Investigating the Effectiveness of Machine Learning Algorithms in Predicting Bitcoin Prices and Improving Trading Strategies

Author: Sam ten Bos
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands

ABSTRACT,
This bachelor thesis investigates the effectiveness of machine learning algorithms in predicting Bitcoin prices and improving trading strategies. The investigation begins with a comprehensive review of the existing literature and approaches in this topic. Various machine learning models written in Python are used to forecast Bitcoin price movements. The models are trained and assessed using historical Bitcoin price data. Metrics such as Mean Squared Error (MSE) and R^2 score are used to evaluate the models performance. The thesis investigates the outcomes of machine learning models and provides insights into their utility in predicting Bitcoin prices. The findings demonstrate the benefits and drawbacks of various algorithms, allowing for a complete understanding of their performance in this specific context.

Graduation Committee members:

Prof. Dr. J. Osterrieder

Prof. V.B. Marisetty

Keywords
Bitcoin, Machine Learning, Python, Price Prediction

1. INTRODUCTION

In today's global investment landscape, investors are consistently seeking assets that offer the potential for high returns while minimizing portfolio risk. Over the past few years, cryptocurrency has gradually emerged as a more widely accepted and recognized investment option. This shift can be attributed to the exceptional performance demonstrated by cryptocurrencies, which has captured the attention of both institutional and individual investors. In fact, the growing popularity of cryptocurrencies has prompted several stockbrokers to introduce investment opportunities in the form of cryptocurrency Exchange Traded Funds (ETFs), with a significant focus on tracking the price movements of the renowned cryptocurrency, Bitcoin.

Bitcoin is a decentralized digital currency that works independently of government or financial institutions. It was created in 2008 by a person or group using the pseudonym Satoshi Nakamoto. The usage of cryptography ensures its security. Bitcoin transactions are recorded on a public database known as the blockchain, which allows for transparency and traceability. (Nakamoto, 2008). The decentralized nature of Bitcoin enables instant global transfers, free from the control of central banks. As a result, Bitcoin has gained popularity as both a medium of exchange and a store of value (Baur & Hoang, The Bitcoin gold correlation puzzle, 2021).

However, Bitcoin's status as a commodity is plagued by high volatility. Over a seven-year period from April 2015 to April 2022, the standard deviation of Bitcoin's daily return rate was 3.85%, significantly higher than that of gold and the S&P 500. This volatility has raised concerns about Bitcoin's function as a stable store of value and a reliable means of transaction (Baur & Dimpfl, The volatility of Bitcoin and its role as a medium of exchange and a store of value, 2021).Consequently, understanding and predicting Bitcoin's trends to minimize the associated risks have become challenging tasks.

Researchers have attempted a variety of approaches to comprehend Bitcoin's development. Some have looked into the relationship between Bitcoin's price and other commodities including gold, stock market indices, and crude oil prices. Previous research, on the other hand, has discovered very minor connections between Bitcoin and these traditional assets.. (Baur & Hoang, The Bitcoin gold correlation puzzle, 2021) (Selmi, Mensi, Hammoudeh, & Bouoiyour, 2018).

Another area of investigation is the use of artificial intelligence (AI) algorithms and powerful computing capabilities to forecast Bitcoin prices. Machine learning, a significant field in the twenty-first century, has found widespread use in a variety of fields, including stock markets, crude oil markets, gold markets, and futures markets. (Huang & Liu, 2020) (Fan, Pan, Li, & Li, 2016).

The prediction of Bitcoin prices using AI can be categorized into two main types: classification and regression. Classification research focuses on predicting whether Bitcoin prices will rise or fall, utilizing evaluation metrics such as DA (classification accuracy) and F1 (F1 score). Regression research, on the other hand, aims to predict specific Bitcoin prices, with evaluation metrics including RMSE (root mean square error) and MAPE (mean absolute percentage error). Given Bitcoin's significant price fluctuations, obtaining the specific price prediction as a reference point is more valuable than solely identifying the upward or downward movement of prices (Chen, Xie, Zhang, Bai, & Hou, 2020)

In this context, this research aims to explore the effectiveness of machine learning algorithms in predicting Bitcoin prices and improving trading strategies. By examining previous studies and their methodologies, I will assess the potential of AI-based approaches to capture Bitcoin's price trends accurately. I will also use several python models to predict bitcoin and some other cryptocurrencies.

2. LITERATURE REVIEW

Machine learning techniques, which incorporate artificial intelligence systems, attempt to extract patterns learned from previous data - a process known as training or learning - in order to later generate predictions about new data. (Xiao, Xiao, Lu, & Wang, 2014). Machine learning-based prediction algorithms have found widespread application in medical, financial, and other fields. Human or artificial intelligence could make purchase and sell trading choices on the financial market. The use of machines to make trading choices on the FX and stock markets has grown quickly. (Meng & Khushi , 2019). Stock market forecasting is one of the most essential and difficult tasks requiring time series. (Chen, Xiao, Sun, & Wu, 2017) The prediction of price time series in financial markets, which have a non-stationary nature, is very difficult (Zhang, Lin, & Shang, 2017).

The basic techniques used in the literature include the following: artificial neural networks (ANNs), support vector machines (SVMs), and random forests (RFs). An artificial neural network (NN) is a computational structure modelled loosely on biological processes. ANNs explore many competing hypotheses simultaneously using a massively parallel network composed of non-linear relatively computational elements interconnected by links with variable weights. It is this interconnected set of weights that contains the knowledge generated by the Neural Network. (Henrique, Sobreiro, & Kimura, 2019)

In general, NN models are defined by network topology, node attributes, and training or learning rules. NNs are made up of a huge number of basic processing units that interact with one another via excitatory or inhibitory connections. (Ayda & Collopy, 1998). Whereas neural networks seek to minimize the errors of their empirical responses in the training stage, an SVM seeks to minimize the upper threshold of the error of its classifications. (Yang, et al., 2020)

Support Vector Machine (SVM) has important applications in function regression estimation, prediction time sequence, stock trend and pattern classification and recognition (Intezari & Gressel, 2017) .Another key property of SVM is that training SVM is equivalent to solving a linearly constrained quadratic programming problem so that the solution of SVM is always unique and globally optimal, unlike neural networks training which requires nonlinear optimization with the danger of getting stuck at local minima. (Huang, Nakamori, & Wang, 2005)

Random forests are an ensemble of tree predictors, where each tree relies on the values of a random vector that is independently and identically distributed across all trees in the forest. As the number of trees in the forest increases, the generalization error of the forest converges almost surely to a limit. The generalization error of a random forest, consisting of multiple tree classifiers, is influenced by the strength of the individual trees and the correlation between them. (Breiman, 2001)

A decision tree is a hierarchical structure that can be binary or non-binary in nature. The tree's non-leaf nodes represent feature testing, with each branch corresponding to a specified range of attribute values for that feature. In contrast, leaf nodes store the given category or class. The decision tree starts with a root node and evaluates the feature properties relevant to the classification category. The tree then branches according to the attribute values

until it reaches a leaf node. The ultimate judgment or classification result is the category recorded in the leaf node. (Zhu, Qui, Ergu, Ying, & Liu, 2019)

Prediction of mature financial markets such as the stock market has been researched at length, Bitcoin presents an interesting parallel to this as it is a time series prediction problem in a market still in its transient stage. (McNally, Roche, & Caton, 2018)

Because cryptocurrency prices are nonlinear and nonstationary, data distribution assumptions are ineffective for forecasting. Machine learning techniques exploit the data's inherent non-linear and non-stationary qualities, while also accounting for explanatory features and underlying factors. Numerous studies on the modeling and forecasting of Bitcoin values using machine learning have been undertaken. (Mudassir, Bennbaia, Unal, & Hammoudeh, 2020)

There has been little research into predicting the price of Bitcoin using machine learning algorithms. Several research have looked into various methodologies and data sources for predicting Bitcoin prices. Sentiment analysis utilizing support vector machines in conjunction with the frequency of Wikipedia views and the network hash rate, for example, was examined by (Georgula, Pournarakis, Bilanakos, Sotiropoulos, & Giaglis, 2015) Additionally, the relationship between Bitcoin price, tweets, and views on Google Trends was analysed by (Matta, Lunesu, & Marchesi, Bitcoin Spread Prediction Using Social And Web, 2015) Another study implemented a similar methodology but focused on predicting trading volume using Google Trends views (Matta, Lunesu, & Marchesi, The predictor impact of Web search media on Bitcoin trading volumes, 2015). It's important to note that these studies often face limitations, including small sample sizes and the potential spread of misinformation through social media platforms like Twitter or Reddit, which can artificially inflate or deflate prices.

In analysing the Bitcoin Blockchain (Greaves & Au, 2015) used support vector machines (SVM) and artificial neural networks (ANN) to predict the price of Bitcoin. They reported a price direction accuracy of 55% with a regular ANN and concluded that there was limited predictability in Blockchain data alone. Another study by (Madan, Saluja, & Zhao, 2015)also utilized Blockchain data and applied SVM, Random Forests, and Binomial GLM, achieving prediction accuracy of over 97%. However, the lack of cross-validation in their models limits the generalizability of their results.

Researchers have also investigated the relationship between various factors and price changes in order to forecast Bitcoin prices. For example, research have been conducted to investigate the relationship between search engine views, network hash rate, mining difficulty, and Bitcoin price. (Kristoufek , 2015) In an attempt to build on these findings, analyses have incorporated data from the Blockchain, including hash rate and difficulty, as well as data from major exchanges provided by CoinDesk. (Delfin-Vidal & Romero-Melendez, 2016)

Bitcoin price prediction is comparable to other financial time series prediction problems, such as currency and stock prediction. Some research have used the Multilayer Perspective (MLP) to forecast stock prices. (Department of Economics, University of California,, 1988) However, a more effective approach involves recurrent neural networks (RNN), which can capture the temporal relationship of the series by retaining outputs from each layer in a context layer (Gilles, Lawrence, & Tsoi, 2001)Specifically, Long Short-Term Memory (LSTM) networks, a type of RNN, have shown promising results due to their ability to selectively remember or forget data based on importance. (Gers, Eck, & Schmidhuber, 2001)

Throughout the debates on various financial applications, machine learning models outperformed statistical models. Nonetheless, a comparison study found that statistical models like linear regression and LDA beat machine learning models in predicting daily bitcoin prices. Meanwhile, in 5-minute bitcoin price forecast, machine learning models outperform statistical models. (Chen, Li, & Sun, 2020).

The evaluation criteria show that SDAE has the best predictive capacity for forecasting bitcoin price. In another study, SVM, ANN with single and double hidden layers, and ensemble models (based on RNN and k-Means clustering) predicted the maximum, minimum, and closing bitcoin values. While both models performed very well, using the regression results as inputs to forecast the direction of bitcoin prices enhanced accuracy by 10%. (Mallqui & Fernandes, 2019)

(Fernández-Delgado, 2014)evaluate 179 classifiers, arising from 17 families, resulting in Random Forests achieving the highest accuracy. (Mayo and Elgazzar, 2022) compare several methods and conclude that Random Forests achieves the best performance, possibly together with Neural Networks.

(Francisco Orte, 2023)focused on predicting the direction of BTC/USD price utilizing Random Forest as the prediction model and technical indicators and candlestick patterns as input variables. The study calculated the best number of estimators for the model and examined various candle intervals, horizons, and features combinations.

According to the statistics, combining a 1-day timeframe with candlestick patterns as characteristics yielded the greatest results. The ideal horizon, however, could not be defined conclusively. The researchers also conducted an out-of-sample forecast for the first five days of 2021, which differed from the testing stage but confirmed the mode's utility.

In conclusion, while research on predicting Bitcoin prices with machine learning algorithms is limited, various approaches have been explored, including sentiment analysis, Blockchain analysis, and the consideration of external factors. Understanding the limitations and leveraging advanced models like RNN and LSTM, along with GPU acceleration, can contribute to improved predictions in this domain.

3. METHODOLOGY

Historical price data for various cryptocurrencies was retrieved from trusted sources such as “Coin Gecko” and “Yahoo Finance” during the data retrieval process. The information includes opening and closing prices, high and low prices, volume, and market capitalization. This information was obtained using Python APIs and packages such as requests and yfinance. The retrieved data served as the foundation for additional analysis and modeling to evaluate the efficiency of machine learning algorithms in predicting Bitcoin values and optimizing trading methods.

3.1 Research Design

In order to measure whether Machine Learning could be effective in prediction bitcoin we used different models and test which one would work best to predict bitcoin. We conducted a similar kind of research as (Jaquart, Dann, & Weinhardt, 2021) We first had to install python and Visual Studio Code to make the codes run. The first model we ran was a Random forest model. After this we extended this code and ran some more complicated models such

as Gradient Boosting, Additional Time Series and Cross Validation

3.2 Data retrieving

For the data retrieving I used several websites such as Coingecko API and Yahoo finance. I also used ChatGPT to help us build these codes.

3.3 Models

In this section of the thesis I describe which different Python models I used to predict the bitcoin returns.

Random forest

As said we first started with some random forest models. In short, Random Forest is a machine learning algorithm that combines multiple decision trees to make predictions. It randomly selects subsets of data and features to build individual decision trees. The final prediction is obtained by aggregating the predictions from all the trees. Random Forest is known for its robustness, ability to handle complex datasets, and avoidance of overfitting. It is widely used in various domains for classification and regression tasks. we used various coins to predict the price of. Also we tested the values we got by buying crypto in real life and tracking if the values where right. (IBM, sd)

The first code we used retrieved historical price data for the 10 most used cryptocurrencies using the CoinGecko API. It pre-processes the data by creating a target column shifted by 12 hours to represent the future price. The data is split into training and test sets, and a random forest regressor model is trained. Finally, the model predicts the price increase for the last 12 hours, and the Mean Absolute Percentage Error (MAPE) is calculated.

The second code predicted which of the 10 most traded crypto's would give me the most profit in 7 days. (see results)

Gradient Boosting

With this model we used Gradient boosting to predict bitcoin price. Gradient boosting is an ensemble machine learning technique that combines multiple weak models, usually decision trees, to create a powerful predictive model. It uses an iterative process that focuses on correcting the errors made by the previous models. By optimizing a loss function through gradient descent, the models in the ensemble are weighted and combined to make accurate predictions. (Hoare, sd) (Appendix 1)

Additional Time series input

This model is a Python function that downloads Bitcoin price data, incorporates additional input time series data, and uses the Random Forest algorithm to predict Bitcoin prices. In this model, 'volume' and 'market cap' are used as additional features, and their corresponding values are provided in the additional data dictionary. (Wikipedia, sd) (Appendix 2)

Cross validation

With this model we demonstrates the usage of Random Forest regression for predicting Bitcoin prices, performs cross-validation to assess the model's performance, and visualizes the predicted prices alongside the actual prices. It first downloads the Bitcoin price data, pre-processes it, and initializes a Random Forest regressor model. Then, cross-validation is performed using the cross_val_score function, which calculates the R^2 scores for five different cross-validation folds. The cross-validated scores are printed to evaluate the model's performance. Finally, the model is trained on the entire dataset, and the predict_crypto_price function is called to generate predictions and plot the results for Bitcoin. (Wikipedia, sd)(Appendix 3)

4. RESULTS

This section gives an overview and analysis of the findings from my research and experiments. It summarizes the findings of my research, such as the performance of machine learning models in predicting Bitcoin values. Statistical indicators such as mean squared error (MSE), R-squared (R2) scores, and other evaluation metrics are included in this section. I also give some visual representations of the results, such as tables and histograms.

4.1 Retrieved values

In the retrieved values section we distinguished between simple and the more complicated models. We started with some simpler models to see what this did. Even though there were some positive tests, I couldn't consider them valid as there were far too few factors included in the calculations.

4.1.1 Simple Models

Coin	Price Increase	MAPE
Bitcoin	-1.29%	0.01%
Ethereum	-1.30%	0.01%
Ripple	0.51%	0.01%
Binancecoin	0.84%	0.01%
Cardano	1.19%	0.01%
Dogecoin	-0.43%	0.01%
Polkadot	-0.89%	0.01%
Litecoin	-3.39%	0.02%
Bitcoin-Cash	-2.02%	0.01%
Chainlink	-1.85%	0.02%

In the table above we see that Cardano was expected to grow the most in the upcoming 12 hours. So we focused on this, bought cardano and tracked the prices. We see that at 21:46, Cardano was worth 0,33666. At 09:45, so 12 hours later Cardano is worth 0,33914. This is an increase of 0,74%. So not the 1,19% the model predicted. But if we look at the graph, we can see that in between the last 12 hours there was a high peak. At around 04:20, Cardano was 0,34607. If we calculate this increase we see an 2,8% increase. This is even more than the model predicted.

Above we used a model which predicted which of the 10 most traded crypto's would give me the most profit in 7 days. The best cryptocurrency to buy for the next 7 days is: Litecoin Expected profit percentage: 16.72%. To test this I bought $54,27 worth of crypto at 16-05-2023. As we can see the highest price Litecoin had was right the day after we bought it, 56,67.(Figure) After this the price varied a bit between these numbers. Now we want to know how much profit we could have made. 56,67-54,27/54,27 * 100% = 4,42%. This is nowhere near the 16,72% the model predicted.

4.1.2 Complicated models

Model	MSE	R^2
Gradient Boosting	0.024	-0.097
Additional Time Series	0.0078	0.828
Cross Validation	0.00098	0.932

Here above we made a table of the values we got from our models. To clarify, the R2 score measures the goodness of fit of a model, while the MAE quantifies the average absolute difference between predicted and actual values.

We also let the models predict the returns for the upcoming 7 days. These are the values we got for:

GB = Gradient Boosting

ATS = Additional Time Series

CV = Cross Validation

Day	GB	ATS	CV
1	0.35%	1.57%	-0.41%
2	0.35%	2.09%	0.79%
3	0.38%	0.19%	-1.57%
4	0.21%	-0.62%	0.24%
5	-0.65%	0.96%	0.54%
6	0.28%	3.24%	-1.12%
7	0.52%	0.92%	0.15%

Gradient Boosting

Looking at the values we got from our model we observe the following things.

1. Mean Absolute Error (MAE): A MAE of around 0.024 means that the model's forecasts for Bitcoin returns depart from the actual returns by approximately 0.024 on average. This shows that the model's accuracy in predicting Bitcoin returns is not particularly good.
2. R2 Score: A negative R2 score of around -0.097 implies that the model performs worse than a horizontal line (the dependent variable's mean) in explaining the variance in Bitcoin returns. This indicates that the model is not properly capturing the underlying patterns and relationships in the data.

Seeing these values we cannot assume that the predicted returns are very reliable.

Additional Time series input

Looking at the values we retrieved from our model we observe the following things.

1. Mean Absolute Error (MAE): A MAE of 0.007755 shows that the model's forecasts depart from real Bitcoin returns by around 0.007755 on average. Because lower MAE values indicate more accuracy, this figure indicates that the model's predictions are reasonably close to the actual returns.
2. The R2 score of 0.828035 implies that the model accounts for approximately 82.8% of the variance in Bitcoin returns. R2 scores range from 0 to 1, with 1 representing a perfect fit. The R2 value in this situation indicates that the model captures a significant amount of the underlying patterns and trends in the data.

The expected returns for the next 7 days indicate a mix of positive and negative returns, reflecting potential Bitcoin market swings.

Cross validation

Looking at the values we retrieved from our model we observe the following things.

1. The MSE on the Entire Dataset is 9.084068292528452e-05, which is less than the average MSE from cross-validation. This means that the model performs better on training data than on unknown data.
2. R2 Score for the Entire Dataset: The overall R2 score is 0.9366279645259508. This means that the model accounts for approximately 93.7% of the volatility in Bitcoin returns. A high R2 value over the entire dataset suggests a good match.

Overall, while the MSE and R2 scores on the entire dataset indicate that the model has some predictive ability, the negative R2 scores from cross-validation and the relatively low average R2 score indicate that the model may not effectively capture the underlying patterns and variability in Bitcoin returns.

4.2 Conclusion of the results

In this section of the thesis, I compare the results and indicate which model performs the best. Model 3 (Random Forest with time series cross-validation) appears to be the best of the three based on these criteria. This is why:

1. Mean Squared Error (MSE): During cross-validation, Model 3 has quite low MSE values, indicating higher predictive ability than Model 1.
2. R2 Score: Despite having negative R2 ratings during cross-validation, Model 3 has the greatest average R2 score among the three models. It implies that Model 3 explains the most variance in Bitcoin returns, despite the fact that individual cross-validation folds may not capture the variability properly.
3. Model 3 outperforms the other models on the complete dataset, with a low MSE and a high R2 score, showing superior predicting ability on unknown data.
4. Model 3 includes time series cross-validation, which is useful for evaluating and training models on time-dependent data such as bitcoin prices. This method captures temporal relationships and gives a more legitimate analysis of the model's performance.

Given these criteria, Model 3 (Random Forest with time series cross-validation) outperforms the other two models in terms of overall performance and appears to be the best option.

5. DISCUSSION

This study has aimed to provide a comprehensive examination of the Effectiveness of Machine Learning Algorithms in Predicting Bitcoin Prices and Improving Trading Strategies. However, there are several limitations that should be considered when interpreting the results of this study.

To begin, the usage of three specific machine learning methods (Gradient Boosting, Cross Validation, and Additional Time

Series) creates the possibility of model selection bias. By selecting these specific models, I risk overlooking other models or algorithms that could potentially deliver greater prediction performance or be more appropriate for the specific properties of Bitcoin price data.

Also the availability and quality of data is a limitation. This can have a major impact on model performance and generalizability. Bitcoin price data can be sensitive to market manipulation, data gaps, and variations among exchanges. Furthermore, the models' capacity to capture long-term patterns or adjust to changing market conditions may be impacted by the insufficient historical data for Bitcoin.

Another limitation is overfitting. Machine learning models, especially those with high complexity or flexibility, can be prone to overfitting the training data. Overfitting is an undesirable machine learning behavior that occurs when the machine learning model gives accurate predictions for training data but not for new data. When data scientists use machine learning models for making predictions, they first train the model on a known data set. Then, based on this information, the model tries to predict outcomes for new data sets. An overfit model can give inaccurate predictions and cannot perform well for all types of new data (What is overfitting?, sd)

Furthermore, the underlying technology of Bitcoin, blockchain, is continually changing and being upgraded. New protocols, consensus processes, or scaling solutions can have a substantial impact on the Bitcoin network's dynamics and behavior. These modifications may create uncertainty and have an impact on the predicted performance of models trained on past data (Hayes, 2023).

In summary, the findings of this study provide substantial insight into the effectiveness of machine learning algorithms in predicting bitcoin prices. However, the limitations of this study should be considered when interpreting the findings. In order to provide a more full understanding of the problem, future study should address these shortcomings.

6. CONCLUSION

In conclusion, this thesis examined the effectiveness of machine learning algorithms in predicting Bitcoin values and enhancing trading techniques. The findings of this study show that machine learning algorithms have a lot of potential for detecting price trends and patterns in the turbulent cryptocurrency market. Among the models tested, Random forest with cross-validation had the highest prediction accuracy and stability. However, it is crucial to note that the effectiveness of these models can be modified by factors such as data availability, feature selection, and Bitcoin market specifics. To improve prediction accuracy, model selection and feature engineering strategies should be carefully examined

This study could provide the basis for future research in the field of machine learning for Bitcoin price prediction and trading technique enhancement. There is room for improvement by combining more data sources, for example social media state or macroeconomic indicators.

7. BIBLIOGRAPHY

Ayda, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting*, 347-495.

Baur, D. G., & Dimpfl, T. (2021). The volatility of Bitcoin and its role as a medium of exchange and a store of value. *Empirical Economics*, 2663–2683. Retrieved from https://link.springer.com/article/10.1007/s00181-020-01990-5?module=inline&pgtype=article

Baur, D. G., & Hoang, L. (2021). The Bitcoin gold correlation puzzle. *Journal of Behavioral and Experimental Finance*. Retrieved from https://www.sciencedirect.com/science/article/pii/S2214635021001052

Breiman, L. (2001). Random Forests. *Machine Learning 45*, 5-32.

Chen, H., Xiao, K., Sun, J., & Wu, S. (2017). A Double-Layer Neural Network Framework for High-Frequency Forecasting. *ACM Transactions on Management Information Systems*, 1-17.

Chen, Y., Xie, X., Zhang, T., Bai, J., & Hou, M. (2020). A deep residual compensation extreme learning machine and applications. *Journal of Forecasting*, 986-999. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/for.2663

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*.

Delfin-Vidal, R., & Romero-Melendez, G. (2016). The Fractal Nature of Bitcoin: Evidence from Wavelet Power Spectra. *Trends in Mathematical Economics*, 73-98. Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-32543-9_5

Department of Economics, University of California,. (1988). Economic prediction using neural networks: the case of IBM daily stock returns. *IEEE 1988 International Conference on Neural Networks.* San Diego: IEEE. Retrieved from

https://ieeexplore.ieee.org/abstract/document/23959/authors#authors

Fan, L., Pan, S., Li, Z., & Li, H. (2016). An ICA-based support vector regression scheme for forecasting crude oil prices. *Technological Forecasting and Social Change*, 245-253. Retrieved from https://www.sciencedirect.com/science/article/pii/S0040162516300579

Fernández-Delgado, M. C. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 3133-3181.

Francisco Orte, J. M. (2023). A random forest-based model for crypto asset forecasts in futures markets with out-of-sample prediction. *Research in International Business and Finance*.

Georgula, I., Pournarakis, D., Bilanakos, C., Sotiropoulos, D., & Giaglis, G. M. (2015). *Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices.* Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2607167

Gers, F. A., Eck, D., & Schmidhuber, J. (2001). Applying LSTM to Time Series Predictable through Time-Window Approaches. *Artificial Neural Networks — ICANN 2001*, 669-676. Retrieved from https://link.springer.com/chapter/10.1007/3-540-44668-0_93

Gilles, C. L., Lawrence, S., & Tsoi, A. C. (2001). Noisy Time Series Prediction using Recurrent. *Machine Learning 44*, 161-183. Retrieved from https://clgiles.ist.psu.edu/papers/MLJ-2001-finance-time-series.pdf

Greaves, A., & Au, B. (2015). *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin.* Retrieved from http://snap.stanford.edu/class/cs224w-2015/projects_2015/Using_the_Bitcoin_Transaction_Graph_to_Predict_the_Price_of_Bitcoin.pdf

Hayes, A. (2023, April 23). *Learn what these digital public ledgers are capable of*. Retrieved from investopedia: https://www.investopedia.com/terms/b/blockchain.asp

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 226-251. Retrieved from https://www.sciencedirect.com/science/article/pii/S095741741930017X?fr=RR-2&ref=pdf_download&rr=7d75041f2c03b891

Hoare, J. (n.d.). *Gradient Boosting Explained – The Coolest Kid on The Machine Learning Block*. Retrieved from Displayr: https://www.displayr.com/gradient-boosting-the-coolest-kid-on-the-machine-learning-block/#:~:text=Gradient%20boosting%20is%20a%20type,order%20to%20minimize%20the%20error.

Huang, J.-Y., & Liu, J.-H. (2020). Using social media mining technology to improve stock price forecast accuracy. *Journal of Forecasting*, 104-116. Retrieved from https://onlinelibrary.wiley.com/doi/10.1002/for.2616

Huang, W., Nakamori, Y., & Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 2513-2522.

IBM. (n.d.). *What is random forest?* Retrieved from IBM: https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.

Intezari, A., & Gressel, S. (2017). Information and reformation in KM systems: big data and

strategic decision-making. *Journal of Knowledge Management*, 71-91. Retrieved from https://www.scopus.com/record/display.uri?eid=2-s2.0-85014758737&origin=inward&txGid=2174990898c6e9707cf5ca662fa2d8a2

Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *The Journal of Finance and Data Science*, 45-66.

Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M.-C., & Johnson, J. (2020). Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting. *European Journal of Operational Research*, 2017-234. Retrieved from https://www.sciencedirect.com/science/article/pii/S0377221719309099

Kim, J.-M., Kim, S.-T., & Kim, S. (2020). On the Relationship of Cryptocurrency Price with US Stock and Gold Price Using Copula Models. *Mathematics*. Retrieved from https://www.mdpi.com/2227-7390/8/11/1859

Kristoufek , L. (2015). *What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis.* PLOS ONE. Retrieved from https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0123923

Madan, I., Saluja, S., & Zhao, A. (2015). *Automated Bitcoin Trading via Machine Learning Algorithms.* Stanford University. Retrieved from https://www.smallake.kr/wp-content/uploads/2017/10/Isaac-Madan-Shaurya-Saluja-Aojia-ZhaoAutomated-Bitcoin-Trading-via-Machine-Learning-Algorithms.pdf

Mallqui, D. C., & Fernandes, R. A. (2019). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, 596-606.

Matta, M., Lunesu, I., & Marchesi, M. (2015). *Bitcoin Spread Prediction Using Social And Web.* Cagliary: Università degli Studi di Cagliari. Retrieved from https://d1wqtxts1xzle7.cloudfront.net/40425066/Bitcoin_Spread_Prediction_Using_Social_A20151127-10804-12xrxm4-libre.pdf?1448629282=&response-content-disposition=inline%3B+filename%3DBitcoin_Spread_Prediction_Using_Social_A.pdf&Expires=1686951171&Signature

Matta, M., Lunesu, I., & Marchesi, M. (2015). The predictor impact of Web search media on Bitcoin trading volumes. *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K).* Lisbon: IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/7526987?casa_token=SW9z6LhSuroAAAAA:mFhYZyuxnYu0XKEwC7lS2CxNXu2guS_FZ-TzepFZ-ihstNCTTfVhFlAx5lAQnyi50mdZVQpI

McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP).* Cambridge: IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/8374483

Meng, T. L., & Khushi , M. (2019). *Reinforcement Learning in Financial Markets.* Darlington: School of Computer Sciences. Retrieved from https://www.mdpi.com/2306-5729/4/3/110

Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2020). Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*. Retrieved from https://link.springer.com/article/10.1007/s00521-020-05129-6#Abs1

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* bitcoin.org. Retrieved from https://bitcoin.org/bitcoin.pdf

R, B. (2019). *An Econometric Analysis of the Relationship between Bitcoin & Gold.* Retrieved from https://medium.com/@blake_richardson/an-econometric-analysis-of-the-relationship-between-bitcoin-gold-2018-584b4c63a17

Selmi, R., Mensi, W., Hammoudeh, S., & Bouoiyour, J. (2018). Is Bitcoin a hedge, a safe haven or a diversifier for oil price movements? A comparison with gold. *Energy Economics*, 787-801. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0140988318302524

What is overfitting? (n.d.). Retrieved from AWS: https://aws.amazon.com/what-is/overfitting/#:~:text=Overfitting%20is%20an%20undesirable%20machine,on%20a%20known%20data%20set.

Wikipedia. (n.d.). *Cross-validation (statistics)*. Retrieved from Wikipedia The Free Encyclopedia: https://en.wikipedia.org/wiki/Cross-validation_(statistics)#:~:text=Cross%2Dvalidation%20is%20a%20resampling,model%20will%20perform%20in%20practice.

Wikipedia. (n.d.). *Time series*. Retrieved from Wikipedia The Free Encyclpedia: https://en.wikipedia.org/wiki/Time_series

Xiao, Y., Xiao, J., Lu, F., & Wang, S. (2014). Ensemble ANNs-PSO-GA Approach for Day-ahead Stock E-exchange Prices Forecasting. *International Journal of Computational Intelligence Systems* , 272-290.

Yang, R., Yu, L., Zhao, Y., Yu, H., Xu, G., Wu, Y., & Liu, Z. (2020). Big data analytics for financial Market volatility forecast based on support vector machine. *International Journal of Information Management*, 452-462. Retrieved from https://www.sciencedirect.com/science/article/pii/S0268401218313604

Zhang, N., Lin, A., & Shang, P. (2017). Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting. *Physica A: Statistical Mechanics and its Applications*, 161-173.

Zhu, L., Qui, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science*, 503-513. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050919320277

8. APPENDIX

Appendix 1

```
Gradient Boosting

import pandas as pd
import yfinance as yf
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_absolute_error
import matplotlib.pyplot as plt

def download_bitcoin_data():
    # Define the desired date range for historical data
    start_date = '2018-01-01'
    end_date = '2022-12-31'

    # Download Bitcoin price data from Yahoo Finance
    bitcoin_data = yf.download('BTC-USD', start=start_date, end=end_date)

    # Return the Bitcoin price data
    return bitcoin_data

def preprocess_data(df):
      df = df['Close'].pct_change().dropna()  # Calculate the daily
percentage change in price
    return df

def train_gradient_boost(X_train, y_train):
    # Create and train the Gradient Boosting model
    gb_model = GradientBoostingRegressor(random_state=42)
      gb_model.fit(X_train, y_train.ravel())  # Reshape y_train to
(n_samples, )
    return gb_model

def evaluate_model(model, X_test, y_test):
    # Make predictions
    y_pred = model.predict(X_test)

    # Calculate MAE
    mae = mean_absolute_error(y_test, y_pred)

    return mae

def predict_bitcoin_returns():
    # Download Bitcoin price data
    bitcoin_data = download_bitcoin_data()
```

```
    # Preprocess the data
    preprocessed_data = preprocess_data(bitcoin_data)

    # Split the data into features (X) and target (y)
    X = preprocessed_data.iloc[:-7].values.reshape(-1, 1)
    y = preprocessed_data.shift(-7).dropna().values.reshape(-1, 1)

    # Split the data into training and testing sets
    train_size = int(0.8 * len(X))
    X_train, X_test = X[:train_size], X[train_size:]
    y_train, y_test = y[:train_size], y[train_size:]

    # Train the Gradient Boosting model
    gb_model = train_gradient_boost(X_train, y_train)

    # Evaluate the model
    mae = evaluate_model(gb_model, X_test, y_test)
    print("Mean Absolute Error:", mae)

    # Plot the actual and predicted returns
     plt.plot(bitcoin_data.index[train_size+7:train_size+7+len(y_test)],
y_test, label='Actual')
     plt.plot(bitcoin_data.index[train_size+7:train_size+7+len(y_test)],
gb_model.predict(X_test), label='Predicted')
    plt.xlabel('Date')
    plt.ylabel('Bitcoin Returns')
    plt.title('Bitcoin Return Prediction')
    plt.legend()
    plt.show()

    # Example: Predict the Bitcoin return for the next 7 days
    last_7_days = preprocessed_data[-7:].values.reshape(-1, 1)
    predicted_returns = gb_model.predict(last_7_days)
    print("Predicted Bitcoin Returns for the next 7 days:")
    for i in range(len(predicted_returns)):
        print(f"Day {i+1}: {predicted_returns[i]}")

# Call the function to predict Bitcoin returns
predict_bitcoin_returns()
```

Appendix 2

```
import requests
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, r2_score
```

```
def download_crypto_data(symbol):
    # Download cryptocurrency price data from CoinGecko API
    url =
f'https://api.coingecko.com/api/v3/coins/{symbol}/market_chart?vs_curre
ncy=usd&days=365'
    response = requests.get(url)
    json_data = response.json()
    df = pd.DataFrame(json_data['prices'], columns=['timestamp',
'price'])
    df['timestamp'] = pd.to_datetime(df['timestamp'], unit='ms')
    df.set_index('timestamp', inplace=True)

    # Calculate returns
    df['return'] = df['price'].pct_change()
    df.dropna(inplace=True)

    return df

def preprocess_data(df, additional_data):
    df = df.resample('D').last().ffill()
    for i in range(1, 8):
        df[f'lag_{i}'] = df['return'].shift(i)
    additional_df = pd.DataFrame(additional_data)
    additional_df = additional_df.reindex(df.index, fill_value=0)  #
Align additional data with Bitcoin return data
    df = pd.concat([df, additional_df], axis=1)
    df = df.dropna()
    X = df.drop('return', axis=1)
    y = df['return']
    return X, y

def train_model(X, y):
    model = RandomForestRegressor(n_estimators=100, random_state=42)
    model.fit(X, y)
    return model

def evaluate_model(model, X, y):
    y_pred = model.predict(X)
    mae = mean_absolute_error(y, y_pred)
    r2 = r2_score(y, y_pred)
    return mae, r2

def plot_predictions(symbol, model, X, y):
    y_pred = model.predict(X)
    plt.plot(y.index, y.values, label='Actual')
    plt.plot(y.index, y_pred, label='Predicted')
    plt.xlabel('Date')
```

```
    plt.ylabel('Return')
    plt.title(f'Actual vs Predicted {symbol} Returns')
    plt.legend()
    plt.show()

def predict_crypto_returns(symbol, additional_data):
    df = download_crypto_data(symbol)
    X, y = preprocess_data(df, additional_data)
    model = train_model(X, y)
    mae, r2 = evaluate_model(model, X, y)
    print("Mean Absolute Error:", mae)
    print("R2 Score:", r2)
    plot_predictions(symbol, model, X, y)
    return X, model

# Additional time series data for prediction
additional_data = {
    'volume': [1000] * 365,  # Example: Set the same value for all days
    'market_cap': [5000] * 365  # Example: Set the same value for all
days
}

# Predict Bitcoin returns using additional input time series
symbol = 'bitcoin'
X, model = predict_crypto_returns(symbol, additional_data)

# Example: Predict Bitcoin returns for the next 7 days
last_7_days = X.iloc[-7:, :]
predicted_returns = model.predict(last_7_days)
print("Predicted Bitcoin Returns for the next 7 days:")
for i, return_value in enumerate(predicted_returns):
    print(f"Day {i+1}: {return_value}")
```

Appendix 3

```
import yfinance as yf
import numpy as np
import pandas as pd
from sklearn.model_selection import TimeSeriesSplit, cross_val_score,
train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Download Bitcoin price data using yfinance
btc_data = yf.download('BTC-USD', start='2010-01-01', end='2023-05-23')
```

```
# Prepare the data
btc_data = btc_data.reset_index()
btc_data['Returns'] = btc_data['Close'].pct_change()
btc_data = btc_data.dropna()  # Remove rows with missing values
X = btc_data[['Open', 'High', 'Low', 'Close', 'Volume']]
y = btc_data['Returns']

# Initialize time series cross-validator
tscv = TimeSeriesSplit(n_splits=5)

# Create a Random Forest regressor model
model = RandomForestRegressor()

# Perform cross-validation and evaluate the model's performance
mse_scores = -cross_val_score(model, X, y, cv=tscv,
scoring='neg_mean_squared_error')
r2_scores = cross_val_score(model, X, y, cv=tscv, scoring='r2')

# Print the mean squared error scores and the average score
print("Mean Squared Error scores:", mse_scores)
print("Average MSE:", np.mean(mse_scores))

# Print the R2 scores and the average score
print("R2 scores:", r2_scores)
print("Average R2:", np.mean(r2_scores))

# Train the model using the entire dataset
model.fit(X, y)

# Make predictions for the next 7 days
last_7_days = X.tail(7)
next_7_days_predictions = model.predict(last_7_days)

# Calculate the MSE on the entire dataset
predictions = model.predict(X)
mse = mean_squared_error(y, predictions)
print("MSE on the entire dataset:", mse)

# Calculate the R2 score on the entire dataset
r2 = r2_score(y, predictions)
print("R2 score on the entire dataset:", r2)

# Plot the actual Bitcoin returns and the predictions
plt.figure(figsize=(10, 6))
plt.plot(btc_data['Date'], y, label='Actual')
plt.plot(btc_data['Date'], predictions, label='Predicted')
plt.xlabel('Date')
```

```
plt.ylabel('Bitcoin Returns')
plt.title('Actual vs. Predicted Bitcoin Returns')
plt.legend()
plt.xticks(rotation=45)
plt.show()

# Print the predictions for the next 7 days
print("Predictions for the next 7 days:")
print(next_7_days_predictions)
```