset consisted of 2191 days of data for the S&P 500 index, Gold/USD, US Dollar Index (DXY), West Texas Oil Spot Price (WTI), and Bitcoin Spot Price (BTC/USD). The sources from which the data were obtained are shown in the Table 1. The present study employed a range of variables, such the S&P 500 index, Gold/USD, US Dollar Index (DXY), and West Texas Oil Spot Price (WTI), to make a forecast of the value of the Bitcoin.

Table 1: Sources Of The Variables.

Variables	Source
BTC/USD	investing.com
WTI	investing.com
DXY	investing.com
Gold/USD	investing.com
SP500 Index	investing.com

The research encompassed six various approaches. The algorithms under consideration

include LSTM, XGBoost, SVR, CNN, and CNN-BiLSTM. The evaluation of the effectiveness of the model encompassed the assessment of several statistical indicators, including the RMSE, MAE, MAPE, and R2. The equations representing the mathematical expressions for computing the aforementioned statistical parameters are denoted as Equations 1, 2, 3, and 4.

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

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \tag{2}$$

$$MAPE = \frac{\sum_{t=1}^n \frac{u_t}{y_t}}{n} * 100 \tag{3}$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \mu)^2} \tag{4}$$

Equation 1 denotes the square root of the average of the squared differences between the observed values (y_i) and the estimated values (\hat{y}_i) for a set of N data points. The error values are squared, summed, averaged, and then the square root of this value is determined. Equation 2 represents the mean of the absolute discrepancies between the actual values (y_i) and the anticipated values (x_i) for a set of n data points. The absolute value of each error is computed, then all values are added and divided by the number of samples to obtain the average. Equation 3 determines the mean of the absolute percent errors between the actual values (Y_t) and the predicted values (u_t) for a given number of sample data points (n). Each error is quantified as a percentage, transformed into its absolute value, aggregated, and then computed by dividing it by the total number of samples and multiplying the result by 100. Equation 4 quantifies the ratio of the

variance between the actual values (y_i) and the anticipated values (\hat{y}_i) to the variance of the original value. A value close to 1 shows the degree to which the model accurately describes the data. A higher value approaching 1 indicates superior performance of the model.

3.1. Long-Short Term Memory (LSTM) Model

The introduction of LSTM models in 1997 by Hochreiter and Schmidhuber made a notable contribution to the field of neural network models, signifying a substantial progress as they were shown to enhance accuracy when compared to conventional approaches. The LSTM is a recurrent neural network RNN architecture characterized by the presence of gates that regulate the transmission of information inside its cells. The input and forget gate structures have the ability to alter the information that is transmitted along the cell state. As a result, the final result is a refined representation of the cell state, which is influenced by the contextual data provided by the inputs [22]. The stages of the LSTM approach are mentioned in Equation 5-11 below;

$$x_{\text{scaled}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (5)$$

$$f_t = \sigma_g (W_f x_t + U_f h_{t-1} + b_f) \quad (6)$$

$$i_t = \sigma_g (W_i x_t + U_i h_{t-1} + b_i) \quad (7)$$

$$C'_t = \sigma_c (W_c x_t + U_c h_{t-1} + b_c) \quad (8)$$

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \quad (9)$$

$$o_t = \sigma_g (W_o x_t + U_o h_{t-1} + b_o) \quad (10)$$

$$h_t = o_t \times \tanh (C_t) \quad (11)$$

Data has been normalized with Equation 5. Equation 6 incorporates the variables x_t , h_{t-1} ,

f_t , and σ_g , which respectively denote the input of the time series, the prior hidden state, output vector, and the activation function respectively. Additionally, the bias coefficient is denoted as b_f , while the forget gates are represented by W_f and U_f . The forget gate is associated with the output vector. This relationship is expressed by Equation 6. Equations 7 and 8 describe the relationship between the current point in the time series input, denoted as x_t , and the hidden state, denoted as h_{t-1} , from the previous time frame. These variables are responsible for determining the values of the coefficients i_t and C'_t within this gate. The calculation of these coefficients involves the utilization of the activation function. The weight coefficients are represented by variables such as W_i , U_i , W_c , and U_c , whereas the activation function is represented by the symbols σ_g and σ_c . In Equation 9, the cell state, represented as C_t , undergoes an update process in which it is obtained by combining the product of the input gate's output, i_t , and the cell candidate data, C'_t , with the result of multiplied the prior cell state, C_{t-1} , by the outcome of the forget gate, f_t . The calculation provides a characterization of the transformed cellular state, denoted as C_{t-1} . The equation denoted as (10) demonstrates the process by which the output vector σ_t is generated by the application of the activation function σ_g to the input vectors h_{t-1} and x_t . The bias coefficient, denoted as b_o , together with the weighted values of the cell state, represented as W_o and U_o , are linked to the input gate. The output gate's resultant value, denoted as o_t , is multiplied by the current sequential cell state, represented as C_t , subsequent to its generation. The results of the hidden layer is generated by

applying the activation function tanh to the outcome, as depicted in Equation (11). The LSTM model is configured with a unit size of 50. The complexity and learning capacity of the model are determined by the number of neurons selected. Although a decrease in the number of neurons may negatively impact the performance of the model, it does result in faster solutions. Increased neuron quantities contribute to a higher likelihood of overfitting. An activation function is a mathematical function that is applied to the output of a neural network node or neuron. It helps to introduce non-linearity into the network, allowing it to learn and model complex relationships Relu has been chosen. This is due to its simplicity and efficacy, which make it a very versatile choice that excels in various scenarios. The Relu function is commonly favored due to its ability to expedite the learning process, its computational efficiency, and its resilience against vanishing gradient issues. The optimizer was chosen with Adam as the parameter. Adam is an efficient optimization algorithm that typically exhibits strong performance in neural network models. The choice of the loss function was the Mean Squared Error (MSE). The Mean Squared Error (MSE) is often used as a loss function in regression issues. The epoch was selected as 100. Higher epoch values contribute to improved generalization in the model, however too high epoch values might lead to overfitting issues. The batch size was selected as 32. A batch refers to the amount of data that the model will analyze in a single iteration. A number of 32 generates acceptable results for numerous applications.

3.2. Convolutional Neural Network (CNN) Model

The CNN model was introduced by LeCun, gaining inspiration from the structure and functioning of the human brain [28]. The architecture of CNN layers primarily comprises two fundamental elements. One of the components utilized in the process of feature extraction is a type of layer referred to as convolution layers. The subsequent layer, known as the pooling layer, is responsible for executing tasks related to regression and classification [23]. The Equations 12-14 depict the sequential phases of the CNN model.

$$x_j^d = \phi \left(\sum_{i \in M_j} x_i^{d-1} * w_{ij}^d + b_i^d \right) \quad (12)$$

$$x_j^d = \phi(\beta_j^d \text{down}(x_j^{d-1}) + b_i^d) \quad (13)$$

$$y^k = \phi(w^k x^{k-1} + b^k) \quad (14)$$

Equation 12 defines the variable x_j^d as the jth feature map of the convolution layer. The feature map is a key component in the context of this discussion. The activation function, denoted as ϕ , is applied to the input M_j , where d is the dimension of the input. The input feature set refers to the set of features that are provided as input to the layer. The variable w_{ij}^d represents the attribute map of the jth attribute in the dth convolutional layer. The characteristic of the (d-1). Attribute j of the convolution layer and attribute i of layer (d-1) are being referred to in Layer i. The variable b_i^d represents the bias term associated with the respective layer. The subsampling function, denoted as $\text{down}(\cdot)$, and the weight matrix β are represented in Equation 13. In Equation 14, subsequent to the application of the convolutional and pooling layers, the process of classification is executed utilizing the complete link layer. Equation 14 defines the

variables used in the context of a layered neural network. The variable "k" represents the layer index, " y^k " denotes the output of the full link layer, " x^{k-1} " represents the input of the full link layer, " w^k " signifies the weighting coefficient, and " b^k " represents the deviation. The CNN model use a filter size of 64. Increasing the number of filters enhances the model's learning capabilities, but it might lead to overfitting issues and lengthen the model's solution process. The kernel size is selected as 3. Minimal values are particularly employed in the context of time series analysis. Greater dimensions can lead to issues of overfitting. The pool size has been chosen as 2. The maxpooling procedure reduces the size of feature maps and enhances the visibility of higher-level features. Maintaining a small value for this parameter guarantees quicker reduction and little loss of information. Additionally, it offers reduced memory use.

3.3. Extreme Gradient Boosting (XGBoost) Model

XGBoost can be characterized as a scalable and comprehensive tree boosting framework. The XGBoost algorithm utilizes the aggregation of weak classifiers in order to construct a robust and high-performing model [24]. Despite the fact that this particular method requires a longer training time in comparison to traditional methods, it distinguishes itself by generating accurate forecasts [25]. The sequential steps of the XGBoost algorithm are illustrated in equations 15-21.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(y_i), f_k \in \mathcal{F} \quad (15)$$

$$\min L^{(t)}(y_i, \hat{y}_i^{(t)}) = \min \left(\sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^t \Omega(f_k) \right) \quad (16)$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda w^2 \quad (17)$$

$$\min L^{(t)} = \min \left(\sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i) \right] + \Omega(f_t) \right) \quad (18)$$

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, y_i^{t-1}) \quad (19)$$

$$h_i = \partial_{\hat{y}_i^{t-1}}^2 l(y_i, y_i^{t-1}) \quad (20)$$

$$w_j^* = - \frac{\sum g_i}{\sum h_i + \lambda} \quad (21)$$

$$obj^* = - \frac{1}{2} \sum_{j=1}^T \frac{(\sum g_i)^2}{\sum h_i + \lambda} + \gamma \cdot T \quad (22)$$

To forecast the result of a dataset comprising n samples and m features, a tree ensemble model employs K additive functions denoted as $D = \{(x_i, y_i)\} (|D| = n, x_i \in \mathcal{R}^m, y_i \in \mathcal{R})$. Equation 15 denotes the space of regression trees, where \mathcal{F} represents this space. The variable f_k represents the quantity of weak learners, whereas K means the overall count of weak learners. The objective function of the algorithm at time t, abbreviated as $L^{(t)}$, is formally defined by Equation 16. The parameter $l(y_i, \hat{y}_i^{(t)})$ encompasses a variety of loss functions that are utilized to tackle specific issues. Equation 17 presents a regularly employed method for quantifying the level of disparity between the actual value (y_i) and the anticipated value ($\hat{y}_i^{(t)}$), as well as the overall complexity of the model, which is measured by $\sum_{k=1}^t \Omega(f_k)$. The evaluation of the objective function involves the replacement of the

predicted value ($\hat{y}_i^{(t)}$) for the i th sample in the t th iteration. The calculation is executed with the second-order approximation of the Taylor expansion at the previous iteration's estimated value of y , denoted as ($\hat{y}_i^{(t-1)}$), as presented in Equation 18. In Equation 18, the variables g_i and h_i denote the first and second derivatives of the loss function $l(y_i, \hat{y}_i^{(t)})$, respectively. By putting the formulas denoted as Equation 18, Equation 19, and Equation 20 into the aforementioned Equation 16, we may thereafter proceed to compute the derivative. Solutions may be obtained by utilizing Equations 21 and 22. Equations 21 and 22 represent the variable obj^* , which signifies the numerical value of the score of the loss function. A decreased score signifies a tree structure that is closer to ideal. The symbol w_j^* represents the optimal solution for the weights in the specific situation under consideration. The XGBoost model is configured with a value of 100 for the `n_estimators` parameter. This value is typically associated with favorable outcomes. Increasing the number of `n_estimators` results in a greater number of trees, but, this also leads to longer training time. Additionally, selecting bigger numbers may give rise to overfitting issues. The `learning_rate` value was chosen to be 0.3. During the research, an initial value is chosen and then iteratively adjusted to enhance the performance of the model. A smaller learning rate corresponds to a higher number of trees and iterations. Higher values facilitate accelerated learning, but might potentially lead to overfitting issues. Lower `max_depth` values result in simpler and more generalizing trees, but might potentially reduce the flexibility of the model. Higher values of

`max_depth` result in more intricate and detailed trees, but they also elevate the likelihood of overfitting. The study utilized a `max_depth` value of 6. This value is typically an initial reference and has consistently demonstrated strong performance in numerous applications. The subsample value in the study was set to 1. Using small subsample values enables training each tree with a less amount of data, hence enhancing the model's generalizability. Higher subsample values enable each tree to be trained with a larger amount of data, leading to enhanced learning capabilities. However, this can also increase the risk of overfitting. The default value of 1 is commonly chosen and is typically regarded as a suitable starting point for utilizing the complete dataset.

3.4. Support Vector Regression (SVR) Model

The initial introduction of the support vector machine SVR approach can be attributed to Cortes and Vapnik in the 1990s [29]. Following this, a regression methodology called support vector machine for regression was devised [30]. SVR technique was initially devised as a classifier. SVR algorithm is utilized to identify the optimal hyperplane that effectively separates distinct variables. The optimal hyperplane is characterized by having the maximum margin, which ensures an equal separation from all variables [26]. The gradual phases of the SVR technique are outlined in Equation 23-29.

$$f(x) = \omega \Phi(x) + b \tag{23}$$

$$L(f(x), y, \epsilon) = f(x) = \begin{cases} 0 & |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & |y - f(x)| > \epsilon \end{cases} \tag{24}$$

$$\begin{cases} \text{Min. } \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n \xi_i \\ \text{sub. t. } \begin{cases} y_i - \omega\Phi(x_i) - b \leq \varepsilon + \xi_i \\ -y_i + \omega\Phi(x_i) + b \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{cases} \quad (25)$$

$$\omega^* = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \Phi(x_i) \quad (26)$$

$$b^* = \frac{1}{N_{nsv}} \left\{ \sum_{0 < \alpha_i < C} [y_i \sum_{x_i \in SV} (\alpha_i - \alpha_i^*) K(x_i, x_j) - \varepsilon] + \sum_{0 < \alpha_i < C} [y_i - \sum_{x_j \in SV} (\alpha_j - \alpha_j^*) K(x_i, x_j) + \varepsilon] \right\} \quad (27)$$

$$K(x_i, x_j) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \quad (28)$$

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x) + b^* \quad (29)$$

The primary aim of SVR is to identify a linear regression function, represented as $f(x)$, within a space that possesses a high number of dimensions. Assume that x represents an element belonging to the set of real numbers, serving as the sample vector. The function Φ has non-linear characteristics in its mapping. The incorporation of a linear insensitivity loss function, represented as $L(f(x), y, \varepsilon)$, contributes to enhancing the robustness of the optimization problem. The numerical illustration of this loss function is given by Equation 25. In equation 25, the variables x_i and y_i represent the input vector and output value, respectively. These variables are associated with a specific serial number, denoted by i . Both x_i and y_i belong to the set of real numbers, denoted as R . The dimension of the input vector is d . In this context, the variable d denotes the cardinality of the elements included within an input vector. The variable n denotes the number of training samples. The symbol ε denotes the quantification of regression precision. The variable C represents a punishment factor that quantifies the extent of penalty applied to a data sample in the event that its mistake surpasses the threshold value ε . The variables ξ_i and ξ_i^* are

utilized as slack variables to enforce penalties on the complexity of the fitting parameters. In order to identify the estimation of ω and b , it is important to address the optimization problem as stated in Equation 26 and 27. The N_{nsv} denotes the number of support vectors that have been identified. The Lagrange multipliers, represented as α_i and α_i^* , are constrained to be non-negative. Equation 28 employs the kernel function, represented as $K(x_i, x_j)$, within this particular situation. The Gaussian kernel function, known for its exceptional generalization capability, is selected. The equation that represents the final regression function is denoted as Equation 29. The SVR model was created using a linear kernel function. When the "Kernel" is utilized, the SVM or SVR model generates a linear regression plane. Furthermore, the `random_state` parameter has been chosen. The `random_state` parameter determines the initial state that an algorithm uses to produce random numbers. By utilizing the identical `random_state` variable, the generation of random numbers remains consistent, hence guaranteeing the replication of results upon subsequent executions of your code. Ensuring the repeatability of the model is crucial in order to enhance the reproducibility of the results by others.

3.5. CNN-BiLSTM Hybrid Model

The hybrid architecture known as the CNN-BiLSTM model integrates the CNN and LSTM models. The BiLSTM model is an improvement upon the BiLSTM model as it integrates a $1 - \tanh$ function into the output gate. The adjustment leads to the output gate possessing a value range of around (0.24, 1). Therefore, it is evident that BiLSTM exhibits enhanced

prediction ability during the course of training, while also possessing comparable robust capability for learning to BiLSTM. Accordingly BiSLSTM is an appropriate method for examining temporal data associations [27]. Proposed CNN-BiLSTM model is shown in Figure 2.

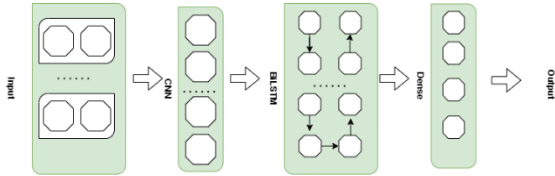


Figure 2: CNN-BiLSTM model.

Each of the stages of the CNN-BiLSTM model are presented in equations 30-35.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (30)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (31)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (32)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{c}_t \quad (33)$$

$$o_t = 1 - \tanh(\sigma(W_o \cdot [h_{t-1}, x_t] + b_o)) \quad (34)$$

$$h_t = o_t * \tanh(C_t) \quad (35)$$

Equation 30 use i_t as the input gate to assess the appropriateness of retaining the current input data via a mathematical approach. The variable \tilde{c}_t is employed in the calculation of data that requires updating using Equation 31. The symbol f_t is frequently utilized in scholarly literature to denote the forget gate. Equation 32 demonstrates the utilization of the sigmoid function in determining the retention of past memories for the present memory state. The variable i_t is utilized as a mechanism for determining the necessity of an update. Furthermore, the variable \tilde{c}_t is employed to calculate if the current state necessitates an update, as determined by Equation

33. After the acquisition of the latest state, the value of the output gate o_t is calculated using equation 34. In contrast to BiSLSTM, the incorporation of the $1 - \tanh(x)$ function is observed at this particular level. The updated memory cell demonstrates the ability to calculate the present concealed state by employing Equation 35.

4. RESULTS AND DISCUSSION

Table 2 presents the results obtained from the implemented LSTM, XGBoost, SVR, CNN, and CNN-BiLSTM models. This investigation employed four statistical metrics. The evaluation tool employed to quantify the discrepancy between projected values and actual values is commonly known as RMSE. A decrease in score indicates a stronger alignment between the model's predictions and actual outcomes. MAPE is a metric used to measure the average size of errors in a forecast, which is presented as a percentage of the actual values. A lower MAPE value is evidence of higher accuracy of the model. The MAE is a statistical measure used to assess the average magnitude of the discrepancies between the expected and observed values. MAE suggests that the model's predictions exhibit a higher level of precision, hence mitigating the impact of significant outliers. The R2 statistic serves as a measure of the proportion of variance in the dependent variable that can be accounted for by the model. When the value of the coefficient of determination R2 approaches 1, it signifies that a significant percentage of the variance in the data can be explained by the model.

Table 2: Statistical performance metrics of the models.

Models/Coeff	RMSE	MAE	MAPE	R ²
LSTM	4.238	2.623	0.520	0.939
CNN-BiLSTM	3.612	2.381	0.820	0.956
XGBoost	1.820	0.956	0.098	0.989
SVR	6.689	5.164	3.555	0.848
CNN	3.834	2.247	0.610	0.950

In comparison to the other models, the SVR model has a lower R² value of 0.848, suggesting a relatively diminished capacity to effectively elucidate the underlying data. The RMSE score of 5.164 is rather large, suggesting that the forecasts exhibit a significant degree of error. The MAE value of 3.555 and MAPE value of 3.555 for this particular model are much greater compared to the other models, suggesting a poor performance of the model.

The LSTM model demonstrates a high R² value of 0.939, suggesting a strong level of explanatory power in relation to the data. Nevertheless, it is worth noting that the RMSE value of 2.623 for this particular model surpasses that of the other models, indicating a greater degree of error in the predictions. The MAE value of 0.520 and MAPE value of 0.520 are deemed acceptable; nonetheless, the model's performance was comparatively inferior to that of the other models.

The R² value (0.950) for the CNN model demonstrates a significant level of accuracy, suggesting that the model effectively elucidates the underlying patterns within the data. The RMSE score of 2.247 demonstrates strong performance in comparison to alternative models. The MAE value of 0.610 and the MAPE value of 0.610 are considered to be within an acceptable range. The performance of CNN was

commendable, however slightly lower compared to that of XGBoost and CNN-BiLSTM.

Based on the findings derived from the CNN-BiLSTM model, it is observed that the R² value exhibits a significantly elevated level (0.956), so signifying a strong capacity of the model to effectively elucidate the underlying data. The RMSE value of 2,381.089 indicates a satisfactory level of performance when compared to alternative models. The MAE value of 0.820 and MAPE value of 0.820 for this model are comparatively higher than those of the other models, suggesting that the predictions generated by this model exhibit a greater degree of error. The CNN-BiLSTM model demonstrates the second highest level of performance.

Upon analysis of the findings presented in Table 2, it is evident that the XGBoost model exhibits a significantly high R² value of 0.989. This substantial number serves as an indication that the model effectively elucidates the underlying patterns within the data. The RMSE value of 0.956 demonstrates a notable decrease in comparison to the other models, suggesting a higher level of accuracy in the predictions, as they closely align with the actual data. The MAE value is seen to be low, specifically 0.098, which suggests that the model's predictions exhibit a generally close proximity to the actual values. The MAPE score exhibits a notable diminution at 0.098, suggesting that the model's prognostications often possess an error margin of under 10%. Based on the obtained findings, it is evident that XGBoost emerges as the model with the highest performance. Upon reviewing the existing literature, it is evident that certain research have demonstrated superior outcomes

when employing hybrid models [13-15]. The LSTM model generated inferior outcomes compared to the CNN model in our investigation. Nevertheless, akin to the aforementioned experiments, the CNN-BiLSTM hybrid model outperformed both the LSTM and CNN models. In their investigation, Sekhar et al. (2022) found that the XGBoost model outperformed the LSTM model [16]. Our investigation demonstrated that the XGBoost model generated superior outcomes. In contrast to the findings of Erfanian et al. (2022), our analysis revealed that the SVR model performed poorly, generating the least favorable outcomes among the algorithms examined [20]. In their study, Cavalli&Amoretti (2021) asserted that the Convolutional Neural Network (CNN) model outperforms the Long Short-Term Memory (LSTM) model in terms of outcomes [8]. Our study generated comparable findings. In contrast to these findings, Aggarwal et al. (2019) discovered that the LSTM model generated superior outcomes compared to the CNN model. Consequently, the XGBoost model emerges as the most effective model in terms of performance on this particular dataset. When the coefficient of determination R^2 is high and the RMSE and MAE are low, the predictions tend to exhibit a high level of accuracy and closely approximate the actual values. The CNN-BiLSTM and CNN models exhibit strong performance, ranking second and third, respectively. However, they fall short in comparison to XGBoost. The performance of the LSTM and SVR models is comparatively inferior to that of the other models.

TEŞEKKÜR (ACKNOWLEDGMENTS)

Bu araştırma hiçbir dış finansman almamıştır.

YAZAR KATKILARI

Ahmed İhsan ŞİMŞEK: Kavramsal tasarım; Araştırma; Metodoloji; Veri düzenleme; Analiz; Yazma; Görselleştirme; Gözden geçirme ve Düzenleme

ÇIKAR ÇATIŞMALARI

Yazar, herhangi bir çıkar çatışması olmadığını beyan eder.

REFERENCES

- [1] S. Nakamoto, "Bridging The Global Digital Divide Through Digital Inclusion: The Role Of ICT Access And ICT Use," Transform. Gov. People, Process Policy, pp. 1–9, 2008.
- [2] R. Murugesan, V. Shanmugaraja, and A. Vadivel, "Forecasting Bitcoin Price Using Interval Graph and ANN Model: A Novel Approach," SN Comput. Sci., vol. 3, no. 5, pp. 1–10, 2022, doi: 10.1007/s42979-022-01291-x.
- [3] W. Yiyang and Z. Yeze, "Cryptocurrency Price Analysis with Artificial Intelligence," 5th Int. Conf. Inf. Manag. ICIM 2019, pp. 97–101, 2019, doi: 10.1109/INFOMAN.2019.8714700.
- [4] Ferdiansyah, S. H. Othman, R. Zahilah Raja Md Radzi, D. Stiawan, Y. Sazaki, and U. Ependi, "A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market," ICECOS 2019 - 3rd Int. Conf. Electr. Eng. Comput. Sci. Proceeding, pp. 206–210, 2019, doi: 10.1109/ICECOS47637.2019.8984499.
- [5] M. J. Hamayel and A. Y. Owda, "A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms," Ai, vol. 2, no. 4, pp. 477–496, 2021, doi: 10.3390/ai2040030.
- [6] S. Dong, "Virtual Currency Price Prediction Based on Segmented Integrated Learning," 2022 IEEE 2nd Int. Conf. Power, Electron. Comput. Appl., no. January, pp. 549–552, 2022, doi: 10.1109/icpeca53709.2022.9719070.
- [7] A. Aggarwal, I. Gupta, N. Garg, and A. Goel, "Deep Learning Approach to Determine the Impact of Socio Economic Factors on Bitcoin Price Prediction," 2019 12th Int. Conf. Contemp. Comput. IC3 2019, pp. 1–5, 2019, doi: 10.1109/IC3.2019.8844928.
- [8] S. Cavalli and M. Amoretti, "CNN-Based Multivariate Data Analysis for Bitcoin Trend Prediction," Appl. Soft Comput., vol. 101, p. 107065, 2021, doi: 10.1016/j.asoc.2020.107065.

- [9] S. H. Hasan, S. H. Hasan, M. S. Ahmed, and S. H. Hasan, "A Novel Cryptocurrency Prediction Method Using Optimum CNN," *Comput. Mater. Contin.*, vol. 71, no. 1, pp. 1051–1063, 2022, doi: 10.32604/cmc.2022.020823.
- [10] E. Edgari, J. Thiojaya, and N. N. Qomariyah, "The Impact of Twitter Sentiment Analysis on Bitcoin Price during COVID-19 with XGBoost," 5th Int. Conf. Comput. Informatics, ICCI 2022, pp. 337–342, 2022, doi: 10.1109/ICCI54321.2022.9756123.
- [11] R. G. Tiwari, A. K. Agarwal, R. K. Kaushal, and N. Kumar, "Prophetic Analysis of Bitcoin price using Machine Learning Approaches," *Proc. IEEE Int. Conf. Signal Process. Control*, vol. 2021-October, pp. 428–432, 2021, doi: 10.1109/ISPCC53510.2021.9609419.
- [12] Q. Guo, S. Lei, Q. Ye, and Z. Fang, "MRC-LSTM: A Hybrid Approach of Multi-scale Residual CNN and LSTM to Predict Bitcoin Price," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2021-July, pp. 1–8, 2021, doi: 10.1109/IJCNN52387.2021.9534453.
- [13] S. Kazemini, H. Sajedi, and M. Arjmand, "Real-Time Bitcoin Price Prediction Using Hybrid 2D-CNN LSTM Model," 2023 9th Int. Conf. Web Res. ICWR 2023, pp. 173–178, 2023, doi: 10.1109/ICWR57742.2023.10139275.
- [14] Y. Li and W. Dai, "Bitcoin Price Forecasting Method Based on CNN-LSTM Hybrid Neural Network Model," *J. Eng.*, vol. 2020, no. 13, pp. 344–347, 2020, doi: 10.1049/joe.2019.1203.
- [15] I. E. Livieris, N. Kiriakidou, S. Stavroyiannis, and P. Pintelas, "An Advanced CNN-LSTM Model for Cryptocurrency Forecasting," *Electron.*, vol. 10, no. 3, pp. 1–16, 2021, doi: 10.3390/electronics10030287.
- [16] P. C. Sekhar, M. Padmaja, B. Sarangi, and Aditya, "Prediction of Cryptocurrency using LSTM and XGBoost," 2022 IEEE Int. Conf. Blockchain Distrib. Syst. Secur. ICBDS 2022, pp. 1–5, 2022, doi: 10.1109/ICBDS53701.2022.9935871.
- [17] J. Chen, "Analysis of Bitcoin Price Prediction Using Machine Learning," *J. Risk Financ. Manag.*, vol. 16, no. 1, 2023, doi: 10.3390/jrfm16010051.
- [18] D. Y. Lee and S. Y. Park, "Global Energy Intensity Convergence Using a Spatial Panel Growth Model," *Appl. Econ.*, vol. 00, no. 00, pp. 1–20, 2022, doi: 10.1080/00036846.2022.2131715.
- [19] R. G. Tiwari, A. K. Agarwal, R. K. Kaushal, and N. Kumar, "Prophetic Analysis of Bitcoin Price Using Machine Learning Approaches," *Proc. IEEE Int. Conf. Signal Process. Control*, vol. 2021-October, pp. 428–432, 2021, doi: 10.1109/ISPCC53510.2021.9609419.
- [20] S. Erfanian, Y. Zhou, A. Razzaq, A. Abbas, A. A. Safeer, and T. Li, "Predicting Bitcoin (BTC) Price in the Context of Economic Theories: A Machine Learning Approach," *Entropy*, vol. 24, no. 10, pp. 1–29, 2022, doi: 10.3390/e24101487.
- [21] K. Sathiyapriya, S. Vankadara, K. S. Babu, and M. Muralidharan, "Performance Comparison of LSTM and XGBOOST for Ether Price Prediction from Spam Filtered Tweets," *Proc. 2023 Int. Conf. Intell. Syst. Commun. IoT Secur. ICISCOIS 2023*, pp. 650–655, 2023, doi: 10.1109/ICISCOIS56541.2023.10100425.
- [22] F. Sherratt, A. Plummer, and P. Irvani, "Understanding Lstm Network Behaviour Of İmu-Based Locomotion Mode Recognition For Applications İn Prostheses And Wearables," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–23, 2021, doi: 10.3390/s21041264.
- [23] S. Mehtab, J. Sen, and S. Dasgupta, "Robust Analysis of Stock Price Time Series Using CNN and LSTM-Based Deep Learning Models," *Proc. 4th Int. Conf. Electron. Commun. Aerosp. Technol. ICECA 2020*, pp. 1481–1486, 2020, doi: 10.1109/ICECA49313.2020.9297652.
- [24] A. Boru İpek, "Prediction of market-clearing price using neural networks based methods and boosting algorithms," *Int. Adv. Res. Eng. J.*, vol. 5, no. 2, pp. 240–246, 2021, doi: 10.35860/iarej.824168.
- [25] H. Abar, "Altın Fiyatlarının Kestirimi," vol. 83, no. Yaz, 2020.
- [26] Y. Zouzou And H. Çıtakoğlu, "Reference Evapotranspiration Prediction From Limited Climatic Variables Using Support Vector Machines and Gaussian Processes," *Eur. J. Sci. Technol.*, no. 28, pp. 346–351, 2021, doi: 10.31590/ejosat.999319.
- [27] H. Wang, J. Wang, L. Cao, Y. Li, Q. Sun, and J. Wang, "A Stock Closing Price Prediction Model Based on CNN-BiSLSTM," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/5360828.
- [28] LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D., 1990. Handwritten Digit Recognition With A Backpropagation Network. In *Advances in Neural Information Processing systems*, 396-404.
- [29] Cortes, C. & Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20, 273–297.
- [30] Vapnik, V. (1999). The nature of statistical learning theory. Springer science & business media.