Analyzing the behavior of cryptocurrency market participants using machine learning

Reza Fakhre Hosseini, Sina Sepahvand*, Shahriar Gharibzadeh

Abstract

This article presents an experimental analysis of the 5-year historical data of seven large-cap cryptocurrencies to predict price direction using machine learning algorithms. The study aims to answer the question of whether price or return is a better input for algorithms, given that cryptocurrency traders typically use returns to evaluate their assets. The focus of this paper is on the behavior of participants rather than forecasting future price movements.

The results indicate that machine learning algorithms are more accurate when using return as input, contrary to expectations. Furthermore, the study investigates the effect of adding indicators and independent variables, revealing that adding indicators improves model accuracy and AUC but adding more independent variables does not have a significant impact. While similar methods have been applied to the stock market before, the results of this study provide unique insights that can enhance our understanding of high-risk markets such as cryptocurrency.

Keywords: deep learning; finance; risk; decision making; cryptocurrency

Introduction

In recent years, the cryptocurrency market has emerged as a significant area of interest for researchers in the field of economics and finance. The market has a complex network structure, which can be an attractive topic for researchers studying network science. The application of network science to the analysis of cryptocurrency markets has been the focus of several research studies. For example, Reid and Harrigan (2013) analyzed the Bitcoin network's topology, while Ober et al. (2013) studied the Bitcoin mining pool network. Similarly, Kondor et al. (2014) used the methods of statistical physics to analyze the structure of the Bitcoin market. Alvarez Pereira et al. (2014) analyzed the correlation structure of cryptocurrencies, while Maesa et al. (2016) studied the market efficiency of Bitcoin and there were also Ron and Shamir (2013), Baumann et al. (2014), Fleder et al. (2015), Lischke and Fabian (2016), Akcora et al. (2017), Maesa et al. (2017), Ranshous et al. (2017) and references therein.

Despite the market's massive $1.1 trillion market cap, cryptocurrencies still seem to have a long way to go to find their substantive place among the financial market participants. One of the reasons for this is the behavior and attitude of traders towards the assets, which has caused the emergence of unusual market values. However, only a small amount of research has been done about how the participants behave to determine the market values of assets.

Machine learning methods have been increasingly used to investigate the behavior of market participants. In the article by Kamalov and Gurrib (2021), machine learning methods were applied to investigate the behavior of stock market participants using price or return as input. They tried to find which one of the prices or returns is a more appropriate input for financial forecasting in

the stock market. The authors used machine learning methods based on recurrent neural network models (such as long short-term memory (LSTM)) to predict the stock price movements. The results of their study have considerable analytical value, as similar methods have been used in other studies on financial forecasting (Eur J Oper Res, 2018).

However, it is not clear whether the same methods would be effective in predicting the behavior of cryptocurrency markets, given their unique characteristics. Therefore, in this study, we apply similar machine learning methods to the cryptocurrency market and compare the results with those obtained by Kamalov and Gurrib. We also introduce some improvements to the methods used by Kamalov and Gurrib. Our primary goal is not to predict the future price of cryptocurrencies but to understand the decision-making process of high-risk market participants in the cryptocurrency market.

The answer to the question of what input we should use for machine learning algorithms in research related to economics and financial markets may seem simple - price or return. However, studying human behavior and decision-making processes is a complex task that requires a nuanced approach. Our research aims to shed light on the importance of the type of decision-making process used by market participants in determining the accuracy of machine learning algorithms in financial forecasting.

Model

In recent years, machine learning algorithms have gained significant attention in the field of financial prediction. The study employs a variety of machine learning algorithms, including logistic regression, random forest, multilayer perceptron and LSTM. These algorithms are based on the works of Ballings et al. (2015), Borovkova and Tsiamas (2019), Kamalov (2020), Patel et

al. (2015) and Wang and Wang (2017). The author has utilized these approaches for classification in order to achieve enhanced accuracy in the results obtained..

Logistic regression, or logit regression, is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. As Juliana Tolles, Meurer, and William J (2016) have specified, logistic regression estimates the parameters of a logistic model given by the equation (Equation). This method has a convex cost function, ensuring a unique global minimum exists, and it is a linear classifier, making it robust to overfitting. However, the authors added non-linear features to logistic regression to improve accuracy. given by the equation:

$$p_x = \frac{1}{1 + e^{-(B_0 + B_1 X_1 + B_2 X_2 + \ldots + B_n X_n)}} \quad (1)$$

Where p_x is the predicted value B_0 is in intercept. through are the coefficients. B_1 through B_n are the coefficients. X_1through X_n are the features. Logistic regression (LR) has a convex cost function which ensures that a unique global minimum exists. LR has a convex cost function and any local minimum of a convex function is also a global minimum. Besides, LR is a linear classifier which means that it is robust to overfitting. To solve this problem, the author tried to add non-linear features to it (for example, different indicators have been added) Although there are more advanced classification algorithms, LR remains a standard benchmark method.

Random forests (RF) or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.

According to Ho, Tin Kam (1995) and Ho TK (1998), Random Forest is based on the bagging algorithm and uses ensemble learning techniques. It creates as many trees on the subset of the data and combines the output of all the trees. Therefore, it reduces overfitting problems in decision trees and also reduces the variance so the accuracy will be improved.

Multi-layer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers: the input layer, output layer and hidden layer, as shown in Figure 3. The input layer receives the input signal to be processed. The required task such as prediction and classification are performed by the output layer. An arbitrary number of hidden layers that are placed in between the input and output layer are the true computational engine of the MLP. Similar to a feed forward network in a MLP the data flows in the forward direction from input to output layer. The neurons in the MLP are trained with the back propagation learning algorithm. MLPs are designed to approximate any continuous function and can solve problems which are not linearly separable. The major use cases of MLP are pattern classification, recognition, prediction and approximation according to S. Abirami, P. Chitra (2020).

Long Short-Term Memory Network is an advanced RNN, a sequential network, that allows information to persist. It is capable of handling the vanishing gradient problem faced by RNN. A recurrent neural network also known as RNN is used for persistent memory as Shipra Saxena (2021) explained. Such a recurrent neural network (RNN) can process not only single data points (such as images), but also entire sequences of data (such as speech or video). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell according to Hochreiter, Sepp; Schmidhuber, Jürgen (1997). LSTM can solve hard

long time lag problems according to Felix A. Gers, Jürgen Schmidhuber and Fred Cummins (2000). These features make LSTM a good tool for forecasting financial markets.

Finally, we use the following equation to calculate the daily return:

$$r_t = ln\frac{p_t}{p_{t-1}} \quad (2)$$

Where r_t and p_t indicate the return and price for day t respectively. The data is split temporally into training and testing is set using a 75/25% ratio to improve the integrity of the experimental results.

Methodes

In this study, the authors employed various machine learning models to investigate the effect of selecting price or return as input on cryptocurrency price direction prediction accuracy. All numerical experiments were performed using Python version 3.10 and the scikit-learn and Keras packages.

For logistic regression, the authors used the LIBLINEAR solver, which is a linear classification that supports logistic regression and linear support vector machines. This solver uses a Coordinate Descent (CD) algorithm to solve optimization problems, which involves successively performing approximate minimization along coordinate directions or hyperplanes. The authors used automatic parameter selection (a.k.a L1 Regularization) and set a maximum of 200 iterations for the solvers to converge. For the random forest model, the authors used the default settings of the scikit-learn library.

For the MLP and LSTM neural network models, the authors used the Keras package based on Chollet (2018). The MLP model included two hidden layers, with the first layer having 64 and the second layer having 32 neurons, and both layers were fully connected. The authors used the ReLU activator for the hidden layers and the sigmoid activator for the output layer. The RMSProp optimizer, binary crossover loss function, and accuracy as metrics were used to compile the model. The EarlyStopping function of the Keras library was selected to prevent overfitting, with accuracy as a monitor. The authors set the mode to "max", patience to 10, restore_best_weights to True, and verbose to 0.

The LSTM model was structured similarly to the MLP model, with 64 and 32 nodes in two hidden layers, and a removal rate of 0.2 for each hidden layer. The same activation, optimizer, and loss functions were used, and EarlyStopping was applied to prevent overfitting.

To evaluate the performance of the classifiers, the authors used accuracy and the Area Under receiver operating Curve (AUC) together. The AUC provides an aggregate measure of performance across all possible classification thresholds.

The authors conducted experiments using the data of the first seven cryptocurrencies in terms of market cap, which comprise a large portion of the crypto market, with their total market dominance being above 70%. The cryptocurrencies selected for testing were Bitcoin (BTC), Ethereum (ETH), Binance Coin (BNB), Ripple (XRP), Cardano (ADA), Solana (SOL), and Polkadot (DOT). By selecting these cryptocurrencies, the authors aimed to generalize the results since they constitute more than 70% of the crypto market.

$$Y_t = \text{sigmoid}(p_t - p_{t-1}) \quad (3)$$

$$X_t = \Sigma(x_{t\text{-}i}) \quad (4)$$

According to Eq. (3), the Y_t is price difference of each day from previous day, within a sigmoid function. Therefore, the Y function for the day t, will be equal to 1 if the price has increased compared to the previous day, and 0 if it has fallen. We briefly used data from 2, 3, and 5 days prior to predict price direction. Therefore, i in the Eq.(4) is 2, 3 or 5 and t is equal to the number of days that the cryptocurrencies have been traded in the last 5 years. Therefore, x_0 means the close price (or return) on the last trading day.

Results

In this study, we evaluated the performance of four machine learning algorithms, namely RL, RF, MLP, and LSTM, for predicting the prices and returns of seven different cryptocurrencies, including Bitcoin (BTC), Ethereum (ETH), Binance Coin (BNB), Ripple (XRP), Cardano (ADA), Solana (SOL), and Polkadot (DOT). The results are presented in Table 2 and Figure 3, which show the AUC and accuracy of the models without using indicators.

Interestingly, our findings reveal that the average AUC and accuracy of return is higher than that of price, except for the AUC of RF and accuracy of MLP, where the difference is not significant. This finding suggests that predicting the returns of cryptocurrencies using machine learning algorithms may be more accurate than predicting their prices. Moreover, Table 3 presents the detailed AUC results of the models, which can be helpful for researchers who want to replicate our experiments.

We also investigated the impact of using technical indicators on the performance of the models. Table 4 shows that using indicators can improve the AUC and accuracy of the models, except for MLP, where the difference is not significant. The average improvement in the AUC and accuracy

is only 1.05%, indicating that adding technical indicators has a limited impact on the performance of the models. However, we found that using indicators can reduce the difference between the AUC and accuracy of the models that use price and those that use return. Specifically, the total difference between the price and return decreased by 15.16% when using indicators. This suggests that the AUC and accuracy of the models that use price as input can approach those of the models that use return when technical indicators are added.

Furthermore, Table 6 shows the average AUC and accuracy of the models categorized by day. Surprisingly, we found no specific pattern among the results, and adding more independent variables (i.e., the number of previous days included in the model) did not have a significant effect on the AUC and accuracy. These findings suggest that using more historical data does not necessarily improve the performance of the models.

To provide a more accurate view, we also tested the LR and RF algorithms on the data of gold spot and medium and long-term treasury bonds. The results show that the performance of these models is considerably different from the results obtained in the cryptocurrency market. Specifically, the price has a noticeably better performance than the return in all 24 types of inputs. This finding may be due to the difference in the characteristics of these assets and their markets.

In summary, our study provides insights into the performance of machine learning algorithms in predicting the prices and returns of cryptocurrencies. The results suggest that predicting the returns of cryptocurrencies using machine learning algorithms may be more accurate than predicting their prices, and adding technical indicators can improve the performance of the models. However, using more historical data does not necessarily improve the performance of the models, and the

performance of the models may vary depending on the characteristics of the assets and their markets.

Discussion

In this experimental research, we explored the effectiveness of using return and price data for financial forecasting in the cryptocurrency market. Our results indicate that using return as input data is a more appropriate choice compared to using price data. This finding contrasts with the results of similar research conducted on the stock market, which showed that price data was more effective for financial forecasting. We suggest that the reason for this discrepancy is that in the stock market, each symbol represents a company with an intrinsic value that can be estimated through fundamental analysis. This allows market participants to estimate a price target for a stock and determine whether the stock is undervalued or overvalued. Therefore, market participants in the stock market pay more attention to the current stock price rather than the stock's return in the last few days when determining the value of an asset.

In contrast, cryptocurrency market participants do not have access to information about the intrinsic value of the assets they trade. Therefore, they tend to rely on returns as an alternative tool to investigate asset performance compared to the distant past. Our research supports this intuition and shows that using return as input data is more effective than using price data for financial forecasting in the cryptocurrency market.

We also explored the impact of adding indicators to the algorithm and found that it can converge the accuracy of the price and return data. However, adding more independent variables to the model did not necessarily improve the result, indicating that changing the type of input data may be a more effective strategy to improve model performance. Furthermore, we found that the

accuracy and AUC values of the tested models were close to 50%, indicating that there is still room for improvement in the accuracy of financial forecasting using A.I models.

Interestingly, when we applied machine learning algorithms to XAUUSD data, 3-, 5-, and 10-year bonds, which are considered lower risk assets, the accuracy of the price data was significantly higher than that of the return data. This suggests that the risk of a market may be directly related to the effectiveness of return data for financial forecasting. This finding may be an interesting topic for future research into measuring market risk using A.I.

In conclusion, our study highlights the importance of considering the type of input data when conducting financial forecasting in different markets. We found that using return data was more effective for financial forecasting in the cryptocurrency market, likely due to the lack of information on intrinsic value in this market. Our results also suggest that adding indicators to the algorithm can slightly improve model performance, but changing the type of input data may be a more effective strategy. Finally, we found evidence that the risk of a market may impact the effectiveness of return data for financial forecasting.

Fig. 1 an example of RNN structure

Fig. 2 Long Short-Term Memory Structure

Fig. 3 Accuracy of machine learning algorithms using price and return as input data; (Fig-3 a) Logistic Regression AUC (LR); (Fig-3 b) Logistic Regression Accuracy (LR); (Fig-3 c) Random Forest AUC (RF); (Fig-3 d) Random Forest Accuracy (RF); (Fig-3 e) Multilayer Perceptron AUC (MLP); (Fig-3 f) Multilayer Perceptron accuracy (MLP); (Fig-3 g) Long Short-Term Memory AUC (LSTM); (Fig-3 h) Long Short-Term Memory Accuracy (LSTM).

Fig. 4 logistic regression and Random Forest algorithms have been applied for 10-years of (Fig-4 a) XAUUSD historical data and 50-years of (Fig-4 b) 1-year treasury bond, (Fig-4 c) 5-year treasury bond and (Fig-4 d) 10-year treasury bond historical data. These figures show the AUC of the Models.

Table 1: Machine learning forecasting models.

Table 2: Selected cryptocurrencies, pair and market cap.

Table 3 The AUC results of four machine learning algorithms with the price/return of 2, 3 and 5 days ago.

Table 4 Experimental results according to average AUC; (T-4 A) with indicators; (T-4 B) without indicators.

Table 5 Experimental results according to average accuracy; (T-5 A) with indicators; (T-5 B) without indicators**.**

Table 6 Experimental results according to average AUC of days; (T-6 A) with indicators; (T-6 B) without indicators.

Authors' information:

Biomedical Engineering Faculty, Amirkabir University Of Technology, Tehran, iran

Computer Science Faculty, K. N. Toosi University of Technology, Tehran, iran

Cognitive rehabilitation clinic, Shahid Beheshti University, Tehran, Iran

Availability of data and materials:

Historical data collected for this paper is from Yahoo Finance and Tradingview, all data, material and also results are available upon request.

Competing interests:

The authors declare that they have no competing interests.

Funding:

Not applicable.

Authors' contributions:

All authors read and approved the final manuscript.

Acknowledgements:

Not applicable.

References

Kamalov, F., Gurrib, I., & Rajab, K. (2021). Financial Forecasting with Machine Learning: Price Vs Return. *Journal of Computer Science*, *17*(3), 251–264.

Reid, F., & Harrigan, M. (2013). An analysis of anonymity in the bitcoin system. *In Security and privacy in social networks*, 197–223.

Ron, D., Shamir, A. (2013). Quantitative Analysis of the Full Bitcoin Transaction Graph. In: Sadeghi, AR. (eds) Financial Cryptography and Data Security. FC 2013. Lecture Notes in Computer Science, vol 7859. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-39884-1_2

Kondor, D., Csabai, I., Szüle, J., Pósfai, M., & Vattay, G. (2014). Inferring the interplay between network structure and market effects in bitcoin. New Journal of Physics, 16(12), 125003. 16. 10.1088/1367-2630/16/12/125003.

Kondor D, Pósfai M, Csabai I, Vattay G (2014) Do the Rich Get Richer? An Empirical Analysis of the Bitcoin Transaction Network. PLoS ONE 9(2): e86197. https://doi.org/10.1371/journal.pone.0086197

Alvarez-Pereira, B., Matthew Ayres, M. A., Gómez López, A. M., Gorsky, S., Hayes, S. W., Qiao, Z., & Santana, J. (2014). Network and conversation analyses of bitcoin. In 2014 complex systems summer school proceedings.

Baumann, A., Fabian, B., & Lischke, M. (2014). Exploring the bitcoin network. In Proceedings of the 10th international conference on web information systems and technologies—Volume 2: WEBIST. https://doi.org/10.5220/0004937303690374.

Fleder, M, Kester MS, Pillai S (2015) Bitcoin transaction graph analysis. arXiv preprint arXiv:1502.01657.

Di Francesco Maesa, D., Marino, A., Ricci, L.: Uncovering the bitcoin blockchain: an analysis of the full users graph. In: IEEE DSAA 2016 Proceeding of 3rd IEEE International Conference on Data Science and Advanced Analytics, Montreal, Canada, October 17-19 (2016)

Di Francesco Maesa, D., Marino, A. & Ricci, L. (2018). Data-driven analysis of Bitcoin properties: exploiting the users graph. Int J Data Sci Anal 6, 63–80. https://doi.org/10.1007/s41060-017-0074-x

Lischke M, Fabian B. Analyzing the Bitcoin Network: The First Four Years. Future Internet. 2016; 8(1):7. https://doi.org/10.3390/fi8010007

Ranshous, S., Joslyn, C. A., Kreyling, S., Nowak, K., Samatova, N. F., West, C. L., & Winters, S. (n.d.). *Exchange Pattern Mining in the Bitcoin Transaction Directed Hypergraph*.

Ranshous, S., Joslyn, C.A., Kreyling, S., Nowak, K., Samatova, N.F., West, C.L., Winters, S. (2017). Exchange pattern mining in the bitcoin transaction directed hypergraph. In Proceedings of the International Conference on Financial Cryptography and Data Security, Sliema, Malta, 3–7 April, 248–263.

Cuneyt Gurcan Akcora, Yulia R. Gel, and Murat Kantarcioglu. 2017. Blockchain: A Graph Primer. 1, 1, Article 1 (August 2017), 16 pages

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654–669

LaValley, M. P. (2008). Logistic regression. In *Circulation*, 117(18), 2395–2399

Ballings, M., van den Poel, D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*(20), 7046–7056

Borovkova, S., & Tsiamas, I. (2019). An ensemble of LSTM neural networks for high-frequency stock market classification. *Journal of Forecasting*, *38*(6),600–619

Kamalov, F. (2020) . Forecasting significant stock price changes using neural networks. Neural Comput & Applic 32, 17655–17667

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, *42*(1), 259–268

Chen, M., Hao, Y., Hwang, K., Wang, L., & Wang, L. (2017). Disease Prediction by Machine Learning over Big Data from Healthcare Communities. *IEEE Access*, *5*, 8869–8879

Tolles, J., & Meurer, W. J. (2016). Logistic Regression Relating Patient Characteristics to Outcomes JAMA Guide to Statistics and Methods. In *JAMA August* (Vol. 2). Available at http://jama.jamanetwork.com/

Tin Kam Ho. (2016). Random decision forests. Proceedings of 3rd International Conference on Document Analysis and Recognition, 278-282.

Tin Kam Ho. (1998). The random subspace method for constructing decision forests. in IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8), 832-844.

Abirami, S., & Chitra, P. (2020). Energy-efficient edge based real-time healthcare support system. In *Advances in Computers*, 117(1), 339–368.

Shipra Saxena. Introduction to Long Short Term Memory. Available at https://www.analyticsvidhya.com/blog/author/shipra_saxena/ Accessed 16 March 2021.

S. Hochreiter and J. Schmidhuber. (1997) Long Short-Term Memory. in Neural Computation, 9(8), 1735-1780.

F. A. Gers, J. Schmidhuber and F. Cummins. (2000). Learning to Forget: Continual Prediction with LSTM, in Neural Computation, 12(10), 2451-2471.

Chollet, Francois. Deep learning with Python. Simon and Schuster, 2021.