Machine Learning-Based Forecasting of Bitcoin Price Movements

Darko Angelovski, Bojana Velichkovska, Goran Jakimovski, Danijela Efnusheva and Marija Kalendar
Faculty of Electrical Engineering and Infomation Technologies, SS. Cyril and Methodius University,
Rugjer Boshkovikj Str. 18, 1000 Skopje, N. Macedonia
darkoangelovski666@gmail.com, {bojanav, goranj, danijela, marijaka}@feit.ukim.edu.mk

Keywords: Cryptocurrency, Bitcoin, Machine Learning, Long Short-Term Memory, Random Forest, Gradient Boosting, Light Gradient Boosting.

Abstract: In the volatile realm of cryptocurrency markets, this research explores the intricate dance of Bitcoin price dynamics through the lens of machine learning. Employing a multifaceted approach, we harness the power of Long Short-Term Memory (LSTM) networks, Gradient Boosting, LightGBM (LGBM) Regressor, and Random Forest algorithms to unravel the complexities of price movements. We perform a comprehensive analysis, and observe patterns and dependencies within historical data at hour-long intervals in the last 30 and 45 days, by using a holdout technique with 80% of the data used for training and 20% used for testing. We evaluate the models using four standard regression metrics. The training data incorporates a diverse range of features capturing hourly trends, day-of-the-week variations, and the correlation between opening and closing prices. Our study delves into the ability for forecasting Bitcoin price movements using ensemble algorithms and LSTM. The results show best performance for the LSTM models, especially when trained on longer training intervals. Namely, our LSTM model obtains R2 of 0.98 when trained on 30 days and 0.99 when trained on 45 days. In comparison, the ensemble methods show volatility and lower predictive ability.

1 INTRODUCTION

In the dynamic realm of cryptocurrency markets, the ability to anticipate price movements holds immense significance for investors and traders. In 2008, Nakamoto [1] has introduced an electronic peer-to-peer cash system Bitcoin to the world. Over the last decade, the cryptocurrency market has grown tremendously, whereby individual cryptocurrency prices have exhibited large volatility [2]. Being able to understand and predict said volatile changes is an ongoing challenge, which if successful can significantly influence the cryptocurrency market.

Cryptocurrency relies on the Blockchain [3]. By implementing an access management mechanism, Blockchain systems provide ways of ensuring the privacy and protection of user data. The currency is based on a decentralized peer-to-peer network that creates currencies and management of transactions without central authority. All Bitcoin transactions are posted in blocks to an open directory, which is called Blockchain. As the flagship cryptocurrency, Bitcoin's volatile nature poses a captivating challenge, necessitating sophisticated tools for accurate predictions.

This research focuses on predictive modeling, by employing advanced machine learning (ML) algorithms: Long Short-Term Memory (LSTM) [4], Gradient Boosting [5], Light Gradient Boosting Machine (LGBM) [6], and Random Forest [7].
Our investigation encompasses the challenges faced in preprocessing historical Bitcoin data, the representation of predictive features, and the comparative evaluation of several algorithms in predicting future price changes of Bitcoin. As we navigate the landscape of Bitcoin price prediction, our aim is to contribute valuable insights to the broader discourse on the application of ML in cryptocurrency markets.

2 RELATED WORK

Researchers have already tackled this problem. Majority use deep learning methodologies, however some venture into the application of standard ML approaches. In spite of the approach, both long-term and short-term strategies are tested. In [8], the authors predict Bitcoin prices at one-minute intervals for data collected from 2012 to 2018. The authors of [9] focus on the performance of deep learning models for three popular cryptocurrencies. In [10], the authors use ensemble learning methods and neural networks to analyze the closing price of several cryptocurrencies for a seven-day dataset collected in 2019.

In [11] the authors observe 20-day history of price, volume, and market capitalization in order to preform one-day trading decisions, whereas the research in [12] LSTM is used for data analysis in a span of three years. The results indicate promise in the realm of cryptocurrency trading. Another daily cryptocurrency approach is presented in [13], where the authors predict binary relative daily market movements of the 100 largest cryptocurrencies and their results show statistically viable predictions. In [14], the time analysis performed results with best model performance that carries a mean average error of 227. The authors of [15] assess various neural network approaches for predicting daily Bitcoin prices, and the findings reveal that a gated recurrent unit implementation with recurrent dropout stands out as the top performer on their dataset. In the research detailed in paper [16], advanced artificial intelligence frameworks are harnessed to understand Bitcoin, Ethereum, and Ripple's business dynamics. The study reveals that the Artificial Neural Network capitalizes on longer-term historical data, whereas the LSTM specializes in capturing swift dynamics. In the context of paper [17], the focus is on extracting and comparing the accuracy of Bitcoin predictions using diverse ML.

3 METHODOLOGY

This section outlines the methodology employed for the research, i.e., the data used and its preprocessing, the algorithms applied, and the metrics used for model evaluation.

3.1 Dataset

The bitcoin historical prices were collected using the python-binance API [18]. Two separate analyses were performed. The first considered collecting data in a period of 30 days from 25th of December 2023 at 01:00h to 24th of January 2024 at 01:00h and the second in a period of 45 days from 10th of December 2023 at 01:00h to 24th of January 2024 at 01:00h. In both cases, we consider the following features: open, high, low and close. Additionally, the API returns a datetime information for when each of these features was observed. For the purpose of the study, we do not consider the entire datetime. Instead, we only extract and use the hour of the day and the day of the week. Moreover, for data brevity purposes and light model building, we do not consider the additional data received as response from the API, namely information on volume, number of trades, taker base volume, etc. are excluded. As most features (open, high, low, close) were retrieved as object types, the processing included adequate conversion to a numeric value. The open feature shows the starting price for every hour, the high represents the highest price in each hour, low is the lowest price for every hour, whereas close the closing price or final price for every hour. The additionally added features are self-explanatory. The target variable we are focusing on is the price movement for each hour, i.e., the difference between the starting and ending prices of bitcoin in each hour. The six features which are the input variables for our models and the output variable (price movement) are given in Table 1.

Table 1: Features for the model. The last row of the table is the target feature, which the model should predict.

Feature	Description	Unit
open	start price at every hour	USDT
high	highest price at every hour	USDT
low	lowest price at every hour	USDT
close	end price at every hour	USDT
hour	hour in the day (extracted from datetime)	Integer value [0-23]
week day	day in the week (extracted from datetime)	Integer value
price movement	difference between ending and starting price at every hour	USDT

As the data covers a significant range of values, we normalize the data before proceeding with training. Moreover, the data is divided in a holdout technique with 80% of the data being used for training and the remaining 20% used for testing purposes, for both the 30-day and the 45-day interval. The reason behind observing historical data for every hour in a period of 30 and 45 days specifically is the question whether successful predictions can be made by capturing shorter-term trends and patterns.

3.2 Algorithms

This study focuses on comparing the performance of ensemble methods to that of LSTM neural networks. From ensemble methods, we are analyzing Random Forest (RF), Gradient Boosting (GB) and Light

Gradient Boosting Machine (LGBM). As ensemble models combine predictions of multiple individual models to form their final assessment, they enhance generalization and can improve overall performance. The idea behind ensemble modeling is that by aggregating the predictions of multiple diverse models, the ensemble can often outperform any individual model in terms of accuracy and robustness, which is why we aimed to compare their performance to a LSTM network.

RF, as an ensemble algorithm creates diverse trees by using bootstrapped samples and random subsets of features, with the final prediction is obtained by averaging. This accounts for reducing overfitting, handling noisy data, and for feature importance. GB builds a strong predictive model also by sequentially improving weak learners, often decision trees. It corrects errors made by the previous models, and as such excels in handling complex relationships in data. LightGBM is a gradient boosting framework that efficiently handles large datasets. It uses a tree-based learning algorithm and employs a novel technique called Gradient-based One-Side Sampling to reduce the amount of data used for building trees.

Compared to ensemble algorithms, LSTM networks focus on understanding long-term dependencies in sequential data, and are particularly effective for time-series data, hence the application. The LSTM contain memory cells which can store and retrieve information over extended periods, particularly relevant information.

During the training phase for every ensemble model, we are going to use 100 decision trees, maximum depth of 15 layers per decision tree and additionally, GB and LGBM have a learning rate of 0.001. On the other hand, the LSTM neural network is trained on 100 epochs, batch size of 32, with a callback for early stopping which monitors the root mean squared error with patience of 50 epochs. The LSTM uses Adam optimizer, as an iterative optimization algorithm which can best minimize the loss function. The LSTM model consists of two LSTM layers with 256 units, two dropout layers, one after each LSTM layer, with a dropout rate of 20% and one dense layer with 1 output unit. Moreover, the first LSTM layer uses the hyperbolic tangent function (tanh) as an activation function.

3.3 Metrics

In order to evaluate the performance of the model, we calculated the following metrics: mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and the coefficient of determination R^2. MSE quantifies the average squared difference between the actual and predicted values, and the formula is given in Figure 1. The RMSE shows the average magnitude of the errors between predicted and actual values, and is calculated as the square root of the MSE. The formula is given in Figure 2. MAE quantifies the accuracy of predictions by measuring the average error magnitude between predicted and actual values. Unlike MSE and RMSE, MAE uses the absolute values of the errors. The formula for MAE is given in Figure 3. R2 gives a sense of how well a model captures the relationship between independent and dependent variables. Unlike MSE, RMSE and MAE, R2 is a score between 0 and 1 where a value of 1 means that the model perfectly predicts target variables based on the provided features. In some cases, the score can be negative, meaning the model is not capturing any meaningful relationships between the independent and the dependent variables. The formula for R2 is given in Figure 4 where SSR (sum of squares) defines the sum of squared differences between the predicted value and the actual values and SST (total sum of squares) represents the total sum of squares of the differences between each data point and the mean of the dependent variable (target) in the dataset.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(actual_i - predicted_i)^2$$

Figure 1: Formula for MSE.

$$RMSE = \sqrt{MSE}$$

Figure 2: Formula for RMSE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|actual_i - predicted_i|$$

Figure 3: Formula for MAE.

$$\boldsymbol{R^2} = \mathbf{1} - \frac{\boldsymbol{SSR}}{\boldsymbol{SST}} = \mathbf{1} - \frac{\sum_{i=1}^{n}(\boldsymbol{actual_i} - \boldsymbol{predicted_i})^2}{\sum_{i=1}^{n}(\boldsymbol{actual_i} - \boldsymbol{mean_i})^2}$$

Figure 4: Formula for R-squared.

4 RESULTS

The obtained results observe performance of four algorithms in their ability to predict Bitcoin price movements using the features given in Table 1, i.e., three ensemble RF, GB, and LGBM, and one deep learning approach, i.e., LSTM. We compare two different time intervals, i.e. we observe price movements in the last 30 and last 45 days. The results are given in Table 2. The results in this table show the predictive performance of the models, with values showing the differences between predicted and actual value.

As can be observed, there is a significant error for all ML models when training on 30 days. The worst performance can be observed by GB, with RF giving the best predictions. As RF offers best performance from all ensemble algorithms, only these predictions are shown in the figures. The performance of the RF model compared to the original price movements is given in Figure 5, with the actual movements denoted with green, and the predicted movements in purple. As the figure shows, RF can almost always predict the direction of the price movements. However, the model struggles with following the amplitude of the change, which is also demonstrated by the MAE of 96.91 seen in Table 2. This can be distinctly and firstly noted around the 10th hour, where the prediction detects a drop in the data, even if on smaller scale compared to the original. Also, the results for RF around the 45th hour in, predicted price movements are in totally different direction compared to the actual price movements.

Table 2 Results from training the algorithms with 30-day and 45-day intervals. Three metrics are observed.

Algorithm	MSE	RMSE	MAE	R^2
30 days				
RF	56945.22	238.63	96.91	0.11
GB	63313.60	251.62	149.01	0.01
LGBM	62864.45	250.73	150.95	0.01
LSTM	641.77	25.33	15.63	0.98
45 days				
RF	18857.53	137.32	76.01	0.66
GB	54632.86	233.74	149.96	0.02
LGBM	52927.02	230.06	143.10	0.05
LSTM	257.29	16.04	11.90	0.99

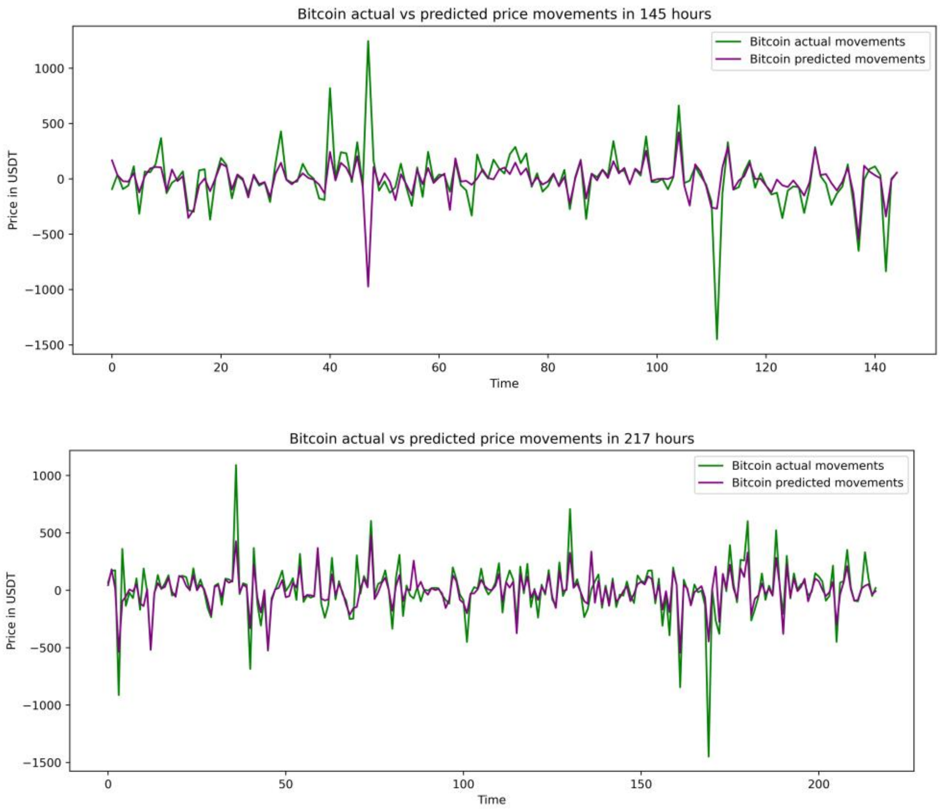

Figure 5: Actual and predicted price movements for RF trained on a 30-day period (above) and 45-day period (under).

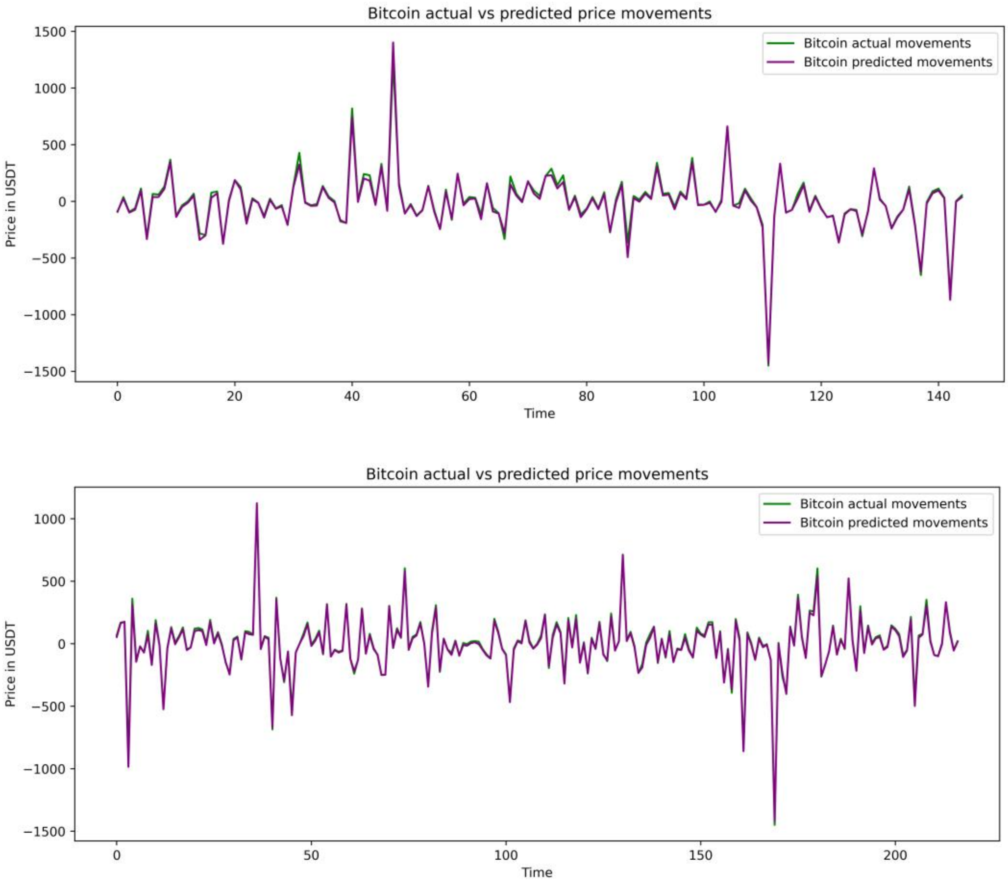

Figure 6: Actual and predicted price movements for LSTM trained on a 30-day period (above) and 45-day period (under).

On the other hand, improvement in performance can be noted in all models when trained on 45-day interval, as Table 2 clearly shows. This is visible in both Figure 5 and Figure 6, i.e., the improvement can be observed all throughout the timeline. It can be seen that predicted price movements are following the line of the actual price movements. The results in Table 2 show highest improvement in performance for the LSTM approach. Namely, from a MAE of 15.63 for the 30-day interval, the model reaches a MAE of 11.9. Moreover, the MAE of 11.9 when trained and tested on data from 45 days is the lowest for all models trained and tested in the 45-day interval. Thus, not only does the LSTM provide highest improvement, it also gives the best results overall.

The actual and predicted price movements for the LSTM for the 30-day and 45-day intervals are given in Figure 6. The figure further illustrates the significant improvement in predictions between the two approaches, showing how LSTM benefits from the added 15 days in understanding the performance better. The error which occurred around the 45th hour when applying the RF does not occur here, as LSTM has a better ability to memorize previous trends in data compared to ensemble algorithms.

The approach provides an accurate prediction model when tested on data from the selected time period. This is one of the model’s limitations, i.e., the model has only been evaluated for testing data in a close period to the data used in training. However, the time frame within which these predictions remain valid can only be determined by further exposing the model to real time data and re-observing its performance, which is intended as a future step in this research. Moreover, as Bitcoin prices have volatile market movements future research can focus on understanding the time interval of model validity before retraining is a necessity.

5 CONCLUSIONS

The study observed how different models perform when predicting cryptocurrency price movements with only six features used as input at an hour-long interval. Four different models were trained, three ensemble algorithms and one LSTM approach, and changes in model performance was observed over the course of both 30 and 45 days of data for the training and testing of the models. The results show better performance for the 45-day interval, and particular improvement can be noted with the RF and LSTM algorithms, and it was also concluded that the LSTM algorithm has significantly better performance compared to the ensemble algorithm.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] Coinmarketcap, [Online]. Available: https://coinmarketcap.com/.

[3] P. Morgen, "Reinforcing the links of Blockchain," IEEE Spectrum Magazine special edition "Blockchain World," 2017.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-80, 1997.

[5] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," Frontiers in Neurorobotics, vol. 7, 2013.

[6] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[7] G. Louppe, "Understanding Random Forests: From Theory to Practice," 2015. arXiv:1407.7502.

[8] T. Phaladisailoed and T. Numnonda, "Machine learning models comparison for bitcoin price prediction," in 2018 10th International Conference on Information Technology and Electrical Engineering, pp. 506-511, 2018.

[9] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, "Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach," Fractal and Fractional, vol. 7, no. 2, pp. 203, 2023.

[10] R. Chowdhury, M. A. Rahman, M. S. Rahman, and M. R. C. Mahdy, "An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning," Physica A: Statistical Mechanics and its Applications, vol. 551, pp. 124569, 2020.

[11] C. Betancourt and W. H. Chen, "Reinforcement learning with self-attention networks for cryptocurrency trading," Appl Sci, vol. 11, pp. 7377, 2021.

[12] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in Proceedings of 2018 Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, pp. 339-343, 2018.

[13] P. Jaquart, S. Kopke, and C. Weinhardt, "Machine learning for cryptocurrency market prediction and trading," The Journal of Finance and Data Science, vol. 8, pp. 331-352, 2022.

[14] M. Iqbal, M. S. Iqbal, et al., "Time-Series Prediction of Cryptocurrency Market using Machine Learning Techniques," Endorsed Transactions on Creative Technologies, vol. 8, no. 28, 2021.

[15] A. Dutta, S. Kumar, and M. Basu, "A gated recurrent unit approach to bitcoin price prediction," J Risk Financ Manag, vol. 13, 2020.

[16] W. Yiying, "Cryptocurrency Price Analysis With Artificial Intelligence," in 2019 5th Int. Conf. Inf. Manag., pp. 97-101, 2019.

[17] K. Rathan, S. V. Sai, and T. S. Manikanta, "Crypto-Currency price prediction using Decision Tree and Regression techniques," in 2019 3rd Int. Conf. Trends Electron. Informatics, pp. 190-194, 2019.

[18] Python-Binance API, [Online]. Available: https://python-binance.readthedocs.io/en/latest.