POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

Bitcoin price direction prediction using machine learning on a very small dataset

Çok küçük veri seti üzerinde makine öğrenmesi ile bitcoin fiyat yönü tahminlemesi

Yazar(lar) (Author(s))*: Kağan ÖKTEM[1], Adem TEKEREK[2]*

ORCID[1]: 0000-0003-3601-5434

ORCID[2]: 0000-0002-0880-7955

To cite to this article: Öktem K. and Tekerek A., "Bitcoin Price Direction Prediction Using Machine Learning on a Very Small Dataset", ***Journal of Polytechnic***, *(*): *, (*).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Öktem K. ve Tekerek A., "Bitcoin Price Direction Prediction Using Machine Learning on a Very Small Dataset", ***Politeknik Dergisi***, *(*): *, (*).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1667403

Bitcoin Price Direction Prediction Using Machine Learning on a Very Small Dataset

Highlights

- *Testing the model using a dataset with only one feature and 2,550 rows.*
- *Overfitting analysis by splitting the dataset into 56% training, 20% testing, and 24% validation.*
- *Updating the dataset every two days, repeating tests, and verifying consistent performance metrics.*
- *Predicting daily price direction based on percentage change forecasts to improve accuracy.*
- *The utilization of the BOP indicator, which independently enhances prediction accuracy.*

Graphical Abstract

A study is conducted to predict the daily direction of Bitcoin prices using Random Forest Regressor, an ensemble model, on a small dataset in a fast and reliable manner. The study ensures an agile framework through the timeliness of the dataset, validation methods, and model performance.

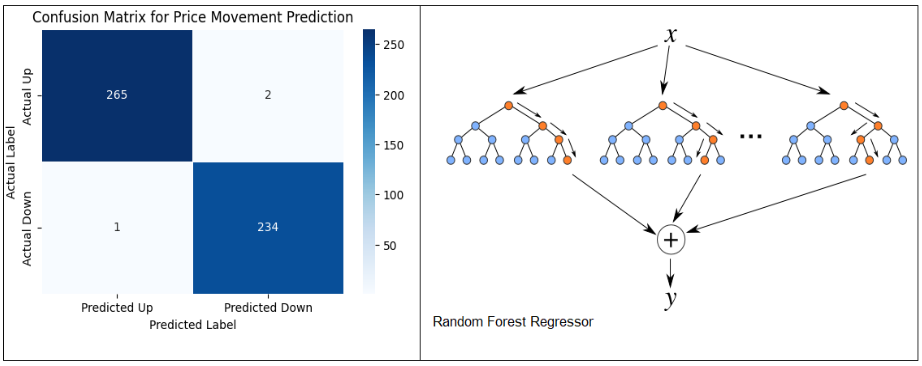

Figure. BTC-PDPR Confusion Matrix and Random Forest Regressor Ensemble Scheme

Aim

The aim is to predict BTC's daily price direction using Machine Learning with the smallest possible dataset and minimal number of features, while achieving the highest accuracy.

Design & Methodology

The dataset, split into 56% training, 20% testing, and 24% validation, is processed using Randomized Search CV to determine the feature importance ranking with only one feature. Subsequently, the ensemble-based Random Forest Regressor algorithm is executed.

Originality

A study utilizing Random Forest Regressor and the impact of the BOP indicator on BTC price direction, achieving consistent accuracy despite using only one feature on a very small dataset of 2,550 rows, with updates every two days, has not been conducted before.

Findings

The use of a validation dataset, the selection of an indicator that facilitates price direction prediction, and the integration of ensemble models collectively lead to successful outcomes.

Sonuç (Conclusion)

For BTC price prediction, accurate feature selection, the right algorithm choice, and the most up-to-date dataset enable the achievement of highly accurate results in a very short time.

Declaration of Ethical Standards

The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Bitcoin Price Direction Prediction Using Machine Learning on a Very Small Dataset

Araştırma Makalesi / Research Article

Kağan ÖKTEM[1*], Adem TEKEREK[2]

[1]Computer Engineering Department, Graduate School of Natural and Applıed Sciences, Gazi University, Ankara, Türkiye
[2]Computer Engineering Department, Technology Faculty, Gazi University, Ankara, Türkiye

(Geliş/Received : 28.03.2025 ; Kabul/Accepted : 17.06.2025 ; Erken Görünüm/Early View : 25.07.2025)

ABSTRACT

Investment advisory services are now commonly offered by consulting firms with financial experts, typically for a monthly fee. Financial markets require specialized knowledge, but advancements in artificial intelligence have revolutionized this field. Deep learning algorithms, especially Long Short-Term Memory and Gated Recurrent Unit, are widely used to predict asset price trends in nonlinear time-series data. However, they demand large datasets and are prone to overfitting. Recently, combining deep learning with reinforcement learning has shown promise, though it requires intensive research and computational resources. This study introduces the Bitcoin Price Direction Prediction Robot model, which predicts Bitcoin's daily price direction using the Random Forest Regressor. As an ensemble-based machine learning model, it works effectively with smaller datasets and identifies key technical indicators influencing price trends. The model achieved a 99.20% accuracy rate on data from March 2018 to the present. It runs efficiently in Google Colab (v5e1 configuration), producing results in just 22 seconds. This paper outlines the methodology, reviews relevant studies from 2017 to 2024, highlights gaps in the literature, and emphasizes the study’s contributions to the field.

Anahtar Kelimeler: BTC, price direction prediction, random forest regressor, technical indicators, machine learning, very small dataset.

Çok Küçük Veri Seti Üzerinde Makine Öğrenmesi ile Bitcoin Fiyat Yönü Tahminlemesi

ÖZ

Günümüzde yatırım danışmanlığı, profesyonel olarak sunulan yaygın bir hizmettir. Bu hizmeti sağlamak için danışmanlık firmaları kurulur ve finans uzmanları istihdam edilir. Hizmetten faydalanmak isteyen bireyler aylık ücret ödeyerek danışmanlık alır. Finansal piyasalar uzmanlık gerektirir. Ancak yapay zekâ sistemlerinin gelişmesiyle, bu alanda da önemli dönüşümler yaşanmıştır. Özellikle Uzun-Kısa Vadeli Bellek ve Geçitli Yinelemeli Birim gibi derin öğrenme algoritmaları, doğrusal olmayan zaman serisi verilerle kısa, orta ve uzun vadeli fiyat tahminlerinde kullanılmaktadır. Ancak bu yöntemler büyük veri setleri gerektirir ve aşırı öğrenmeye yatkındır. Derin öğrenme ile pekiştirmeli öğrenmenin birlikte kullanımı başarılı sonuçlar verse de, bu modellerin entegrasyonu yoğun araştırma ve hesaplama gücü gerektirir. Bu çalışmada, Random Forest Regressor kullanılarak Bitcoin için günlük fiyat yönünü tahmin eden Bitcoin Fiyat Yönü Tahminleme Robotu modeli tanıtılmaktadır. Topluluk modellemesi tabanlı bu makine öğrenimi modeli, büyük veri ihtiyacı duymadan, fiyatı etkileyen teknik indikatörleri başarıyla belirleyerek yüksek doğruluk sunmaktadır. Mart 2018’den günümüze kadar olan verilerle yapılan testlerde model, %99.20 başarı oranına ulaşmıştır. Google Colab v5e1 konfigürasyonunda yalnızca 22 saniyede sonuç üretmektedir. Ayrıca çalışmada 2017-2024 arası literatür incelenmiş, eksiklikler belirlenmiş ve bu çalışmanın alana katkısı ortaya konmuştur.

Keywords: BTC, fiyat yönü tahmini, rastgele orman regresyonu, teknik indikatörler, makine öğrenimi, çok küçük veri seti.

1. INTRODUCTION

Investment advisory typically follows two main approaches: technical analysis, which interprets asset price movements using mathematical models, and fundamental analysis, which evaluates intrinsic value through financial data and forecasts. Non-professional investors often rely on financial advisors, paying fees for guidance.

Cryptocurrency markets have seen rapid institutional adoption. The U.S. approved Bitcoin ETFs in January 2024, followed by preliminary Ethereum spot ETF approvals in May 2024 attracting major firms like BlackRock and Fidelity.

In the current literature, various studies have employed Random Forest Regression and other ensemble models to predict Bitcoin prices. These models often aim to forecast absolute closing prices, which can mislead investors due to high volatility and lack of confidence intervals. Additionally, they typically rely on numerous technical indicators without assessing individual feature relevance, increasing model complexity and limiting accessibility. Most notably, overfitting is seldom addressed, and outdated datasets are often used without periodic updates, undermining model reliability.

To bridge these gaps we have specifications given below:

1. This study proposes a streamlined and interpretable model that predicts daily percentage price changes using

*Sorumlu Yazar (Corresponding Author)
e-posta : kagan.oktem@tubitak.gov.tr

only one feature and the calculated predicted value: Balance of Power (BOP) indicator and close price percentage change.

2. The model is built exclusively using the Random Forest Regressor algorithm, which naturally benefits from ensemble learning by aggregating multiple decision trees to improve generalization and reduce variance. This ensemble nature eliminates the need for more complex hybrid models, making the approach computationally efficient and cost-effective for practical investor use.

3. A key contribution of this work is the identification of the BOP indicator as a highly influential and underutilized predictor in crypto price movement, which offers a novel addition to the literature.

4. Furthermore, the study empirically determines optimal percentage-based dataset splits for training (56%), validation (24%), and testing (20%), providing a reproducible and generalizable methodology for mitigating overfitting. Overfitting is explicitly assessed by comparing RMSE values between validation and test sets—if the test RMSE exceeds the validation RMSE by more than 10%, the model is flagged as overfitting.

5. With a bi-daily updated dataset and repeated validation on recent data, this model offers a reliable, cost-efficient, and lightweight forecasting tool with significant methodological and practical relevance.

Following the introduction, Section 2 reviews related work and provides a comparison table highlighting the most relevant studies. Section 3 defines the problem. Section 4 details the proposed method, includes validation graphs, test results from various machine learning algorithms, and model outputs. Section 5 outlines contributions to the literature. The conclusion summarizes findings and suggests future research directions.

2. RELATED WORK

This section reviews the literature on price forecasting and investment advisory services from 2017 to the present. Various approaches have been explored, including statistical methods, artificial intelligence algorithms, and direct utilization of technical indicators, market sentiment analysis, and hybrid models that combine multiple techniques.

2.1. Utilization of Statistical Methods

The Autoregressive Integrated Moving Average (ARIMA) model is a widely used statistical method for time series forecasting. One study compared ARIMA with LSTM using daily closing prices of Brazil's top 50 companies (2012–2022), with 90% training and 10% testing data. Evaluation via MAPE and RMSE showed that while ARIMA performed well in short-term, stable markets, it underperformed in volatile or long-term conditions [4].

Another study used 1,045 monthly data points from the M3 competition to compare eight traditional statistical methods with several ML techniques (e.g., MLP, RBF, BNN, SVR, GP). Traditional models generally outperformed ML approaches in both accuracy and practical use. The authors emphasized the need for better data preprocessing, computational efficiency, and overfitting control for ML to become more viable [5].

A third study forecasted Bitcoin prices using ARIMAX and LSTM-based RNNs, incorporating Twitter sentiment. Both achieved MSEs below 0.14%, but LSTM faced optimization and computational challenges. While LSTM adapted better to market changes, it did not surpass ARIMAX highlighting the need for further refinement or hybrid models [6].

2.2. Utilization of Supervised Machine Learning Methods

A price prediction study was conducted using regression and classification models on daily opening, closing, highest, and lowest stock prices of companies listed in the S&P 500 index. Multiple regression and classification models were tested, and the results were compared. Some regression models achieved an accuracy rate of nearly 95%. The most successful one was the Random Forest Regressor, which achieved the highest accuracy rate of 99.57%. Among classification models, the highest accuracy rate was obtained by Support Vector Machines (Linear Kernel) with 68.41%. It was observed that feature selection significantly impacts the model's performance. In addition, the time period of the study, market behavior patterns, and the low volatility environment had a notable manipulative effect on the obtained results [6].

Another study used Prophet, an algorithm developed by Facebook that applies supervised learning for time series forecasting, to predict Bitcoin prices. Prophet is widely used due to its ease of implementation and compatibility with Python and R, two of the most popular languages for artificial intelligence applications. This algorithm incorporates seasonality, trends, and holidays into its forecasts by default, allowing for more realistic predictions. However, its accuracy is highly dependent on historical data, and its effectiveness in highly volatile and high-entropy cryptocurrency markets remains limited to specific conditions [7].

In another study covering the 2008–2018 period, stock price forecasting was performed using by OHLCV (Open Price, High Price, Low Price, Close Price, Trading Volume) values of Amazon, Apple, Microsoft, Tesla, IBM, and Facebook. Additionally, public sentiment data collected from Twitter was analyzed to enhance prediction accuracy. Several machine learning models were tested, including regression models such as Logistic Regression, Random Forest, XGBoost and Naïve Bayes, and SVM (Support Vector Machines). The Stacking (Meta-Learning Approach) model performance was evaluated using F1-score and it has the highest accuracy of 60% [8].

In 2023, a study examined the performance of supervised machine learning methods and ensemble models in predicting Bitcoin prices using a small dataset containing 1,972 records and 24 features. The dataset was divided into 80% for training and 20% for testing purposes. The study compared traditional regression models such as Linear Regression, Support Vector Regression, Artificial Neural Network Regression, Decision Tree Regression, Gaussian Process Regression, and KNN, as well as ensemble models such as Random Forest, XGBoost, Histogram-Based Gradient Boosting Regression, and Gradient Boosting Regression. The Gradient Boosting Regression model achieved the highest performance with 99.5% accuracy, while Random Forest Regression ranked second with 99.44% accuracy [9].

In another study conducted in January 2023, a medium-sized dataset containing 53,940 records related to diamond prices was analyzed using data obtained from Kaggle. The dataset was divided into 70% for training and 30% for testing purposes. Price prediction was performed using XGBoost Regression, Linear Regression, Decision Tree, Random Forest, and CatBoost Regression models. According to the R^2 score, the Random Forest Regression model outperformed all other methods [10].

2.3. Utilization of Deep Learning Methods

Deep learning excels at extracting meaningful patterns from complex datasets but faces challenges such as hyper parameter tuning, data requirements, and high computational demands. Numerous studies have explored its application in time series price forecasting.

One study used a 2D-CNN to predict price movements of 12 ETFs, including S&P 500 and commodity ETFs, over a 10-year period with 2.5 million records. Features included OHLCV and technical indicators (SMA, RSI and MACD). The model achieved 72% daily prediction accuracy but struggled with overfitting and high volatility [11]. LSTM models, known for handling long-term dependencies, are widely used in financial forecasting. A study on TATA stock prices highlighted the need for continuous evaluation in volatile markets to maintain accuracy [12].

Another study integrated CNN, LSTM, BiLSTM, and GRU models to forecast cryptocurrency price drops using Kaggle's "Cryptocurrency Historical Prices" dataset. While the model achieved a low RMSE of 0.0089, the ensemble approach introduced heavy computational load and was less effective under rapidly changing conditions [13].

2.4. Utilization of Reinforcement Learning and Deep Learning Methods

Reinforcement Learning (RL) differs from other machine learning approaches by using a reward mechanism, along with a Critic module to evaluate past actions and a Replay Buffer to store experiences—mirroring human learning processes. One study combined social media comments and stock price data, using BERT for sentiment analysis and Deep Q-Learning for prediction. Stocks from Borsa Istanbul (Garanti Bank, Akbank, Türkiye İş Bankası) were analyzed for trend prediction. Among several RL methods, CA-DDDQN achieved the highest returns, though reliance on sentiment analysis and high computational cost were noted as limitations [14].

Another study on NASDAQ 100 and SSE 50 used GRU, ALSTM, and Transformer models, with min-max scaling and time windowing to support RL and ensemble forecasting. Outputs were fed into a DDPG model, which dynamically reweighted predictors over time. The method outperformed traditional strategies (SMA, EMA, Mean Reversion, Momentum) and ML models (CNN,

Table 1. BTC-PDPR Model Comparison Table

Proposed Method Comparison	Dataset / Size	Dataset Usage	Prediction Style	Overfitting Analysis	Dataset Constantly Updated	1 Feature	Accuracy
Momentum & Volatility Index [19]	S&P 500 Companies / Medium (1771 Rows x 13 Features)	Test and Train Split	Price	No	No	No	99.57%
Facebook Prophet Time Series [8]	Bitcoin / Small (1825 Rows x 6 Features)	Test and Train Split	Price	No	No	No	X
Textual Sentiment and SVM-XGBoost [22]	NASDAQ / Big (2800 Rows x 8 Features and 170,000 Tweets)	Test and Train Split	Price	No	No	No	60%
ML Performance Comparison for BTC Price - Gradient Boosting Regression [22]	Bitcoin / Medium (1972 Rows x 24 Features)	Test and Train Split	Price	No	No	No	99.49%
Diamond Price Prediction With ML - Random Forest Regression [21]	Diamond / Big (53,940 Rows x 10 Features)	Test and Train Split	Price	No	No	No	97%
BTC-PDPR	**Bitcoin / Very Small (2550 Rows x 1 Feature)**	**Test, Train, Validation Split**	**Price Movement**	**Yes**	**Yes**	**Yes**	**99.20%**

RNN, LSTM, Q-Learning, SARSA, Decision Trees, ANN), reducing MSE by 21.4% for SSE 50 and 28.6% for NASDAQ 100 [15, 16]. A separate study analyzed stock data (2010–2018) from companies like Apple, Microsoft, and Goldman Sachs, using TD3—a reinforcement learning algorithm that mitigates overestimation via Double Critic networks and improves exploration with Action Noise. The model incorporated transaction costs and portfolio changes in the reward function, while a fine-tuned BERT model handled sentiment analysis [17]. Performance metrics included average returns and a Sharpe ratio of 2.68, with an $110,308 annual profit on test data [18].

Finally, a study on Bitcoin and Gold investments (2013–2022) applied a multi-layered attention-based LSTM to extract features and forecast prices. State representations were input into RL models (DQN, Double DQN, Dueling DQN), with the Dueling DQN yielding the best results—growing an initial $1,000 to over $900,000 in five years [19]. The related work and BTC-PDPR model comparison can be shown in Table 1.

Column descriptions and definitions in Table 1. :

1. Proposed Method Comparison: Provides the abbreviated names of comparative studies involving time series-based financial asset prediction. Includes either the primary algorithm used or the target asset.

2. Dataset / Size: Summarizes the dataset used in each study, including the total dataset size (calculated as number of features × number of rows).

3. Dataset Usage: Describes how the dataset is partitioned (e.g., training/validation/test split) and how the data was utilized in the modeling process.

4. Prediction Style: Indicates whether the prediction task was performed using direct price forecasting or price direction estimation based on percentage change.

5. Overfitting Analysis: States whether the study evaluated or reported any analysis to detect or prevent model overfitting.

6. Dataset Constantly Updated: Identifies whether the dataset used in the study is periodically updated to reflect real-time market conditions.

7. 1 Feature: Specifies whether the model was trained using only one feature, indicating model simplicity and potential interpretability.

8. Accuracy: Reports the performance metric achieved in the study, typically referring to price direction prediction success or classification accuracy.

3. PROBLEM DEFINITION

In the chaotic and constantly changing conditions of the cryptocurrency ecosystem, the probability of individual investors making a profit is significantly low. This probability decreases even further in leveraged trading, where price fluctuations can result in rapid liquidations. Moreover, professional investment advisory services are often not financially accessible to everyone. Even when investors rely on financial experts, short-term trading (hourly, daily, or weekly) still carries a high risk of losses.

In this study, we propose the Bitcoin Price Direction Prediction Robot (BTC-PDPR), a machine learning-based model designed to predict Bitcoin's price direction using a limited dataset. Given that Bitcoin follows discernible long-term trends, our model aims to provide an effective forecasting tool. The tool can help investors for trading decisions without requiring access to large-scale financial resources or expert advisory services.

4. PROPOSED METHOD

The BTC-PDPR model depends on the Random Forest Regressor algorithm, which is designed to predict Bitcoin's daily price direction rather than its absolute price change. Overall design of this model can be shown in Figure 1. Instead of forecasting the daily price movement, the model predicts the daily percentage change in price by just using BOP indicator value. Additionally, MinMaxScaler is applied to the input feature to transforming it into values between 0 and 1 with precisions included to improve model stability and performance. BOP values are mapped into like 0.44664352, 0.61022766, 0.07322616, 0.92371537, 0.84028586 etc. This approach normalizes the target variable to a certain extent, reducing the effect of absolute price fluctuations.

By its nature, the Random Forest Regressor operates as an ensemble-type learning algorithm, combining multiple decision trees to enhance prediction accuracy. This characteristic ensures that the model maintains high accuracy and adaptability in varying market conditions.

4.1. Dataset Details

The dataset includes daily Bitcoin closing prices and BOP (Balance of Power) indicator values from March 2018 to March 2025. It consists of a total of 2,550 rows and 1 feature. However, based on research findings, only the Balance of Power (BOP) indicator was retained due to its strong predictive performance during the study.

New data is added to the dataset every two days to ensure continuous validation of the model's accuracy. The dataset is sourced from two primary providers:

1. CryptoDataDownload.com – Provides historical market data [20].

2. Taapi.io – Supplies daily real-time updates to keep the dataset current [21].

3. Yahoo Finance – Historical BTC Closing Price [22]

4.2. Data Preparation

To ensure data relevance and accuracy, the dataset is regularly updated with real-time and historical data from taapi.io and CryptoDataDownload.com. The original dataset contained 73 features, but a feature selection process was applied to remove low-impact attributes, resulting in a simplified and optimized feature set.

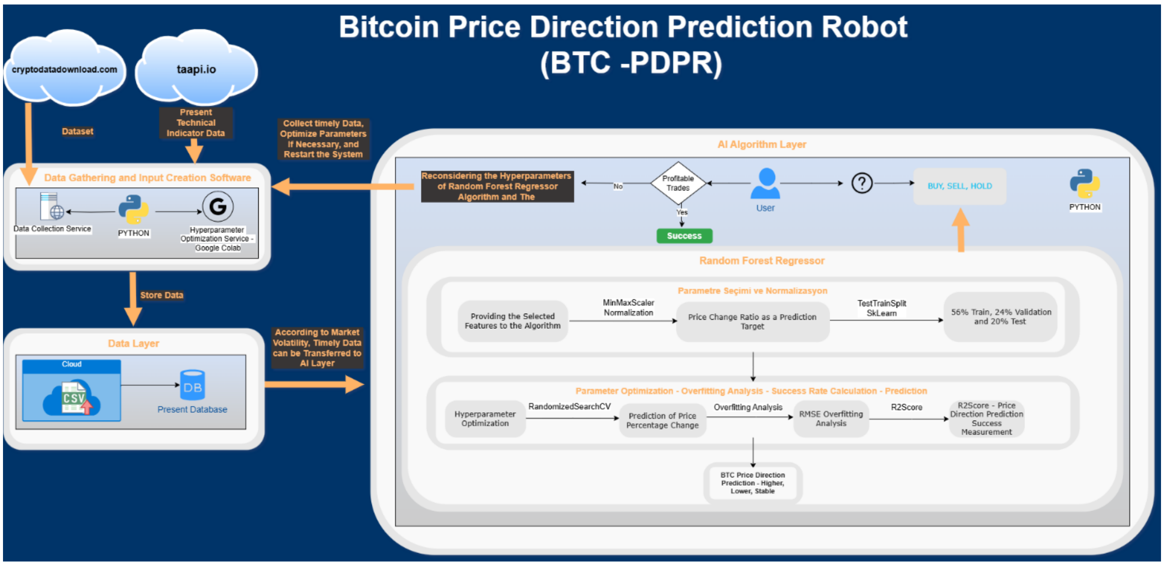

Figure 1. BTC-PDPR Proposed Method

The dataset is structured as a time series, sorted in chronological order from past to present. Non-numeric values have been filtered out to maintain consistency and improve model interpretability. The closing price data in the dataset was transformed into percentage changes between consecutive days and used in the model as close_pct_change. Additionally, all values of BOP indicator were scaled between 0 and 1 including precisions using the MinMaxScaler method to ensure normalization. The column containing the BTC/USD unit symbol was removed from the dataset to eliminate unnecessary attributes.

To maintain the model's predictive accuracy, data updates occur at least once per month or every two days when model performance needs to be re-evaluated and tested.

4.3. Bitcoin Price Direction Prediction Robot (BTC-PDPR) and Component Selection

4.3.1. Data collection and input generation

The Data Collection and Input Generation Software (VTGOY) is responsible for creating and storing the dataset that feeds the BTC-PDPR model. This software also enables the updating of data and parameters when necessary. It is developed in Python and runs on Google Colab, with the dataset stored in Google Drive and updated directly through this implementation. Data design structure can be shown in Figure 2.

In the next step, model moves on running the Random Forest Regressor algorithm, which is developed in Python and executed on Google Colab. The VTGOY software transmits the dataset to next part of BTC-PDPR for training and inference.

During the feature discovery process by us, among hundreds of tested technical indicators, the Balance of Power (BOP) indicator was identified as a key factor. It significantly improved the model's predictive performance, demonstrating a substantial impact on the results. The following parameters are used in the BTC-PDPR model:

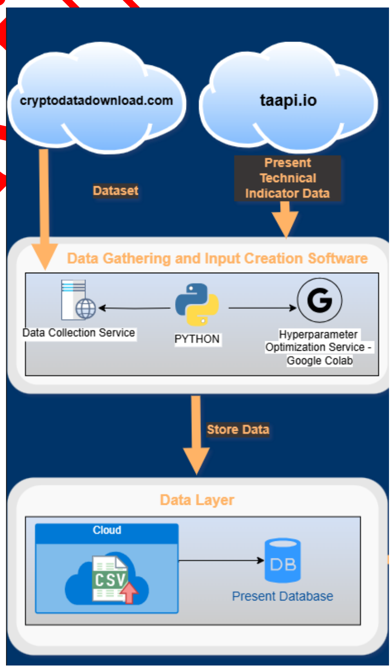

Figure 2. Data Layer

1. Bitcoin daily closing price to calculate price change percentage value as a target for prediction.

2. Balance of Power (BOP) for the training algorithm as a single feature.

This process leads us to improve the model's prediction accuracy while ensuring computational efficiency.

4.3.2. Model's percentage change and price direction mechanism

The raw closing price values are removed from the dataset used for model training, leaving only the Balance of Power (BOP) indicator values. The model learns to predict close price percentage change according to yesterday value (close_pct_change) as the target variable. The computed percentage change in the closing price serves as the prediction target for the model. Percentage change in the closing price is calculated as follows:

$$Close\_Pct\_Change_t = \left(\frac{(Close_t - Close_{\{t-1\}})}{Close_{\{t-1\}}}\right) \times 100 \quad (1)$$

1. $Close_t$: Given time t closing price

2. $Close_t$: Yesterday (t-1) closing price

3. $Close_{\{t-1\}}$: Daily price percentage change

BTC-PDPR model uses scaled values between 0 and 1 with precisions included and BOP values to predict close price percentage change at given time t values. These predicted values can be positive or negative according to input and this means:

1. Using input feature, the model is trained to forecast percentage change in price at given time t.

2. The predicted values (e.g., y_test_pred) are in percentage format and can be both positive and negative.

3. A positive percentage change indicates a price increase (e.g., +2.3%).

4. A negative percentage change indicates a price decrease (e.g., -1.5%).

5. A value close to zero suggests that the price remains almost unchanged.

The direction of price movement is determined by evaluating the sign of the predicted percentage changes (y_test_pred):

predicted_movement=np.sign(y_test_pred)predicted_movement = np.sign(y_test_pred)predicted_movement=np.sign(y_test_pred)

The np.sign() function classifies the predicted percentage changes as follows:

1. Positive prediction (+): Indicates an expected price increase.

2. Negative prediction (-): Indicates an expected price decrease.

3. Zero (0): Indicates no significant price change.

The predicted movement is compared with the actual price movement, which is also converted to its sign representation:

actual_movement = np.sign(y_test) {True direction of percentage change}

correct_directions=(actual_movement==predicted_movement)

This comparison of (t-1) day evaluates the model's accuracy in forecasting the correct price direction by checking how often the predicted and actual movements match.

4.3.3. Price direction prediction layer

The Random Forest Regressor algorithm was developed by Leo Breiman in 2001. It consists of multiple decision trees. These trees are independent and identically distributed. Each tree votes for a specific class. The highest number of votes for a class is selected as the final prediction. In the regression case, Random Forest Regressor computes the arithmetic mean of all decision tree predictions and returns this as the final estimate. The Random Forest method typically has a lower generalization error and is highly resistant to noise. However, its exact accuracy rates depend on the dataset used [23].

After, BTC-PDPR predicts the price percentage change value for given time t, it calculates a value contains:

1. Up (percentage change bigger than zero)

2. Down (percentage change less than zero)

3. No Change (percentage change equals to zero)

After comparing actual price percentage change and also labeling it as above. Model can calculate the accuracy of its predictions. Some of the outputs are below:

1. Day 497: Actual Movement: **Down**, Predicted Movement: **Down**

2. Day 498: Actual Movement: **Down**, Predicted Movement: **Down**

3. Day 503: Actual Movement: **Up**, Predicted Movement: **Down**

Movement

4.3.3.1. Implementation and key methodology

In this model, the Random Forest Regressor is implemented using the "Sklearn.Ensemble" module in Python. The Random Forest algorithm follows an ensemble learning approach, where more than one weak learner classes are combined to build a strong learner. In Random Forest, decision trees are built from weak learners.

4.3.3.2. Four key steps of the random forest algorithm

1. Bootstrap Sampling (Bagging) – Random subsets of the training data are created using bootstrap sampling. Each decision tree is trained on a different bootstrap dataset, ensuring independence between trees and reducing overfitting.

2. Decision Tree Construction – Each tree is trained on its bootstrap dataset, using information gain or the Gini index as the splitting criterion.

3. Prediction via Regression – The model averages the predictions from all decision trees.

4. Aggregation of Results – Random Forest combines predictions from individual trees to generate a more accurate and generalized output.

4.3.3.3. Preprocessing and data preparation

The dataset is retrieved from Google Drive and 1 key feature is selected. Feature is scaled using the MinMaxScaler method from "Sklearn.preprocessing", normalizing values between 0 and 1 with precisions included. The dataset is split into three subsets using "Sklearn.model_selection":

1. 56% for training

2. 20% for testing

3. 24% for validation

When splitting the dataset into three parts, we allocated the largest portion to the training dataset, as a larger training set allows the model to learn effectively and generalize well. However, to prevent overfitting, the training dataset should not be excessively large.

To detect potential overfitting, we created a validation dataset, which should be comparable in size to the test dataset but slightly larger. This ensures that the model performs better during training while allowing for effective performance evaluation. The validation set should be at least one-fifth of the entire dataset to provide meaningful results.

The remaining portion of the dataset was assigned as the test dataset, ensuring that validation and test sets are similar in size, making them easier to compare and analyze.

The Random_State parameter is set to 42 to ensure reproducibility. This guarantees that, when randomness is required, the same sequence of results is obtained in every model run. This consistency is crucial for evaluating model performance objectively.

The Shuffle parameter is set to False to preserve the chronological order of time-series data. If shuffling were applied, it would mix past and future data, making the model unrealistic for real-world applications.

4.3.3.4. Hyperparameter optimization

After the initial model setup, the hyperparameter optimization step is applied. The key hyperparameters, their values, and their effects are as follows:

1. Number_of_Estimators – Total number of decision trees used in the model. A value between 50 and 150 is assigned. More trees generally increase accuracy, but also extend training time. Therefore, an optimal range suitable for the dataset is selected.

2. Maximum_Depth – Maximum depth of each decision tree. The possible values are 10, 20, or unlimited depth. A greater depth results in a more complex model that can capture intricate patterns but may lead to overfitting. Thus, the model is balanced within this range.

3. Minimum_Samples_Division – Minimum number of samples to divide a node. A value between 2 and 10 is assigned. A higher value ensures more generalized learning, while a lower value leads to more splits and increased model complexity. A balanced assignment is used to optimize performance.

4. Minimum_Samples_Leaf – Minimum number of samples in a leaf node. A value between 1 and 5 is assigned. A higher value helps prevent overfitting and improves generalization, but an appropriate range is chosen to maintain model stability.

4.3.3.5. "RandomizedSearchCV" for hyperparameter optimization

The RandomizedSearchCV method is employed to optimize hyperparameters. Instead of exhaustively testing all combinations, it randomly samples a subset to find the best combination efficiently. Five key evaluation criteria are considered:

1. Parameter_distributions – Defines the range of values for each hyperparameter. The values mentioned above are passed to this parameter.

2. Iteration_Count – The number of random hyperparameter combinations to be tested. This is set to 30, ensuring an optimal balance between computational cost and performance.

3. CV (Cross-Validation) – The number of cross-validation folds. A value of 5 is used to improve model generalization and prevent overfitting.

4. Scoring – The metric used for calculation of model's performance. "neg_root_mean_squared_error" was chosen to minimize RMSE, optimizing predictive accuracy.

5. Random_State – Controls randomness to ensure that results are reproducible and performance evaluation is consistent. This is set to 42.

4.3.3.6. Advantages of randomizedsearchcv

The RandomizedSearchCV method provides a model trained with the best hyperparameter combination while avoiding the computational cost of testing every possible combination. Focusing on randomly selected subsets achieves efficient hyperparameter tuning with lower computational expense while delivering highly optimized results. The details of AI prediction layer of this model can be shown in Figure 3.

4.3.4. Experimental study and evaluation method

The compact dataset and algorithm developed for this study include Bitcoin's daily closing price and BOP (Balance of Power) data from March 2018 to the present. The model was executed in multiple consecutive runs, generating predictions within 22 seconds regarding whether the given t time price direction would be upward, downward, or unchanged.

4.3.4.1. Evaluation metrics

The model's performance was calculated using the RMSE and R^2 Score methods.

The coefficient of determination, denoted as R^2, is used to evaluate the proportion of the variance in the dependent variable that is predictable from the independent variables. It is defined as follows:

$$R^2 = 1 - \left\{\frac{\sum_{\{i=1\}}^{\{n\}}(y_i - \hat{y}_i)^2}{\sum_{\{i=1\}}^{\{n\}}(y_i - \bar{y})^2}\right\} \quad (2)$$

Where:

1. y_i is the actual (observed) value

2. $\hat{y}_i$ is the predicted value

3. $\bar{y}$ is the mean of the observed values

4. n is the number of observations

A value of R^2 closer to 1 indicates better model performance, signifying that the model explains a larger portion of the variance in the target variable.

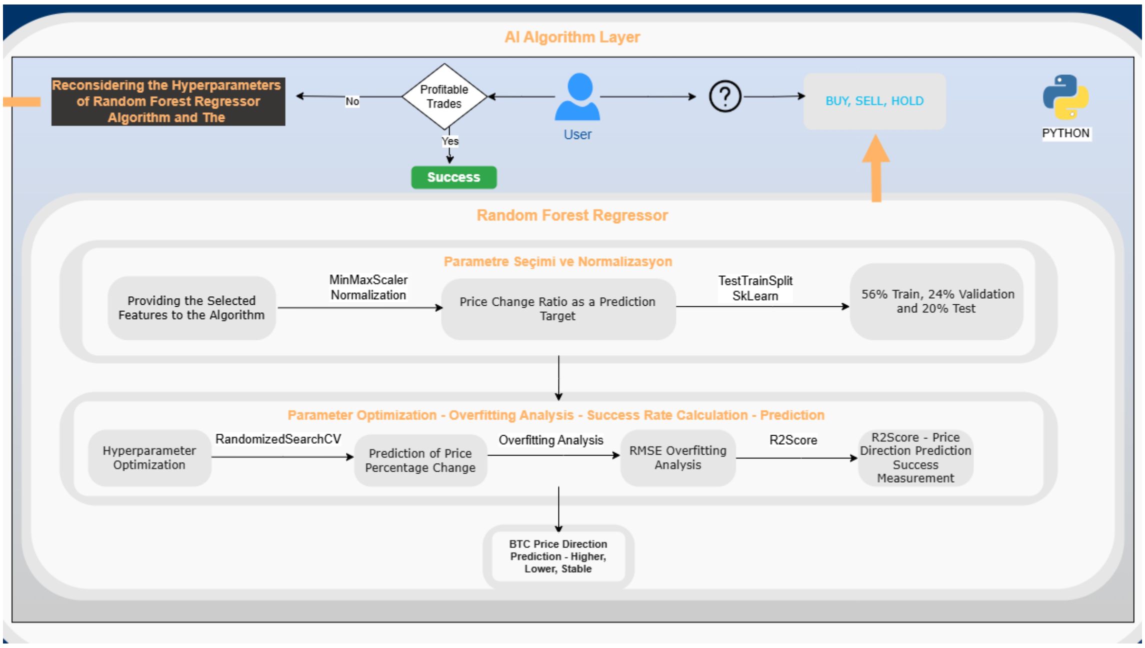

Figure 3. AI Prediction Layer

The Root Mean Squared Error (RMSE) is a widely used metric for evaluating the accuracy of regression models. It measures the average magnitude of the prediction error by taking the square root of the mean of squared differences between predicted and actual values. It is defined as follows:,

$$RMSE = \sqrt{\left\{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right\}} \quad (3)$$

Where:

1. y_i is the actual (observed) value
2. $\hat{y}_i$ is the predicted value
3. n is the number of observations

A lower RMSE value indicates a better fit of the model to the data, implying smaller prediction errors on average.

For the validation dataset:

1. RMSE measures the accuracy of predicting percentage price changes, focusing on the magnitude of errors.
2. R^2 Score evaluates the accuracy of price direction predictions, comparing the actual and predicted price movement directions.
3. RMSE and R^2 Score provide complementary insights
4. RMSE focuses on error magnitude.
5. R^2 Score assesses the correctness of directional predictions.

For the test dataset:

1. RMSE quantifies the error in percentage price change predictions.
2. R^2 Score measures the accuracy of price direction forecasts.

Using the same metrics for both the validation and test datasets ensures that results are directly comparable

4.3.4.2. Overfitting analysis

To detect overfitting, the difference between the RMSE values of the validation and test datasets was analyzed. If the test RMSE is significantly higher than the validation RMSE (by a factor of 1.1 or more), it indicates overfitting—meaning the model has over-adapted to the validation dataset. Conversely, if the difference remains within a reasonable range, the model is considered to have learned the data properly without overfitting. In our evaluation, the RMSE difference was only 1.0532×, confirming that overfitting did not occur.

4.3.4.3. Accuracy of price direction predictions

The percentage accuracy of price direction predictions was calculated by comparing the actual and predicted price movements. The accuracy percentage was reported based on the test dataset, indicating the model's effectiveness in forecasting future price movements. For 500 days in our test dataset, the predicted and actual price percentage change graphic can be shown in Figure 4.

4.3.4.4. Confusion matrix

By focusing on predicting the direction of price movement (increase or decrease), we can frame this as a binary classification problem. Confusion matrix for price movement prediction can be shown in Figure 5. In this case:

1. Actual price movement (actual_movement): Whether the price has increased compared to the previous day.
2. Predicted price movement (predicted_movement): The direction of price change as forecasted by the model.

The confusion matrix can be defined as in Table 2.

1. TP (True Positive): The model predicted a price increase, and the price actually increased.
2. FN (False Negative): The model predicted a price decrease, but the price actually increased.

Table 2. BTC-PDPR Confusion Matrix Definition Table

Actual Movement	**Predicted Increase (Up)**	**Predicted Decrease (Down)**
Actual Increase (Up)	TP (True Positive)	FN (False Negative)
Actual Decrease (Down)	FP (False Positive)	TN (True Negative)

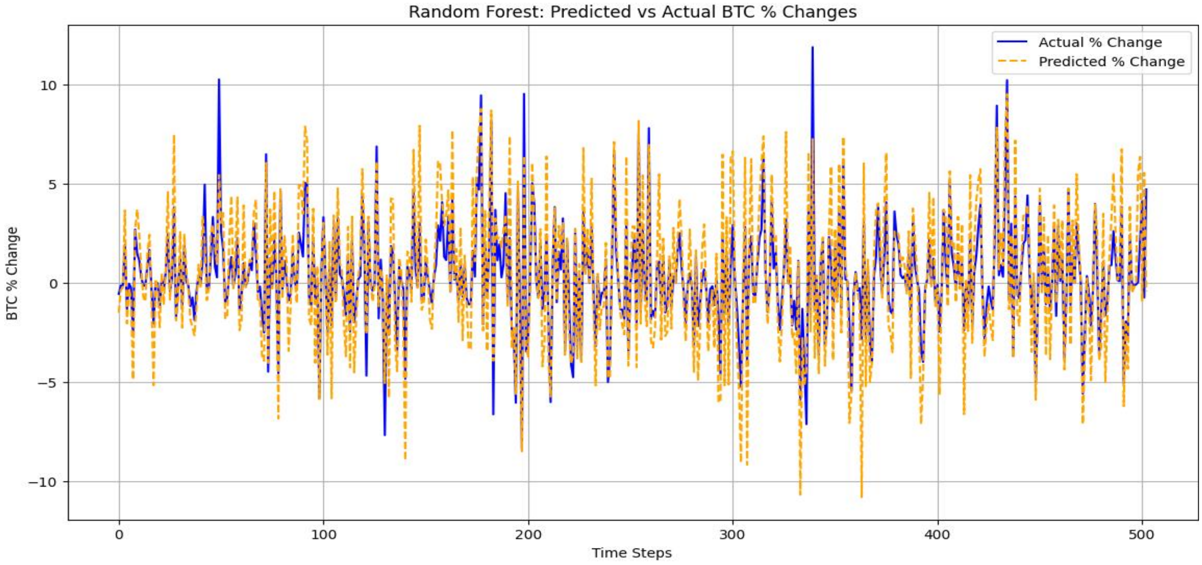

Figure 4. BTC-PDPR Model Price % Change Predicted Values & Actual % Change of Prices in 500 Days Result

3. FP (False Positive): The model predicted a price increase, but the price actually decreased.

4. TN (True Negative): The model predicted a price decrease, and the price actually decreased.

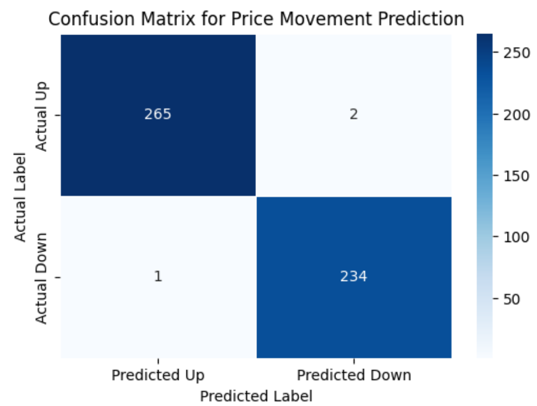

Figure 5. Price Movement Direction Confusion Matrix

4.3.5. Impact of the bop (balance of power) indicator

The BOP indicator was developed by Igor Livshin [24]. The Balance of Power (BOP) indicator measures the relative strength between buyers and sellers by analyzing price movements within a given trading period. It is calculated using the open, high, low, and close prices of the asset.

$$\mathrm{BOP} = \frac{\mathrm{High} - \mathrm{Low}}{\mathrm{Close} - \mathrm{Open}} \tag{4}$$

1. Close: Closing price
2. Open: Opening price
3. High: Highest price during the period
4. Low: Lowest price during the period

The values range between -1 and 1, where positive values indicate stronger buying pressure, while negative values indicate stronger selling pressure.

Applications in Technical Analysis:

1. Identifying price trends,
2. Assisting in momentum analysis,
3. Detecting trend reversals, often used in combination with moving averages.

The influence of the BOP indicator on our algorithm is so significant that replacing it with any other indicator dramatically reduces accuracy rates. In various experiments, alternative indicators were tested in place of BOP, and the results showed a substantial decline in performance:

1. Replacing BOP with the 14-day RSI (Relative Strength Index) resulted in a 55.87% accuracy rate for BTC price direction prediction.
2. Replacing BOP with the Stochastic RSI indicator reduced the accuracy to 53.28%.
3. Replacing BOP with the MACD (Moving Average Convergence Divergence) led to an accuracy of 52.29%.
4. Replacing BOP with the MFI (Money Flow Index) resulted in 51.65% accuracy.

Even the most widely used moving average indicators in technical analysis produced poor results:

Table 3. Comparison of Results by Testing Different Algorithms Instead of Random Forest Regressor in the BTC-PDPR - Core Prediction Model Comparison Results Table

Prediction Model	Dataset	Price Percentage Change Prediction RMSE	Price Direction Prediction R2-Score	Price Direction Prediction Accuracy
BTC-PDPR(Random Forest Regressor)	**Test**	**1.78**	**0.97**	**99.20%**
BTC-PDPR(Random Forest Regressor)	**Validation**	**1.69**		
BTC-PDPR (GRU)	Test	3.84	-0.87	53.50%
BTC-PDPR (GRU)	Validation	2.71		
BTC-PDPR (LSTM)	Test	3.84	-0.87	53.50%
BTC-PDPR (LSTM)	Validation	2.71		
BTC-PDPR (XGBoost)	Test	1.31	0.82	95.43%
BTC-PDPR (XGBoost)	Validation	1.405		

1. Replacing BOP with the 50-day Moving Average (MA) led to an accuracy rate of 51.49%.

2. Replacing BOP with the 50-day Exponential Moving Average (EMA) resulted in an accuracy of 49.74%.

These are just a few of our tests. In total, over 70 different indicators were evaluated, and BOP consistently delivered the highest accuracy. Additionally, when replacing the BOP indicator with the day's lowest price, the accuracy dropped further to 52.56%. These results highlight BOP as the most effective indicator for predicting BTC price direction in our model.

4.3.6. Comparison of results by testing different algorithms instead of “Random Forest Regressor” in the “BTC-PDPR” model

During the literature review, we observed that LSTM, GRU, and XGBoost were among the most frequently used algorithms for price forecasting models. To ensure a fair evaluation, we conducted the same experiments using these algorithms. Experimental comparison results can be shown in Table 3.

4.3.6.1. GRU comparison

For the GRU model, we used 128 hidden units and 2 layers in its configuration. To reduce overfitting, a dropout rate of 0.2 was applied during training. However, due to the lack of error reduction during training, the model, which was initially set to run for 60 epochs, triggered early stopping after only 5 epochs.

The model’s error metrics were measured using the Mean Squared Error (MSE):

1. Test dataset MSE: 14.7410

2. Validation dataset MSE: 8.3672

Since the model failed to capture a decreasing error trend, its percentage change prediction error was evaluated using Root Mean Squared Error (RMSE), which resulted in 2.71. For price direction prediction, the R^2 score was -0.87, which translates to a very low accuracy rate of 53.5%. These results indicate that the GRU model could not approximate the performance of our BTC-PDPR model on a small dataset. It failed to learn the data, lacked generalization ability, and ultimately could not make reliable predictions.

4.3.6.2. LSTM comparison

The LSTM model, initially set to run for 60 epochs, triggered early stopping after only 6 epochs due to a lack of performance improvement. The model was configured with 128 hidden units, and to prevent overfitting, a dropout rate of 0.2 was applied during training.

The error metrics were measured using the Mean Squared Error (MSE):

1. Test dataset MSE: 14.7410

2. Validation dataset MSE: 8.3751

Since the model failed to capture a decreasing error trend, the percentage change prediction error was evaluated using Root Mean Squared Error (RMSE), which resulted in 2.71. For price direction prediction, the R^2 score was -0.87, translating to a very low accuracy rate of 53.5%.

Notably, the GRU and LSTM models produced nearly identical results, highlighting their structural similarities. These findings indicate that, like GRU, the LSTM model was unable to achieve comparable performance to our BTC-PDPR model on a small dataset. The LSTM model failed to learn effectively, lacked generalization, and ultimately could not make reliable predictions.

4.3.6.3. XGBoost comparison

For the XGBoost model, the following hyper parameters were configured:

1. Learning Rate: 0.05

2. Maximum Depth: 6

3. Sub-sample Ratio: 0.8

4. Seed Value: 42

5. Number of Boosting Rounds: 200

6. Early Stopping Rounds: 10

The percentage change prediction error was evaluated using RMSE:

1. Test dataset RMSE: 1.31

2. Validation dataset RMSE: 1.405

The R^2 score was 0.82, with an overall accuracy rate of 95.43%. When compared to BTC-PDPR, this result was approximately 4% lower in accuracy. However, when compared to LSTM and GRU, XGBoost performed significantly better, demonstrating its suitability for small datasets.

4.3.6.4. Comparison conclusion

When we look at the RMSE values for LSTM and GRU, it can be seen that both test dataset's RMSE values are bigger exceeding the maximum 10% difference. Since, it confirmed that overfitting is existed and the accuracy values cannot be trusted. On the other hand, XGBoost algorithm shows no overfitting. But, interesting result is that, validation dataset RMSE value is bigger than the test dataset. In the end, XGBoost algorithm is giving the accuracy rate of 95.43% which is eligible but not as much as our BTC-PDPR model.

5. EXPECTED CONTRIBUTION TO LITERATURE

The ability to predict the price direction of high-risk and volatile assets such as Bitcoin with high accuracy on a daily basis provides valuable insights for future research. Many financial instruments exhibit general price trends, even among high-risk assets. In reality, all assets follow a trend over the long term, influenced by specific economic events.

This study demonstrates that accurate predictions can be achieved using only closing price data and the BOP indicator on a highly compact dataset of approximately 100 KB. The ability to generate high-accuracy predictions on such a small dataset offers a notable contribution to the literature. Additionally, the BTC-PDPR model adapts exceptionally well to real-time data, as it maintains its high accuracy even after updates every two days. While AI algorithms typically perform better on large datasets, this study shows that proper feature selection and an optimized evaluation methodology can yield successful results on small datasets as well.

Furthermore, rather than predicting absolute asset prices, this study focuses on forecasting price direction based on percentage changes, a method that has been shown to enhance both model accuracy and reliability. This approach eliminates the need for direct price forecasting and enables the creation of sustainable daily revenue models.

Another key issue in previous studies claiming high accuracy rates is the lack of overfitting analysis. Many existing models do not incorporate validation datasets, raising concerns about their ability to generalize beyond the training data. In contrast, this study explicitly conducts overfitting analysis, reinforcing the reliability of the BTC-PDPR model. This approach provides a valuable methodology for future research in financial forecasting.

Additionally, a common limitation observed in the literature review is that most existing studies rely on outdated datasets, with no incorporation of recent market data. This study addresses this gap by ensuring that the BTC-PDPR model is updated every two days with real-time data, allowing for continuous accuracy validation.

We believe that this study will serve as a reference for future research in the field of financial market prediction. Moreover, by incorporating leveraged trading, profit potential can be maximized. The concept of predicting price direction using percentage changes introduces a new perspective to financial forecasting. Additionally, this research highlights that daily investment strategies an be derived from long-term asset trends, offering a practical approach for traders and investors.

6. CONCLUSION

The experimental results demonstrate exceptional accuracy in predicting Bitcoin's daily price direction. This model reached an accuracy rate of 99.20% on a randomly selected 20% test dataset, which is a remarkable outcome. The Root Mean Squared Error (RMSE) for the validation dataset was 1.69, while the test dataset RMSE was 1.78, indicating that the model generalized well without overfitting.

The R^2 Score for the validation dataset was 0.99, and for the test dataset, it was 0.97, further confirming the model's high accuracy and robustness. These results suggest that with properly selected feature and well-optimized hyper parameters, a Random Forest Regressor can effectively predict Bitcoin's price direction at a daily level. This capability allows investors to strategically take positions in bullish or bearish markets, potentially enabling a sustainable profit model.

However, when making daily predictions, the model's dependence on technical indicator data should not be overlooked. For future forecasts, these indicators must be predicted before they are actually generated. If technical indicators are not incorporated, the model's accuracy will decrease.

The greatest advantage of our model is that it relies on only one feature, making future research less computationally demanding compared to other studies. Additionally, the effort required for further development will be significantly lower than in more complex models.

The BTC-PDPR method, proposed for financial time series forecasting, has broader applications beyond cryptocurrency markets. It could be adapted for use in health sciences, soil fertility studies, corporate growth forecasting, and other domains that rely on time-series data. The model integrates previously successful methodologies while introducing new BTC-specific feature, providing a novel approach to financial forecasting.

ACKNOWLEDGEMENT

I would like to thank my thesis advisor, Assoc. Prof. Dr. Adem TEKEREK, for his valuable contributions to my study.

DECLARATION OF ETHICAL STANDARDS

The author of this article declares that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

AUTHORS' CONTRIBUTIONS

Adem TEKEREK: Guidance the whole process and check of original draft.

Kağan ÖKTEM: Conceptualization, methodology, investigation, analysis and writing original draft.

CONFLICT OF INTEREST

There is no conflict of interest in this study.

REFERENCES

[1] PhamToan D., VoThiHang N., PhamThi B., "Improving forecasting model for fuzzy time series using the self-updating clustering and bi-directional long short term memory algorithm", ***Expert Systems With Applications***, 241: 122767, (2024).

[2] Mardjo A., Choksuchat C., "HyBiLSTM: Multivariate Bitcoin Price Forecasting Using Hybrid Time-Series Models With Bidirectional LSTM", ***IEEE Access***, 12: 50792-50808, (2024).

[3] Serafini G., Yi P., Zhang Q., Brambilla M., Wang J., Hu Y., Li B., "Sentiment-Driven Price Prediction of the Bitcoin based on Statistical and Deep Learning Approaches", ***Proceedings of the International Joint Conference on Neural Networks (IJCNN)***, Glasgow, UK, 1-8, (2020).

[4] Taslim D. G., Murwantara I. M., "A Comparative Study of ARIMA and LSTM in Forecasting Time Series Data", ***Proceedings of the 2022 International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)***, Semarang, Indonesia, 231-235 (2022).

[5] Makridakis S., Spiliotis E., Assimakopoulos V., "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward", ***PLoS ONE***, 13: e0194889, (2018).

[6] Kumari S. E., Ramesh B., "Prediction of Stock Prices using Machine Learning Regression and Classification Algorithms", ***Proceedings of the 2020 IEEE International Conference for Emerging Technology (INCET)***, Belgaum, India, 1-5, (2020).

[7] Ramya N., Sanjay Roshan R., Vishal Srinivas R., Krishna Prasad D., "Crypto-Currency Price Prediction using Machine Learning", ***Proceedings of the Sixth International Conference on Trends in Electronics and Informatics (ICOEI)***, Tirunelveli, India, 1455-1458, (2022).

[8] Bouktif S., Fiaz A., Awad M., "Augmented Textual Features-Based Stock Market Prediction", ***IEEE Access***, 8: 40269-40282, (2020).

[9] Ahmed F. U., Ahmed M., F. Mahi H., Abdullah S. H., Suha S. A., "A Comparative Performance Evaluation of Bitcoin Price Prediction Using Machine Learning Techniques", ***International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)***, Dhaka, Bangladesh, 194-198, (2023).

[10] Basha M. S. A., Oveis P. M., Prabavathi C., Lakshmi M. B., Sucharitha M. M., "An Efficient Machine Learning Approach: Analysis of Supervised Machine Learning Methods to Forecast the Diamond Price", ***2023 International Conference for Advancement in Technology (ICONAT)***, Goa, India, 1-6, (2023).

[11] Gudelek M. U., Boluk S. A., Ozbayoglu A. M., "A Deep Learning based Stock Trading Model with 2-D CNN Trend Detection", ***Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)***, Honolulu, HI, USA, 1-8, (2017).

[12] Behura J. P., Pande S. D., Ramesh J. V. N., "Stock Price Prediction using Multi-Layered Sequential LSTM", ***EAI Endorsed Transactions on Scalable Information Systems***, 1: 1-8, (2023).

[13] Akter M. S., Shahriar H., Rahman M. A., Rahman M., Cuzzocrea A., "Early Prediction of Cryptocurrency Price Decline: A Deep Learning Approach", ***Proc. 26th International Conference on Computer and Information Technology (ICCIT)***, Cox's Bazar, Bangladesh, 1-6, (2023).

[14] Altuner A.B., Kilimci Z.H., "A Novel Deep Reinforcement Learning Based Stock Price Prediction Using Knowledge Graph and Community Aware Sentiments", ***Turkish Journal of Electrical Engineering and Computer Sciences***, 30: No. 4, (2023).

[15] Zouaghia Z., Kodia Aouina Z., Ben Said L., "Hybrid Machine Learning Model for Predicting NASDAQ Composite Index", ***Proceedings of the 2023 International Symposium on Networks, Computers and Communications (ISNCC)***, Doha Qatar, 1-6, (2023).

[16] Lin W., Xie L., Xu H., "Deep-Reinforcement-Learning-Based Dynamic Ensemble Model for Stock Prediction", ***Electronics***, 12: 4483 (2023).

[17] Haarnoja T., Zhou A., Abbeel P., Levine S., "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor", ***35th International Conference on Machine Learning (ICML 2018)***, Stockholm, Sweden, https://arxiv.org/abs/1801.01290, (2018).

[18] Kabbani T., Duman E., "Deep Reinforcement Learning Approach for Trading Automation in the Stock Market", ***IEEE Access***, 10: 93564-93574, (2022).

[19] Liu R., Guo W., Jin K., Ding Y., Chen Z., Ma S., Li Z., "Computer Intelligent Investment Strategy Based on Deep Reinforcement Learning and Multi-Layer LSTM Network", ***Proceedings of the 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA)***, Dalian, China, 1006-1015, (2022).

[20] ***https://www.cryptodatadownload.com/data/premium/***, "BTC - Bitcoin_technical_analysis (2018 - 2025)", (2025).

[21] ***https://taapi.io/indicators/balance-of-power***, "Technical Analysis Indicators API (Updated Daily BOP Indicator Values)", (2025).

[22] ***https://finance.yahoo.com/quote/BTC-USD***, "BTC Daily Closing Price", (2025).

[23] Breiman L., "Random Forests", ***Machine Learning***, 45: 5-32, (2001).

[24] Livshin I., "Balance of Power", ***Technical Analysis of Stocks & Commodities***, 19: 26-30, (2001).