



**University of
Zurich**^{UZH}

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2023

Forecasting mid-price movement of Bitcoin futures using machine learning

Akyildirim, Erdinc ; Cepni, Oguzhan ; Corbet, Shaen ; Uddin, Gazi Salah

DOI: <https://doi.org/10.1007/s10479-021-04205-x>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-235842>

Journal Article

Published Version



The following work is licensed under a Creative Commons: Attribution 4.0 International (CC BY 4.0) License.

Originally published at:

Akyildirim, Erdinc; Cepni, Oguzhan; Corbet, Shaen; Uddin, Gazi Salah (2023). Forecasting mid-price movement of Bitcoin futures using machine learning. *Annals of Operations Research*, 330(1-2):553-584.

DOI: <https://doi.org/10.1007/s10479-021-04205-x>



Forecasting mid-price movement of Bitcoin futures using machine learning

Erdinc Akyildirim^{1,2} · Oguzhan Cepni^{3,4} · Shaen Corbet^{5,6} · Gazi Salah Uddin⁷

Accepted: 30 June 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In the aftermath of the global financial crisis and ongoing COVID-19 pandemic, investors face challenges in understanding price dynamics across assets. This paper explores the performance of the various type of machine learning algorithms (MLAs) to predict mid-price movement for Bitcoin futures prices. We use high-frequency intraday data to evaluate the relative forecasting performances across various time frequencies, ranging between 5 and 60-min. Our findings show that the average classification accuracy for five out of the six MLAs is consistently above the 50% threshold, indicating that MLAs outperform benchmark models such as ARIMA and random walk in forecasting Bitcoin futures prices. This highlights the importance and relevance of MLAs to produce accurate forecasts for bitcoin futures prices during the COVID-19 turmoil.

Keywords Cryptocurrency · Bitcoin futures · Machine learning · Covid-19 · k-Nearest neighbours · Logistic regression · Naive Bayes · Random forest · Support vector machine · Extreme gradient boosting

1 Introduction

In this research, we use high-frequency Bitcoin pricing data together with machine learning algorithms to predict mid-price movement for bitcoin futures price series across a variety of time frequencies, ranging between 5 and 60-min. The novelty of our research surrounds the

✉ Erdinc Akyildirim
erdinc.akyildirim@bf.uzh.ch

¹ Department of Banking and Finance, Burdur Mehmet Akif Ersoy University, Burdur, Turkey

² Department of Banking and Finance, University of Zurich, Zurich, Switzerland

³ Department of Economics, Copenhagen Business School, Porcelænshaven 16A, 2000 Frederiksberg, Denmark

⁴ Central Bank of the Republic of Turkey, Hacı Bayram Mah. İstiklal Cad. No:10, 06050 Ankara, Turkey

⁵ DCU Business School, Dublin City University, Dublin 9, Ireland

⁶ School of Accounting, Finance and Economics, University of Waikato, Hamilton, New Zealand

⁷ Department of Management and Engineering, Linköping University, 581 83 Linköping, Sweden

use of all available Bitcoin futures series from the Chicago Mercantile Exchange (CME). The CME had offered the product as a mechanism to hedge Bitcoin exposure or harness its performance with futures and options on futures, both of which have been markets presenting tremendous growth since their introduction (Akyildirim et al. 2020; Corbet et al. 2018a, b). While liquidity proved to be a substantial issue for some long-ranged futures such as those 6-months and 7-months into the future, after several specification tests, we present results based on the first 5-month futures products.¹ The contract is found to be quite substantial in size, representing the ownership of 5 bitcoin, as defined by the CME CF Bitcoin Reference Rate (BRR), quoted in U.S. dollars and cents per bitcoin. This exposure to Bitcoin is based on a leverage rate of 43%, therefore the investment outlay is below that of the face-value of 5 BTC. The minimum price fluctuation is \$5.00 per bitcoin, where calendar spreads are \$1.00 per bitcoin. Monthly contracts are listed for six consecutive months and two additional December contract months.²

The decision for the CME to provide Bitcoin futures on 10 December 2017 was viewed as a significant milestone in the development of such a relatively young financial product, where to this point, few major exchanges, underpinning with such reputation and historic experience had considered similar responses. The launch of CME Bitcoin futures was viewed as the first step in the new cryptocurrency's path toward legitimacy, hoping to entice institutional investors who had been, until late 2017, had been unwilling to enter the market for a variety of issues. In late 2020, CME futures possessed over \$1 billion in open interest, representing the significant growth of the market over a very short amount of time. The use of settlement pricing from multiple sources was initially identified as a strong beneficial characteristic, particularly with the many problems pertaining to cyber-criminality and illicit behaviour across exchanges and directly through product development and creation.

We contribute to the literature by evaluating the application of high-frequency bitcoin futures prices using six machine learning algorithms during the outbreak of COVID-19. We attempt to forecast the mid-point movement of CME Bitcoin futures pricing across multiple futures products during COVID-19 where we use the sign prediction rate or accuracy rate which is calculated as the proportion of times the related methodology correctly predicts the next time mid-price return direction. If the underlying process were fully random then the correct sign prediction ratio would be 50%, where any accuracy rate greater than 50% would indicate the ability of the algorithm to beat the market, further supported with the use of optimal profit ratios to measure the performance of the related classification algorithm. Furthermore, we report most of the methods to provide close results to each other, the best performing model which is the support vector machine yields on average out-sample success rates of around 56%. Another important point to note is that while the maximum value of accuracy one can obtain with the ARIMA model is only 56% among all cases considered, this number even increases up to 71% for the support vector machine algorithm. Further evidence suggests that such predictability increases in magnitude as we focus on futures with larger maturities, particularly those of 4- and 5-month duration. Such evidence indicates that Bitcoin futures products present evidence of sign predictability using machine learning.

The paper is structured as follows: previous research that guides our selected theoretical and methodological approaches are summarised in Sect. 2. Section 3 presents a thorough explanation of the wide variety of data used in our analyses, while Sect. 4 presents a concise

¹ Such a decision was made for the brevity of presentation, although further estimates and variations of time-frequency of analysis are available on request

² As per the CME, Trading terminates at 4:00 p.m. London time on the last Friday of the contract month. If this is not both a London and U.S. business day, trading terminates on the prior London and the U.S. business day.

overview of the methodologies utilised. Section 5 presents a concise overview of the results and their relevance for policy-makers and regulatory authorities, while Sect. 6 concludes.

2 Previous literature

This research develops upon three key areas of research. The first is built on the development of machine learning and the inherent processes contained therein. The second is based on the development of cryptocurrencies with an emphasis on futures pricing behaviour, while finally, the third area through which we develop our work is based on several pieces that have examined the predictability of cryptocurrency spot prices. Primarily, machine learning has been used across a variety of areas such as that of stock markets (Wittkemper and Steiner 1996; Ntakaris et al. 2018; Sirignano 2019; Huck 2019; Sirignano and Cont 2019; Huang and Liu 2020; Philip 2020); currency markets during crises (El Shazly and El Shazly 1999; Zimmermann et al. 2001; Auld and Linton 2019); energy markets such as West Texas Intermediate (Chai et al. 2018), crude oil markets (Fan et al. 2016), Cushing oil and gasoline markets (Wang et al. 2018), gold markets (Chen et al. 2020); gas markets (Ftiti et al. 2020), agricultural futures (Fang et al. 2020); copper markets (Sánchez Lasheras et al. 2015); and coal markets (Matyjaszek et al. 2019; Alameer et al. 2020); cryptocurrency spot markets (Akyildirim et al. 2020; Chowdhury et al. 2020; Chen et al. 2021) options markets (Lajbcygier 2004; De Spiegeleer et al. 2018); and futures markets (Kim et al. 2020).

Our work further develops on that based on the use of neural networks for forecasting purposes, through which a concise synthesis of the earlier literature is provided by Zhang et al. (1998). Ghodduzi et al. (2019) presented a critical review of the literature based on the application of machine learning, suggesting that Support Vector Machine (SVM), Artificial Neural Network (ANN), and Genetic Algorithms (GAs) are among the most popular techniques used to focus on energy markets. Nakano et al. (2018) previously investigated Bitcoin intraday technical trading based on artificial neural networks for the return prediction, through which Akyildirim et al. (2020) further developed by examining the predictability of the most liquid twelve cryptocurrencies are analysed at the daily and minute level frequencies. The authors found that machine learning classification algorithms reach about 55–65% predictive accuracy on average at the daily or minute level frequencies, while the support vector machines demonstrate the best and consistent results in terms of predictive accuracy compared to the logistic regression, artificial neural networks, and random forest classification algorithms. Saad et al. (2020) provided evidence of prediction accuracy of up to 99% for Bitcoin and Ethereum prices. Whereas Hubáček et al. (2019) introduced a forecasting system designed to profit from the sports-betting market specifically developing their work through the application of convolutional neural networks for match outcome prediction.

Previous research on cryptocurrency futures has been quite extensive to date. An extensive overview of the key areas of research was presented by Corbet et al. (2019), through which areas surrounding market efficiency, the development of futures exchanges and illicit behaviour are outlined. Akyildirim et al. (2020) utilised a high-frequency analysis to show significant pricing effects sourced from both fraudulent and regulatory unease within the industry, verifying that CME Bitcoin futures dominate price discovery relative to spot markets. Alexander et al. (2020) found similar evidence when considering the role that BitMEX derivatives played when similarly, informationally leading spot markets. Corbet et al. (2020) found further evidence of Bitcoin market maturity through significant response to macroeconomic news, while Koutmos (2020) found that interest rates and implied stock market

and foreign exchange market volatilities are important determinants of Bitcoin returns. de la Horra et al. (2019) analysed the demand for Bitcoin to determine whether it stems from Bitcoin's utility as a medium of exchange, a speculative asset, or as a safe-haven commodity, finding that the asset is highly speculative in the short run. Giudici and Polinesi (2019) primarily identified that Bitcoin exchange prices are positively related to each other and large exchanges Bitstamp, drive the prices. Such destabilising effects of fraud and regulatory events had also been identified by Akyildirim et al. (2020), Corbet et al. (2020), Katsiampa et al. (2019a), Katsiampa et al. (2019b) and Hu et al. (2020). Evidence supporting the predictability of Bitcoin futures prices through the use of machine learning would not explicitly be a unique characteristic to cryptocurrency markets, as it has been previously identified across several assets such as foreign exchange markets (Plakandaras et al. 2015) and several other asset markets (Akyildirim et al. 2020), it is essential to note that it is contrary to the efficient markets hypotheses, where prices should follow a martingale process. However, such a result could present another evidence supporting the developing growth of operational and technical efficiency that has been observed in such markets in recent years. Such markets have grown to such an international status, that substantial amounts of research have identified the usage of cryptocurrency markets as a hedging mechanism against the significant financial market pressures and contagion effects that were associated with the development and broad confusion associated with the COVID-19 pandemic, with emphasis on contagion effects (Akhtaruzzaman et al. 2020; Corbet et al. 2020, 2021; Mensi et al. 2020), asset price discovery (Corbet et al. 2020), safe haven effects (Conlon et al. 2020), hedge fund performance (Yarovaya et al. 2020), sentiment (Corbet et al. 2020), political risk (Sharif et al. 2020), and the basis of future research focus (Goodell 2020). As Bitcoin futures market development was observed to be a significant milestone in the transition of not only Bitcoin in isolation, but cryptocurrency as a broad financial product, it is important to specifically understand whether there exist differentials of behaviour in comparison to traditional financial market assets.

While specifically forecasting Bitcoin spot prices, using neuro-fuzzy techniques, Atsalakis et al. (2019) estimated that their selected PATSOS methodological structure performed 71% than buy-and-hold strategies. Similarly, Faghih Mohammadi Jalali and Heidari (2020) found that through the use of a first-order grey model (GM (1,1)), Bitcoin's price could be predicted accurately, to the extent of a confidence level of approximately 98% through the selection of specific periods. Alonso-Monsalve et al. (2020) found that Convolutional LSTM neural networks outperformed all the rest significantly, while CNN neural networks were also able to provide satisfactory results, especially in the Bitcoin, Ether and Litecoin cryptocurrency markets. Further, Ma et al. (2020) found that the proposed novel MRS-MIDAS model exhibits statistically significant improvement for forecasting the RV of Bitcoin Between 2011 and 2018, Adcock and Gradojevic (2019) found that backpropagation neural networks dominate various competing models in terms of their forecast accuracy. Further, when attempting to predict Bitcoin bubble crashes, Shu and Zhu (2020) showed that an LPPLS confidence indicator presented superior detection capability to bubbles and accurately forecast the bubble crashes, even if a bubble existed for only a short period of time. Such work built on that of the same structure of Samitas et al. (2020), who found that the effectiveness of machine learning reached 98.8% as an early warning system to predict the financial crisis. Zoumpikas et al. (2020) found that Convolutional Neural Network and four types of Recurrent Neural Network including the Long Short Term Memory network, the Stacked Long Short Term Memory network, the Bidirectional Long Short Term Memory network, and the Gated Recurrent Unit network could be utilised to predict the Ethereum closing price in real-time with promising accuracy and experimentally proven profitability. Such results present evidence that pre-

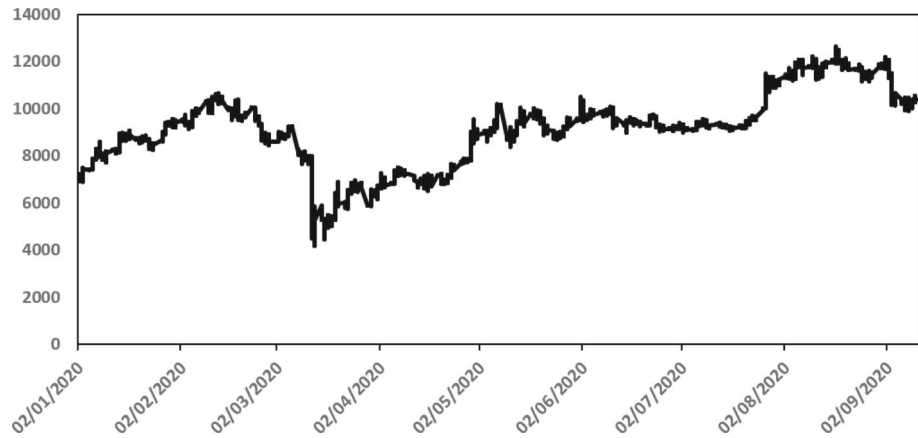


Fig. 1 1-month futures mid-price at 5-min frequency. Note: The above figure presents a plot of the 1-month futures price at 5-min frequency during 2020

diction of cryptocurrency markets was statistically possible in direct opposition to that of the efficient markets hypothesis [previously examined in cryptocurrency markets by Sensoy (2019) and Akyildirim et al. (2020)], but this is not the first market to have presented such evidence as Plakandaras et al. (2015) had previously identified similar atheoretical outcomes had been identified on spot foreign exchange markets.³

3 Data

In this study, we use dollar-denominated Bitcoin futures data from Chicago Mercantile Exchange. We obtain the data for 1-month futures up to 9-month futures at a minutely level which have the earliest starting points at different dates. To have enough observations to draw meaningful and robust conclusions we use only 1-month to 5-month futures data with an initial date of 2 January 2020 and an end date of 10 September 2020. Bitcoin futures can be traded at any time during the day at CME after 11:00 PM on Sunday until 10:00 PM on Friday (because of daylight saving time change on March 8, 2020, the trading hours shifted as after 10:00 PM on Sunday till 9:00 PM on Friday). The time interval that we studied corresponds to 217 trading days which we sample at 5-, 10-, 15-, 30-, 60-min frequencies. For each time frequency, we compute the mid-price from the best ask and bid prices using the last observation in that time interval. Then we compute the log-returns for each time scale from these mid-prices. Table 1 shows the total number of observations for the mid-price returns of bitcoin futures at different time scales, while Figure 1 presents a plot of the 1-month futures price at 5-min frequency during 2020. For instance, while there are 4345 observations at the 60-min frequency, this number increases to 51733 at the 5-min frequency. Table 2 provides descriptive statistics for mid-price future returns for different maturities and time scales. As it is clear from the table, mean and median values are always around zero independent of time to maturity and time-frequency. As an expected min, max and standard deviation values get larger in absolute value as the time to maturity increases.

³ The authors present evidence against the efficient markets hypotheses based on the spot markets for EUR/USD, USD/JPY, AUD/NOK, NZD/BRL and ZAR/PHP.

Table 1 Number of observations for the mid price returns of bitcoin futures at different time scales for different train/test set divisions

	Total	5-min 51,733	10-min 25,885	15-min 17,269	30-min 8653	60-min 4345
0.7/0.3	Train	36,213	18,119	12,088	6057	3041
	Test	15,520	7766	5181	2596	1304
0.8/0.2	Train	41,386	20,708	13,815	6922	3476
	Test	10,347	5177	3454	1731	869
0.9/0.1	Train	46,559	23,296	15,542	7787	3910
	Test	5174	2589	1727	866	435

We use 1-month through 5-month CME Bitcoin futures data with initial date of 2 January 2020 and end date of 10 September 2020. Bitcoin futures can be traded at any time during day at CME after 23:00 PM on Sunday till 22:00 PM on Friday (due to daylight saving time change on March 8, 2020, the trading hours shifted as after 22:00 PM on Sunday till 21:00 PM on Friday). Results based on a variety of other time frequencies and other Bitcoin futures markets are available from the authors upon request and have been omitted from the above table due to brevity of presentation

4 Classification algorithms

4.1 Machine learning models

We apply six different machine learning algorithms (k-Nearest Neighbours, Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine, Extreme Gradient Boosting) to classify the target variable (it refers to the return of mid-price of the bitcoin futures in our study) as “up” or “down” at varying time frequencies. These methods are selected due to their popularity and fast implementation, and they are performed with Python’s well-known scikit-learn package. In what follows, we briefly describe how each of these classification algorithms helps to forecast the sign of the target variable.

4.1.1 k-Nearest neighbours classifier

The *k*-nearest neighbours’ algorithm (kNN) is a commonly used, simple yet successful classification method that has been applied in a large number of classification and regression problems such as handwritten digits and satellite image scenes (Melgani and Bruzzone 2004; Munder and Gavrila 2006). The kNN is a supervised machine learning model where the model learns from the labelled data how to map the inputs to the desired output so that it can make predictions on test data. It is a non-parametric algorithm as it does not make any assumptions about the data, such as normality. The kNN model picks an entry in the database and then looks at the ‘*k*’ entries in the database which are closest to the chosen point. Then, the data point is assigned the label of the majority of the ‘*k*’ closest points. For instance, if $k = 6$ with 4 of points being as ‘up’ and 2 as ‘down’, the data point in question would be labelled ‘up’ since ‘up’ is the majority class.

More generally, the kNN algorithm works as follows: For a given value of *k*, it computes the distance between the test data and each row of the training data by using a distance metric like Euclidean metric (some of the other metrics that can also be used are cityblock, Chebychev, correlation, and cosine). The distance values are sorted in ascending order and then top *k* elements are extracted from the sorted array. It finds the most frequent class among

Table 2 Descriptive Statistics for mid price future returns for different maturities and time scales

	5-min	10-min	15-min	30-min	60-min
<i>1-month futures</i>					
Mean	0.000007	0.000014	0.000021	0.000042	0.000084
Median	0.00	0.00	0.00	0.00	0.00
Min	-0.13	-0.13	-0.15	-0.19	-0.22
Max	0.12	0.14	0.13	0.15	0.12
SD	0.003	0.005	0.006	0.008	0.011
<i>2-month futures</i>					
Mean	0.000007	0.000014	0.000021	0.000043	0.000085
Median	0.00	0.00	0.00	0.00	0.00
Min	-0.41	-0.41	-0.41	-0.41	-0.41
Max	0.40	0.41	0.41	0.42	0.42
SD	0.005	0.007	0.008	0.012	0.016
<i>3-month futures</i>					
Mean	0.000007	0.000014	0.000021	0.000042	0.000084
Median	0.00	0.00	0.00	0.00	0.00
Min	-0.40	-0.40	-0.40	-0.40	-0.40
Max	0.41	0.41	0.41	0.41	0.41
SD	0.007	0.010	0.012	0.017	0.024
<i>4-month futures</i>					
Mean	0.000006	0.000013	0.000019	0.000038	0.000076
Median	0.00	0.00	0.00	0.00	0.00
Min	-0.61	-0.61	-0.60	-0.61	-0.61
Max	0.60	0.60	0.60	0.60	0.60
SD	0.010	0.014	0.017	0.023	0.033
<i>5-month futures</i>					
Mean	0.000007	0.000014	0.000021	0.000042	0.000083
Median	0.00	0.00	0.00	0.00	0.00
Min	-0.69	-0.69	-0.69	-0.70	-0.72
Max	0.79	0.79	0.74	0.74	0.77
SD	0.018	0.024	0.028	0.038	0.054

We use 1-month through 5-month CME Bitcoin futures data with initial date of 2 January 2020 and end date of 10 September 2020. Bitcoin futures can be traded at any time during day at CME after 23:00 PM on Sunday till 22:00 PM on Friday (due to daylight saving time change on March 8, 2020, the trading hours shifted as after 22:00 PM on Sunday till 21:00 PM on Friday). Results based on a variety of other time frequencies and other Bitcoin futures markets are available from the authors upon request and have been omitted from the above table due to brevity of presentation

these k elements and returns as the predicted class. In our application of kNN, we optimize the algorithm over the k values from 1 to 20.

4.1.2 Naive Bayes classifier

The Naive Bayes is another widely used classification algorithm as it is easy to build and particularly useful for very large data sets (Chawla et al. 2002; Zhang et al. 2014). This

method is a supervised learning algorithm based on the application of the Bayes' theorem, and also called a probabilistic machine learning algorithm. It makes the "naive" assumption that the input features are conditionally independent of each other given the classification. If this assumption holds then the naive Bayes classifier may perform even better than more complicated models. However, in real life, most of the time it is not possible to get a set of predictors which are completely independent.

The naive Bayes classifier assigns observations to the most probable class by first estimating the densities of the predictors within each class. As a second step, it computes the posterior probabilities according to Bayes' rule:

$$\hat{P}(Y = k | X_1, \dots, X_P) = \frac{\pi(Y = k) \prod_{j=1}^P P(X_j | Y = k)}{\sum_{k=1}^K \pi(Y = k) \prod_{j=1}^P P(X_j | Y = k)} \quad (1)$$

where Y is the random variable corresponding to the class index of an observation, X_1, \dots, X_P are the random predictors of an observation, and $\pi(Y = k)$ is the prior probability that a class index is k . Finally, it classifies an observation by estimating the posterior probability for each class, and then assigns the observation to the class yielding the maximum posterior probability.

4.1.3 Logistic regression classifier

The Logistic Regression is a machine learning classification algorithm that is used to forecast the probability of a categorical dependent variable (Chen et al. 2013; Fischer and Krauss 2018). In logistic regression, the outcome of the target variable is dichotomous (i.e., there are only two possible classes). The classification algorithm forecasts the probability of occurrence of a binary event utilizing a logit function. More explicitly, logistic regression outputs a probability value by using the logistic sigmoid function and then this probability value is mapped to two discrete classes.

In our case, we have a binary classification problem of identifying the next time excess return as up or down. Logistic regression assigns probabilities to each row of the feature's matrix X . Let us denote the sample size of the dataset with N and thus we have N rows of the input vector. Given the set of d features, i.e., $x = (x_1, \dots, x_d)$, and parameter vector w , the logistic regression with the penalty term minimizes the following optimization problem:

$$\min_{w,c} \frac{w^T w}{2} + C \sum_{i=1}^N \log(\exp(-y_i(x_i^T w + c)) + 1) \quad (2)$$

where we find the optimal value of C by making a grid search over a set of reasonable values for C .

4.1.4 Random forest classifier

The Random Forest Classifier is an ensemble algorithm such that it combines more than one algorithm of the same or different kind for classifying objects. Decision trees are the building blocks of the random forest model. In other words, the random forest consists of a large number of individual decision trees that function as an ensemble. Random forest classifier creates a set of decision trees from a randomly selected subset of the training set, and each tree makes a class prediction. It then sums the votes from different decision trees to decide

the final class of the test object. For instance, assume that there are 5 points in our training set that is (x_1, x_2, \dots, x_5) with corresponding labels (y_1, y_2, \dots, y_5) then random forest may create four decision trees taking the input of subset such as (x_1, x_2, x_3, x_4) , (x_1, x_2, x_3, x_5) , (x_1, x_2, x_4, x_5) , (x_2, x_3, x_4, x_5) . If three of the decision trees vote for “up” against “down” then the random forest predicts “up”. This works efficiently because a single decision tree may produce noise but a large number of relatively uncorrelated trees operating as a choir will reduce the effect of noise, resulting in more accurate results.

More generally, in the random forest method as proposed by Breiman (2001), a random vector θ_k is generated, independent of the past random vectors $\theta_1, \dots, \theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and θ_k resulting in a classifier $h(x, \theta_k)$ where x is an input vector. In random selection, θ consists of several independent random integers between 1 and K . The nature and dimension of θ depend on its use in tree construction. After a large number of trees are generated, they vote for the most popular class. This procedure is called random forest. A random forest is a classifier consisting of a collection of tree-structured classifiers $h(x, \theta_k)$, $k = 1, \dots$ where the θ_k 's are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x .

4.1.5 Support vector machine classifier

The Support Vector Machine (SVM) is a supervised machine learning algorithm used for both regression and classification tasks (Fauvel et al. 2008; Suykens and Vandewalle 1999). The support vector machine algorithm’s objective is to find a hyperplane in an N -dimensional space where N is the number of features that distinctly classify the data points. Hyperplanes can be thought of as decision boundaries that classify the data points. Data points falling on different sides of the hyperplane can be assigned to different classes. Support vectors are described as the data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. The margin of the classifier is maximized using these support vectors. In more technical terms, the above process can be summarized as follows. Given the training vectors x_i for $i = 1, 2, \dots, N$ with a sample size of N observations, the support vector machine classification algorithm solves the following problem given by

$$\min_{w, h, \xi} \frac{w^T w}{2} + C \sum_{i=1}^N \xi_i \tag{3}$$

subject to $y_i(w^T \phi(x_i)) \geq 1 - \xi_i$ and $\xi_i \geq 0, i = 1, 2, \dots, N$. The dual of the above problem is given by

$$\min_{\alpha} \frac{\alpha^T Q \alpha}{2} - e^T \alpha \tag{4}$$

subject to $y^T \alpha = 0$ and $0 \leq \alpha_i \leq C$ for $i = 1, 2, \dots, N$, where e is the vector of all ones, $C > 0$ is the upper bound. Q is an n by n positive semi-definite matrix. $Q_{ij} = y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel. Here training vectors are implicitly mapped into higher dimensional space by the function ϕ . The decision function in the support vector machines classification is given by

$$\text{sign} \left(\sum_{i=1}^N y_i \alpha_i K(x_i, x) + \rho \right). \tag{5}$$

The optimization problem in Equation 3 can be solved globally using the Karush-Kuhn-Tucker (KKT) conditions. This optimization problem depends on the choice of the Kernel functions. Our study employs the Gaussian (rbf) kernel, which is denoted by $\exp(-\gamma \|x - x'\|^2)$ where γ must be greater than 0. When SVM is implemented, we try to find an optimal value of C and γ for each stock by using a grid search for each of these parameters.

4.1.6 Extreme gradient boosting classifier

The Extreme Gradient Boosting (XGBoost) is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework. As we said before, an ensemble method is a machine learning technique that combines several base models to produce one optimal predictive model (Weldegebriel et al. 2020). An algorithm is called boosting if it works by adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor until the training data is accurately predicted or reproduced by the model. A method is called gradient boosting if, instead of assigning different weights to the classifiers after every iteration, it fits the new model to new residuals of the previous prediction and then minimizes the loss when adding the latest prediction. Namely, if a model is updated using gradient descent, then it is called gradient boosting. XGBoost improves upon the base gradient boosting framework through systems optimization and algorithmic enhancements. Some of these enhancements can be listed as parallelised tree building, tree pruning using a depth-first approach, cache awareness and out-of-core computing, regularisation for avoiding over-fitting, efficient handling of missing data, and in-built cross-validation capability.

4.2 Calculating the prediction success and potential profitability

Assume that the real label of the target variable is denoted by Y and predicted label is denoted by Y' , we employ the following two measures to assess the usefulness of our selected forecasting techniques:

- The Sign Prediction Ratio (SPR): Correctly predicted excess return direction is assigned 1 and 0 otherwise, then sign prediction ratio is calculated by

$$SPR = \frac{\sum_{j=1}^M matches(Y_j, Y'_j)}{M}, \quad (6)$$

where

$$matches(Y_j, Y'_j) = \begin{cases} 1 & \text{if } Y_j = Y'_j \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

and M denotes the size of the set for which the sign prediction ratio is measured.

- The Maximum Return is obtained by adding absolute value of all the excess returns (denoted by h)

$$MaxReturn = \sum_{j=1}^M abs(h_j) \quad (8)$$

and represents the maximum achievable return assuming perfect forecast.

- The Total Return is computed in the following way

$$Total\ Return = \sum_{j=1}^M sign(Y'_j) * h_j \quad (9)$$

where $sign$ is the standard sign function and $*$ denotes the usual multiplication. Notice that the better the prediction method, the larger the total return is.

- Ideal Profit Ratio is the ratio of the total return in Eq. (9) and the maximum return in Eq. (8).

$$IPR = \frac{Total\ Return}{Max\ Return} \quad (10)$$

5 Empirical results

As explained in Sect. 3, we sample our data at five different time scales including 5-, 10-, 15-, 30-, 60-min time scales and then implement six different machine learning algorithms that is kNN, logistic regression, naive Bayes, random forest, SVM, XGBoost which are described in the previous section. The target variable for all of these classification methods is the sign of the one-step ahead mid-price return of Bitcoin futures with different maturities at different frequencies. As features again we use the one-step before mid-price returns of Bitcoin futures. As an example, when the target variable is the sign of 5-min mid-price return of 1-month futures at time $t + 1$, we use the 5- min returns of 1-, 2-, 3-, 4-, 5-month futures at time t as features in the machine learning algorithm. We consider three different divisions of the dataset as train and test sets, called hold-outs. For the first hold-out, we take 70% of the total sample size rounded to the closest integer value as the training sample size and the remaining 30% as the test sample size. Similarly, we also look at the 0.8/0.2 and 0.9/0.1 divisions as train/test set partitions.

The number of observations for the mid-price returns of Bitcoin futures at different time frequencies for different train/test set divisions is reported in Table 1. For instance, while there are 51,733 mid-price returns in total at the 5-min scale, only 4345 mid-price returns are available at the 60-min scale. 0.7/0.3 hold-out at 5-min frequency corresponds to 36,213-time intervals in the train set and 15,520-time intervals in the test set. Similarly, 0.8/0.2 and 0.9/0.1 train vs test set partitions correspond to 41,386/10,347 and 46,559/5,174 five-minute time intervals, respectively. The descriptive statistics for mid-price returns of Bitcoin futures at different time scales with different maturities are presented in Table 2, showing that both mean and median values are almost zero across different maturities and time frequencies. Minimum (maximum) values of the returns are getting smaller (larger) as the time to maturity increases across all the time scales. However, minimum and maximum values do not change significantly from a one-time scale to another within the same Bitcoin futures (except the 1-month futures). As expected, standard deviations increase both with time to maturity and time-frequency within the same futures.

The performance of the different machine learning algorithms is compared based on two key metrics. First, we use the sign prediction rate or accuracy rate calculated as the proportion of times the related methodology correctly predicts the next time mid-price return direction. If the underlying process were entirely random, then the correct sign prediction ratio would be 50%. However, in our case, it is essential to note that we use the information contained only in the Bitcoin futures. In other words, we do not use any other information source, which can also be challenging to determine as they are many different parameters affecting

Bitcoin prices. Hence, any accuracy rate greater than 50% already indicates the success of the algorithm to beat the market. Second, we also apply the ideal profit ratio to measure the performance of the related classification algorithm. As formalized in Sect. 4, the ideal profit ratio is the ratio of the return generated by a given algorithm to the perfect sign forecast. The numerical results are produced with a 2.6 GHz Intel Core i7 computer utilizing Python 3.7 with the Scikit-learn machine learning package. Running the algorithms at the hourly timescale can be completed in the order of seconds, whereas at higher frequencies, such as the five minutes sampling frequency, the computational time requirements for training and prediction increase to the order of minutes for the random forest algorithm.

Tables 3, 4, 5, 6, 7, and 8 present the accuracy rates for the train (in-sample) and test (out-sample) periods in the first two columns for Bitcoin futures with maturities from 1-month to 5-month at different time scales. Similarly, ideal profit ratios are given in the following column for out of the sample period. Mean value (together with t-stat), standard deviation, maximum, and minimum of each column across different maturities and time frequencies are given below the tables. Table 3 provides results for the kNN algorithm, which are computed by optimizing over neighbourhood numbers from 1 to 20. The kNN methodology yields an average out-of-sample (in-sample) success ratio of 55% (77%) for the first hold-out, 55% (75%) for the second hold-out, and 56% (75%) for the third hold-out. The maximum average ideal profit ratio is computed around 6% for the 0.9/0.1 division of the data set. The kNN methodology yields the highest in-sample accuracy results after a random forest algorithm with a maximum value reaching as high as %94 for 5-month futures at 5-min frequency. Similarly, the maximum value for the out-of-sample success rate (66%) is attained by month futures at 15-min frequency. In most cases, we observe that the accuracy rate increases for the same maturity futures under the kNN method as the time frequency decreases. It is also evident from Table 3 that for most of the cases, we obtain a positive ideal profit ratio with a maximum value of 23 % in the first hold-out (27% for the second hold-out, 31% for the third hold-out) with 3-months futures at 60 min frequency.

Table 4 provides the results for the logistic regression algorithm, which is based on a linear classification. We observe that this method yields relatively stable results across different maturities and time scales. For instance, the average success rate for both in the sample and out-sample periods is almost always 54% for three different hold-outs. The same is also true for the min (51%) and max (57%) values for different cases. Again, we obtain a positive ideal profit ratio with this prediction method in most cases. Although the average values of the ideal profit ratios are not quite high, the maximum value of them (29% for 0.7/0.3 division, 32% for 0.8/0.2 division, 35% for 0.9/0.1 division) can be considered satisfactory. Table 5 shows the results for Naïve Bayes classification which is the worst-performing one among the six different methodologies. The in-sample success ratios are almost always indistinguishable from 50% which is nothing but the result coming from a random walk model for which the empirical results are given in Table 9. Although the max accuracy rate can reach even up to 60% for 1-month futures at 5 min frequency for the first hold-out, the average accuracy rate is only 45% across the different cases. This result also holds for 0.8/0.2 and 0.9/0.1 divisions. Similar results are also obtained for the ideal profit ratios. We receive an average negative ideal profit ratio only for the Naïve Bayes algorithm.

As can be noted in Table 6, the in-sample fits of the random forest algorithm are the highest among all the machine learning algorithms considered. The highest average in-sample success rate can even reach up to 87% for the first hold-out, 83% for the second hold-out, and 87% for the third hold-out. However, the out-of-sample average performance is significantly worse than the in-sample fits. This designates the high variance in the random forest classification with high sample fit to the noisy data but lower out-of-sample performance. For all of the

Table 3 KNN classification: in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
1-Month futures	5-min	0.86	0.56	-0.002	0.86	0.55	0.009	0.86	0.54	0.86	0.54	0.033	
	10-min	0.83	0.54	0.004	0.83	0.55	0.024	0.83	0.56	0.69	0.56	0.091	
	15-min	0.81	0.54	-0.004	0.82	0.53	0.016	0.66	0.54	0.66	0.54	0.100	
	30-min	0.66	0.54	0.074	0.66	0.54	0.022	0.62	0.57	0.62	0.57	0.090	
2-Month futures	5-min	0.62	0.53	0.065	0.62	0.53	0.083	0.64	0.52	0.64	0.52	0.032	
	10-min	0.84	0.55	0.000	0.85	0.54	-0.002	0.85	0.54	0.85	0.54	0.020	
	15-min	0.81	0.55	0.028	0.82	0.55	0.034	0.72	0.56	0.72	0.56	0.084	
	30-min	0.79	0.54	-0.006	0.68	0.53	0.041	0.71	0.54	0.71	0.54	0.038	
3-Month futures	5-min	0.64	0.55	0.107	0.69	0.54	0.040	0.61	0.55	0.61	0.55	0.147	
	10-min	0.61	0.53	0.051	0.60	0.55	0.115	1.00	0.52	1.00	0.52	0.083	
	15-min	0.84	0.55	-0.030	0.85	0.54	-0.016	0.84	0.54	0.84	0.54	-0.080	
	30-min	0.81	0.54	0.043	0.81	0.54	0.063	0.72	0.55	0.72	0.55	0.101	
60-min	5-min	0.80	0.54	0.021	0.62	0.54	0.101	0.66	0.55	0.66	0.55	0.222	
	10-min	0.63	0.55	0.168	0.63	0.55	0.177	0.63	0.55	0.63	0.55	0.295	
	15-min	0.61	0.53	0.231	0.60	0.54	0.279	0.62	0.53	0.62	0.53	0.311	
	30-min	0.61	0.53	0.231	0.60	0.54	0.279	0.62	0.53	0.62	0.53	0.311	

Table 3 continued

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	
4-Month futures	5-min	0.84	0.56	-0.109	0.85	0.56	-0.114	0.85	0.55	0.55	0.017		
	10-min	0.82	0.54	0.003	0.83	0.54	0.018	0.82	0.55	-0.264			
	15-min	0.80	0.54	-0.181	0.66	0.54	-0.256	0.66	0.56	-0.251			
	30-min	0.60	0.54	-0.252	0.66	0.54	-0.318	0.71	0.53	0.030			
5-Month futures	60-min	0.60	0.55	-0.056	0.61	0.53	-0.070	0.61	0.52	0.072			
	5-min	0.94	0.61	0.026	0.94	0.61	0.038	0.93	0.64	0.032			
	10-min	0.93	0.61	-0.008	0.93	0.61	-0.002	0.92	0.65	-0.056			
	15-min	0.91	0.60	0.012	0.79	0.61	0.001	0.91	0.66	0.146			
30-min	0.79	0.60	0.116	0.78	0.60	-0.013	0.69	0.65	0.177				
	0.90	0.57	0.029	0.66	0.57	-0.053	0.89	0.65	0.045				
mean	0.77***	0.55***	0.01	0.75***	0.55***	0.01	0.75***	0.56***	0.06				
t-stat	11.96	11.36		11.28	10.44		10.56	7.28					
std	0.11	0.02	0.10	0.11	0.03	0.12	0.12	0.05	0.13				
max	0.94	0.61	0.23	0.94	0.61	0.28	1.00	0.66	0.31				
min	0.60	0.53	-0.25	0.60	0.53	-0.32	0.61	0.52	-0.26				

The t-statistics values are for the one-sided t-test

***, ** and * denote significant at the 1%, 5% and 10% level respectively

Table 4 Logistic regression: in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
1-Month futures	5-min	0.54	0.54	0.038	0.54	0.54	0.065	0.54	0.55	0.087	0.54	0.55	0.087
	10-min	0.53	0.52	-0.013	0.53	0.53	0.018	0.53	0.53	0.017	0.53	0.53	0.017
	15-min	0.54	0.53	0.019	0.54	0.53	0.040	0.54	0.53	0.018	0.54	0.53	0.018
	30-min	0.56	0.55	0.056	0.56	0.55	0.092	0.55	0.55	0.064	0.55	0.55	0.064
	60-min	0.54	0.56	0.061	0.53	0.56	0.066	0.54	0.55	0.050	0.54	0.55	0.050
	5-min	0.54	0.54	0.064	0.54	0.55	0.069	0.54	0.55	0.093	0.54	0.55	0.093
2-Month futures	10-min	0.54	0.52	0.012	0.53	0.53	0.017	0.54	0.53	0.027	0.54	0.53	0.027
	15-min	0.54	0.53	0.035	0.52	0.52	0.061	0.52	0.51	0.027	0.52	0.51	0.027
	30-min	0.55	0.55	0.048	0.55	0.54	0.082	0.55	0.54	0.035	0.55	0.54	0.035
	60-min	0.55	0.55	0.109	0.55	0.55	0.088	0.55	0.54	0.088	0.55	0.54	0.088
	5-min	0.54	0.54	0.081	0.54	0.55	0.092	0.54	0.56	0.093	0.54	0.56	0.093
	10-min	0.54	0.52	0.068	0.53	0.53	0.087	0.51	0.51	0.030	0.51	0.51	0.030
3-Month futures	15-min	0.55	0.54	0.151	0.54	0.54	0.176	0.52	0.51	0.004	0.52	0.51	0.004
	30-min	0.56	0.56	0.163	0.56	0.55	0.173	0.55	0.54	-0.240	0.55	0.54	-0.240
	60-min	0.54	0.56	0.291	0.54	0.57	0.319	0.55	0.56	0.349	0.55	0.56	0.349
	5-min	0.54	0.52	0.011	0.54	0.53	0.020	0.53	0.53	-0.105	0.53	0.53	-0.105
	10-min	0.54	0.51	-0.067	0.53	0.52	-0.059	0.53	0.53	0.004	0.53	0.53	0.004
	15-min	0.54	0.52	-0.055	0.51	0.51	0.049	0.51	0.50	0.000	0.51	0.50	0.000
4-Month futures	30-min	0.56	0.55	0.045	0.55	0.54	0.151	0.55	0.53	0.188	0.55	0.53	0.188
	60-min	0.55	0.56	0.181	0.55	0.57	0.194	0.54	0.55	0.065	0.54	0.55	0.065

Table 4 continued

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	In-sample
5-Month futures	5-min	0.53	0.53	-0.070	0.53	0.53	-0.068	0.53	0.53	0.53	0.53	0.53	0.023
	10-min	0.52	0.51	-0.167	0.52	0.52	-0.160	0.53	0.53	0.51	0.51	0.090	
	15-min	0.54	0.53	-0.066	0.54	0.53	-0.072	0.54	0.50	0.50	0.50	0.145	
	30-min	0.55	0.56	0.028	0.56	0.54	0.020	0.56	0.54	0.54	0.54	-0.119	
	60-min	0.51	0.57	-0.011	0.53	0.55	-0.008	0.53	0.53	0.53	0.53	0.035	
mean	0.54***	0.54***	0.04	0.54***	0.54***	0.06	0.54***	0.54***	0.53***	0.53***	0.53***	0.04	
t-stat	19.52	11.07		15.62	12.21		13.66	9.46					
std	0.01	0.02	0.09	0.01	0.02	0.10	0.01	0.02	0.02	0.02	0.02	0.11	
max	0.56	0.57	0.29	0.56	0.57	0.32	0.56	0.56	0.56	0.56	0.56	0.35	
min	0.51	0.51	-0.17	0.51	0.51	-0.16	0.51	0.51	0.51	0.51	0.51	-0.24	

The t-statistics values are for the one-sided t-test
 ***, **, and * denote significant at the 1%, 5% and 10% level respectively

Table 5 Naive Bayes Classification: in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
1-Month futures	5-min	0.51	0.60	0.015	0.015	0.51	0.57	0.024	0.024	0.51	0.58	0.045	0.045
	10-min	0.50	0.43	0.001	0.001	0.50	0.45	-0.007	-0.007	0.50	0.44	-0.031	-0.031
	15-min	0.50	0.45	0.008	0.008	0.50	0.46	0.005	0.005	0.50	0.45	-0.031	-0.031
	30-min	0.51	0.47	0.020	0.020	0.51	0.47	0.022	0.022	0.51	0.46	-0.040	-0.040
2-Month futures	60-min	0.51	0.54	-0.011	-0.011	0.51	0.52	-0.010	-0.010	0.52	0.53	0.069	0.069
	5-min	0.50	0.41	0.014	0.014	0.50	0.44	0.009	0.009	0.50	0.43	-0.006	-0.006
	10-min	0.50	0.44	0.009	0.009	0.50	0.46	0.002	0.002	0.50	0.45	-0.038	-0.038
	15-min	0.50	0.46	0.014	0.014	0.50	0.47	0.007	0.007	0.50	0.46	-0.042	-0.042
3-Month futures	30-min	0.51	0.47	0.015	0.015	0.51	0.48	0.016	0.016	0.51	0.47	-0.050	-0.050
	60-min	0.51	0.48	0.051	0.051	0.51	0.49	0.014	0.014	0.50	0.48	-0.081	-0.081
	5-min	0.50	0.42	0.022	0.022	0.53	0.56	0.070	0.070	0.52	0.57	0.005	0.005
	10-min	0.50	0.45	0.026	0.026	0.50	0.46	0.017	0.017	0.50	0.45	-0.044	-0.044
	15-min	0.51	0.46	0.032	0.032	0.50	0.47	0.022	0.022	0.50	0.46	-0.041	-0.041
	30-min	0.51	0.48	0.037	0.037	0.51	0.48	0.038	0.038	0.51	0.47	-0.042	-0.042
60-min	0.52	0.50	0.359	0.359	0.52	0.53	0.402	0.402	0.51	0.47	0.262	0.262	

Table 5 continued

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	
4-Month futures	5-min	0.50	0.41	0.002	0.44	0.50	0.44	0.014	0.50	0.50	0.42	-0.005	
	10-min	0.50	0.44	-0.001	0.45	0.50	0.45	-0.006	0.50	0.50	0.44	-0.029	
	15-min	0.51	0.46	-0.076	0.47	0.51	0.47	-0.095	0.51	0.51	0.45	-0.032	
	30-min	0.51	0.47	-0.096	0.47	0.51	0.47	-0.123	0.51	0.51	0.46	-0.027	
	60-min	0.53	0.54	0.257	0.54	0.53	0.54	0.289	0.51	0.51	0.49	-0.187	
	5-min	0.51	0.33	-0.045	0.36	0.51	0.36	-0.049	0.50	0.50	0.30	-0.057	
5-Month futures	10-min	0.51	0.37	-0.042	0.38	0.51	0.38	-0.046	0.51	0.51	0.31	0.004	
	15-min	0.51	0.38	-0.042	0.39	0.51	0.39	-0.047	0.51	0.51	0.32	-0.032	
	30-min	0.52	0.41	-0.058	0.41	0.52	0.41	-0.067	0.52	0.52	0.34	-0.088	
	60-min	0.52	0.43	-0.094	0.43	0.52	0.43	-0.116	0.52	0.52	0.34	-0.061	
	mean	0.51***	0.45***	0.02	0.47***	0.51***	0.47***	0.02	0.51***	0.51***	0.44***	-0.02	
	t-stat	5.67	-4.43		-3.34	5.06	-3.34		5.94	5.94	-4.03		
std		0.01	0.06	0.10	0.05	0.01	0.05	0.11	0.01	0.01	0.07	0.08	
	max	0.53	0.60	0.36	0.57	0.53	0.57	0.40	0.52	0.52	0.58	0.26	
	min	0.50	0.33	-0.10	0.36	0.50	0.36	-0.12	0.50	0.50	0.30	-0.19	

The t-statistics values are for the one-sided t-test

***, ** and * denote significant at the 1%, 5% and 10% level respectively

Table 6 Random Forest (RF) classification: in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis

Target variable	Time-frequency	Train/test = 0.7/0.3		Train/test = 0.8/0.2		Train/test = 0.9/0.1	
		Success ratio		Success ratio		Success ratio	
		In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
1-Month futures	5-min	0.98	0.55	0.98	0.54	0.99	0.53
	10-min	0.99	0.54	0.99	0.54	0.99	0.53
	15-min	0.98	0.54	0.71	0.55	0.70	0.54
	30-min	0.74	0.55	0.73	0.56	0.73	0.56
	60-min	0.84	0.54	0.80	0.55	0.79	0.54
	5-min	0.69	0.55	0.69	0.55	0.68	0.54
2-Month futures	10-min	0.98	0.54	0.99	0.54	0.98	0.53
	15-min	0.68	0.53	0.72	0.53	0.71	0.53
	30-min	0.73	0.56	0.69	0.55	0.69	0.56
	60-min	0.82	0.53	0.80	0.54	0.79	0.52
	5-min	0.68	0.55	0.68	0.54	0.99	0.54
	10-min	0.98	0.54	0.71	0.53	0.68	0.53
3-Month futures	15-min	0.72	0.54	0.70	0.55	0.69	0.55
	30-min	0.73	0.55	0.71	0.56	0.71	0.55
	60-min	0.77	0.55	0.82	0.58	0.79	0.56
	5-min	0.68	0.55	0.68	0.54	0.99	0.54
	10-min	0.98	0.54	0.71	0.53	0.68	0.53
	15-min	0.72	0.54	0.70	0.55	0.69	0.55
Ideal-profit ratio	5-min						
	10-min						
	15-min						
	30-min						
	60-min						
	5-min						

Table 6 continued

Target variable	Time-frequency	Train/test = 0.7/0.3		Ideal-profit ratio		Train/test = 0.8/0.2		Ideal-profit ratio		Train/test = 0.9/0.1	
		Success ratio	Out-sample	In-sample	Out-sample	Success ratio	Out-sample	In-sample	Out-sample	Success ratio	Out-sample
4-Month futures	5-min	0.98	0.55	-0.157	0.54	0.98	0.54	-0.138	0.53	0.99	0.53
	10-min	0.98	0.53	-0.006	0.53	0.98	0.53	0.013	0.53	1.00	0.53
	15-min	1.00	0.54	-0.201	0.54	0.73	0.54	-0.126	0.54	0.71	0.54
	30-min	0.74	0.54	-0.057	0.55	0.70	0.55	-0.062	0.55	0.69	0.55
	60-min	0.80	0.56	0.135	0.56	0.79	0.56	-0.071	0.54	0.78	0.54
	5-min	0.99	0.62	0.042	0.62	0.98	0.62	0.061	0.67	0.98	0.67
5-Month futures	10-min	0.99	0.60	0.051	0.61	1.00	0.61	0.130	0.66	0.99	0.66
	15-min	0.99	0.60	0.130	0.61	0.99	0.61	0.013	0.67	0.99	0.67
	30-min	0.99	0.60	-0.022	0.61	1.00	0.61	0.009	0.66	1.00	0.66
	60-min	1.00	0.56	0.017	0.58	0.99	0.58	0.017	0.63	1.00	0.63
	mean	0.87***	0.55***	0.05	0.56***	0.83***	0.56***	0.06	0.56***	0.84***	0.56***
	t-test	14.38	11.25		10.42	12.57	10.42		6.59	12.27	6.59
std	0.13	0.02	0.10	0.03	0.13	0.03	0.10	0.05	0.14	0.05	
max	1.00	0.62	0.29	0.62	1.00	0.62	0.30	0.67	1.00	0.67	
min	0.68	0.53	-0.20	0.53	0.68	0.53	-0.14	0.52	0.68	0.52	

The t-statistics values are for the one-sided t-test

***, ** and * denote significant at the 1%, 5% and 10% level respectively

Table 7 Support Vector Machine (SVM) classification: in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	In-sample	In-sample	Out-sample	Out-sample	In-sample
1-Month future	5-min	0.53	0.59	0.020	0.52	0.57	0.026	0.52	0.58	0.049	0.52	0.58	0.049
	10-min	0.52	0.57	-0.023	0.53	0.56	-0.004	0.53	0.55	0.019	0.53	0.55	0.019
	15-min	0.55	0.57	0.023	0.54	0.57	0.028	0.54	0.57	0.136	0.54	0.57	0.136
	30-min	0.57	0.57	0.065	0.57	0.56	0.072	0.57	0.56	0.100	0.57	0.56	0.100
	60-min	0.56	0.56	0.021	0.56	0.55	0.004	0.52	0.54	0.122	0.52	0.54	0.122
	5-min	0.53	0.59	0.043	0.52	0.57	0.026	0.52	0.57	0.052	0.52	0.57	0.052
2-Month future	10-min	0.53	0.56	0.005	0.54	0.55	0.002	0.52	0.55	-0.009	0.52	0.55	-0.009
	15-min	0.54	0.56	0.032	0.54	0.55	0.045	0.55	0.55	0.114	0.55	0.55	0.114
	30-min	0.57	0.55	0.064	0.57	0.54	0.059	0.57	0.55	0.056	0.57	0.55	0.056
	60-min	0.56	0.55	0.119	0.56	0.55	0.074	0.56	0.54	0.108	0.56	0.54	0.108
	5-min	0.53	0.58	0.041	0.54	0.56	0.043	0.53	0.57	0.051	0.53	0.57	0.051
	10-min	0.53	0.55	0.057	0.53	0.55	0.075	0.54	0.56	0.054	0.54	0.56	0.054
3-Month future	15-min	0.55	0.56	0.095	0.55	0.56	0.126	0.55	0.56	0.101	0.55	0.56	0.101
	30-min	0.56	0.54	-0.036	0.57	0.53	-0.049	0.56	0.54	-0.226	0.56	0.54	-0.226
	60-min	0.56	0.57	0.296	0.55	0.58	0.312	0.56	0.55	0.322	0.56	0.55	0.322

Table 8 XGBOOST classification: in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
1-Month future	5-min	0.57	0.56	0.049	0.049	0.57	0.55	0.052	0.052	0.57	0.52	0.029	0.029
	10-min	0.58	0.50	-0.011	-0.011	0.58	0.51	-0.008	-0.008	0.58	0.49	-0.063	-0.063
	15-min	0.59	0.53	0.007	0.007	0.58	0.54	0.043	0.043	0.59	0.54	0.046	0.046
	30-min	0.62	0.56	0.063	0.063	0.61	0.56	0.094	0.094	0.61	0.55	0.055	0.055
	60-min	0.64	0.54	0.006	0.006	0.64	0.55	0.014	0.014	0.63	0.53	0.005	0.005
	5-min	0.58	0.56	0.076	0.076	0.58	0.54	0.065	0.065	0.58	0.54	0.059	0.059
2-Month future	10-min	0.58	0.52	0.014	0.014	0.58	0.51	-0.017	-0.017	0.57	0.52	-0.010	-0.010
	15-min	0.59	0.53	0.049	0.049	0.59	0.54	0.064	0.064	0.58	0.53	0.028	0.028
	30-min	0.61	0.55	0.105	0.105	0.60	0.55	0.126	0.126	0.60	0.55	0.085	0.085
	60-min	0.65	0.55	0.087	0.087	0.63	0.55	0.092	0.092	0.63	0.54	0.065	0.065
	5-min	0.57	0.54	0.150	0.150	0.58	0.54	0.167	0.167	0.57	0.54	0.083	0.083
	10-min	0.58	0.53	0.128	0.128	0.59	0.53	0.207	0.207	0.58	0.53	0.198	0.198
3-Month future	15-min	0.59	0.54	0.156	0.156	0.59	0.56	0.220	0.220	0.58	0.55	0.104	0.104
	30-min	0.61	0.56	0.170	0.170	0.61	0.55	0.166	0.166	0.60	0.55	0.053	0.053
	60-min	0.66	0.55	0.246	0.246	0.64	0.57	0.298	0.298	0.65	0.55	0.359	0.359

Table 8 continued

Target variable	Time-scale	Train/test = 0.7/0.3				Train/test = 0.8/0.2				Train/test = 0.9/0.1			
		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio		Success ratio		Ideal-profit ratio	
		In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
4-Month future	5-min	0.58	0.54	-0.026	0.53	0.58	0.53	0.029	0.52	0.58	0.52	0.115	0.115
	10-min	0.59	0.52	-0.216	0.52	0.59	0.52	-0.318	0.51	0.58	0.51	-0.422	-0.422
	15-min	0.59	0.53	-0.122	0.54	0.59	0.54	-0.139	0.53	0.59	0.53	-0.129	-0.129
	30-min	0.62	0.55	-0.052	0.55	0.61	0.55	-0.071	0.55	0.60	0.55	0.052	0.052
	60-min	0.63	0.55	0.054	0.54	0.64	0.54	-0.105	0.53	0.63	0.53	-0.124	-0.124
	5-Month future	5-min	0.66	0.58	0.087	0.59	0.66	0.59	0.113	0.63	0.65	0.63	0.067
5-Month future	10-min	0.67	0.56	0.063	0.58	0.66	0.58	-0.034	0.63	0.65	0.63	0.137	0.137
	15-min	0.67	0.55	0.097	0.58	0.66	0.58	0.115	0.63	0.65	0.63	0.228	0.228
	30-min	0.69	0.57	0.107	0.59	0.67	0.59	0.028	0.64	0.67	0.64	0.120	0.120
	60-min	0.69	0.53	-0.109	0.55	0.67	0.55	-0.275	0.61	0.67	0.61	-0.004	-0.004
	mean	0.62***	0.54***	0.05	0.55***	0.61***	0.55***	0.04	0.55***	0.61***	0.55***	0.05	0.05
	t-stat	14.75	13.25		11.63	16.46	11.63		6.22	16.51	6.22		
Std	0.04	0.02	0.10	0.02	0.03	0.02	0.14	0.04	0.03	0.04	0.14	0.14	
Max	0.69	0.58	0.25	0.59	0.67	0.59	0.30	0.64	0.67	0.64	0.36	0.36	
Min	0.57	0.50	-0.22	0.51	0.57	0.51	-0.32	0.49	0.57	0.49	-0.42	-0.42	

The t-statistics values are for the one-sided t-test

***, ** and * denote significant at the 1%, 5% and 10% level respectively

Table 9 Performance of Random walk and ARIMA time series forecasting over different maturities and different time scales

Random walk		Time Scale					
Target Variable	5-min	10-min	15-min	30-min	60-min		
1-Month futures	0.5002	0.5000	0.4997	0.4999	0.4996		
2-Month futures	0.5002	0.4999	0.4997	0.5000	0.4997		
3-Month futures	0.5002	0.4999	0.4996	0.5000	0.4996		
4-Month futures	0.5002	0.4998	0.4998	0.5000	0.4995		
5-Month futures	0.5001	0.4999	0.4998	0.5000	0.4997		
ARIMA model		Train/test = 0.7/0.3			Train/test = 0.8/0.2		Train/test = 0.9/0.1
		Success ratio		Success ratio		Success ratio	
Target variable	Time-scale	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
1-Month future	5-min	0.53	0.52	0.52	0.54	0.53	0.54
	10-min	0.51	0.50	0.51	0.50	0.51	0.50
	15-min	0.51	0.50	0.51	0.51	0.51	0.51
	30-min	0.52	0.52	0.52	0.52	0.52	0.53
	60-min	0.53	0.55	0.53	0.54	0.54	0.54
2-Month future	5-min	0.53	0.52	0.53	0.53	0.53	0.54
	10-min	0.53	0.52	0.53	0.52	0.53	0.52
	15-min	0.52	0.50	0.51	0.50	0.51	0.50
	30-min	0.54	0.50	0.53	0.50	0.52	0.53
	60-min	0.56	0.56	0.56	0.55	0.56	0.54

Table 9 continued

ARIMA model	Train/test = 0.7/0.3		Train/test = 0.8/0.2		Train/test = 0.9/0.1	
	Success ratio	Out-sample	Success ratio	Out-sample	Success ratio	Out-sample
Target variable	Time-scale	In-sample	In-sample	In-sample	In-sample	In-sample
3-Month future	5-min	0.53	0.53	0.53	0.53	0.55
	10-min	0.52	0.50	0.52	0.51	0.51
	15-min	0.52	0.51	0.52	0.51	0.51
	30-min	0.54	0.53	0.54	0.52	0.53
	60-min	0.56	0.54	0.55	0.55	0.55
	5-min	0.54	0.51	0.54	0.52	0.53
4-Month future	10-min	0.53	0.51	0.52	0.51	0.50
	15-min	0.52	0.52	0.52	0.51	0.51
	30-min	0.54	0.54	0.54	0.52	0.53
	60-min	0.55	0.56	0.56	0.53	0.51
	5-min	0.50	0.52	0.50	0.54	0.55
	10-min	0.50	0.53	0.50	0.54	0.54
5-Month future	15-min	0.51	0.54	0.51	0.54	0.51
	30-min	0.51	0.54	0.51	0.52	0.52
	60-min	0.51	0.55	0.51	0.55	0.55
	mean	0.53***	0.52***	0.53***	0.52***	0.52***
	t-stat	8.30	6.20	8.28	7.33	9.05
	Std	0.02	0.02	0.02	0.02	0.01
Max	0.56	0.56	0.56	0.55	0.55	
Min	0.50	0.50	0.50	0.50	0.51	

The t-statistics values are for the one-sided t-test

***, ** and * denote significant at the 1%, 5% and 10% level respectively

data divisions, the average success rate is around 56% with a maximum value of 67%, which is attained for 5-months futures at 5-min frequency. It is also observed that except in a few cases, one can gain positive ideal profit ratios with a maximum value of 36% for 3-month futures at a 60-min time scale.

Table 7 shows the best out-of-sample forecasting results across different maturities and frequencies is acquired from the implementation of a nonlinear classification method by choosing a radial kernel in the support vector machine. A maximum value of 61% is obtained for the first division, 64% for the second division, and 71% for the third division. The average out-sample success rate is stable, around 56% for different hold-outs. The results are also similar to the ideal profit ratios. It is evident from the results given in Table 8 that the XGBoost method provides similar results to the kNN algorithm but with a lower level of average in-sample fits. The average out-sample fit accuracy is around 55% for different hold-outs.

As a benchmark for our models, we also utilize the classical ARIMA model to predict the next time return direction of the mid-price. At first sight, one can argue that most of the methods provide close results to each other and the ARIMA model. For instance, while the average of the out-sample success rates is around 56% for the support vector machine, it is also around 52% for the ARIMA model, as can be seen from the results are presented in Table 9. For a small-sized sample, 1% of difference may not mean a significant difference in terms of the robustness of the method. However, in our case of a large sample, on average only 1% of increase in the success rate of 5-min frequency (0.7/0.1 hold-out) means 1,552 more correct predictions of the target variable. In our case, 4% of difference corresponds to 6,208 more accurate predictions of the target variable at 5 min level, which is not a substantive amount when considering the sample size analysed. We also have similar results for the other frequencies. Another critical point to note is that the maximum value of accuracy one can obtain with the ARIMA model is only 56% among all cases considered. However, this number even increases up to 71% for the support vector machine algorithm.

In Table 10, *t*-test is applied to check the statistical significance of the estimation results between alternative algorithms. We compare the methods using the success ratio results for each in-sample and out-of-sample period across the different maturities and time scales. For instance, to compare kNN and ARIMA models for the in-sample period, we use the success ratio values from the same 0.7/0.3 in-sample period across different maturities and time scales. We test the null hypothesis that the difference of the values computed from Logistic and ARIMA models comes from a normal distribution with a mean equal to zero and unknown variance. The results show that except for a few cases, the pairwise differences between these algorithms are statistically significant.

6 Conclusion

We examine the forecast performance of the midpoint movement of CME Bitcoin futures starting from 2 January 2020 to 10 September 2020 by using 1-, 2-, 3-, 4-, 5-month futures at 5-, 10-, 15-, 30-, 60-min frequencies. To this end, we employ machine learning algorithms (MLAs) which are compared with standard ARIMA and random walk models. Sign prediction ratios together with the ideal profit ratios are utilized to measure the forecasting performance of the suggested MLAs. Our findings indicate that the k-nearest neighbour (kNN) approach and the random forest (RF) algorithm yield the highest in- and out-of-sample accuracy rates at varying frequencies. For instance, for the random forest algorithm in-sample success rate can reach up to 87% for the first hold-out (0.7/0.3), 83% for the second hold-out (0.8/0.3), and

Table 10 *t*-test results for the mean of pairwise difference of seven classification algorithms, including ARIMA, kNN, logistic regression (Logistic), Naïve Bayes (NB), Random Forest (RF), Support Vector Machines (SVM), and XGBoost over different time train and test sets

	Train/test = 0.7/0.3		Train/test = 0.8/0.2		Train/test = 0.9/0.1	
	Success ratio		Success ratio		Success ratio	
	In-sample	Out-sample	In-sample	Out-sample	In-sample	Out-sample
kNN vs ARIMA	0.2447***	0.0307***	0.2199***	0.0286***	0.2266***	0.0398***
Logistic vs ARIMA	0.0147***	0.0151***	0.0123***	0.0146***	0.0106***	0.0074**
NB vs ARIMA	-0.0185***	-0.0717***	-0.0169***	-0.0586***	-0.019***	-0.0834***
RF vs ARIMA	0.3439***	0.0307***	0.3083***	0.0338***	0.3154***	0.0386***
SVM vs ARIMA	0.0243***	0.0395***	0.0235***	0.0342***	0.0198***	0.0386***
XGBoost vs ARIMA	0.0895***	0.0203***	0.0859***	0.0242***	0.0814***	0.0274***
kNN vs XGBoost	0.1552***	0.0104**	0.134***	0.2777	0.1452***	0.0124**
Logistic vs XGBoost	-0.0748***	0.1925	-0.0736***	0.0592*	-0.0708***	-0.0200**
NB vs XGBoost	-0.1080***	-0.092***	-0.11028***	-0.0828***	-0.1004***	-0.1108***
RF vs XGBoost	0.2544***	0.0104**	0.2224***	0.0096***	0.2340***	0.0112***
SVM vs XGBoost	-0.0652***	0.0192***	-0.0624***	0.1646	-0.0616***	0.3584
kNN vs SVM	0.2204***	0.2346	0.1964***	0.4332	0.2068***	0.9254
Logistic vs SVM	-0.0096**	-0.0244***	-0.0112***	-0.0196**	-0.0092**	-0.0312**
NB vs SVM	-0.0428***	-0.1112***	-0.0404***	-0.0928***	-0.0388***	-0.1220***
RF vs SVM	0.3196***	0.2243	0.2848***	0.9567	0.2956***	0.9999
kNN vs RF	-0.0992***	0.9999	-0.0884***	0.0790*	-0.0888***	0.7366
Logistic vs RF	-0.3292***	-0.0156**	-0.296***	-0.0192***	-0.3048***	-0.0312**
NB vs RF	-0.3624***	-0.1024***	-0.3252***	-0.0924***	-0.3344***	-0.122***
kNN vs NB	0.2632***	0.1024***	0.2368***	0.0872***	0.2456***	0.1232***
Logistic vs NB	0.0332***	0.0868***	0.0292***	0.0732***	0.0296***	0.0908***
kNN vs Logistic	0.2300***	0.0156**	0.2076***	0.0140**	0.2160***	0.0324***

***, ** and * denote significance at the 1%, 5% and 10% level respectively

87% for the third hold-out (0.9/0.1). However, the highest out-of-sample success rates are obtained by SVM. For instance, an accuracy rate of 71% can be achieved for the third hold-out. On the other hand, the logistic regression, the Naïve Bayes, and the XGBoost algorithm yield relatively stable results across different maturities with an average out-of-sample accuracy rate of 54%, 45%, and 55%, respectively. As a benchmark model, ARIMA model provides an average out-of-sample accuracy rate of around 52%. In general, our findings indicate that most of the MLAs outperform the benchmark model, with both in- and out-of-sample forecasting accuracy. This highlights the importance and relevance of MLAs to forecast bitcoin futures prices during periods of turmoil. From a policy perspective, such research has important implications regarding the monitoring of cryptocurrency market developments, and of course, from a regulatory perspective, signals of market discontinuity could demonstrate the sources of atypical influence. Furthermore, it is a known fact that derivative products complete the market and increases efficiency. Hence, as the cryptocurrency derivatives market enlarges, machine learning algorithms can also be used for the prediction and valuation of the products such as options and swaps in the future.

Acknowledgements Gazi Salah Uddin is thankful for the Visiting Fellow Programme Grant provided by the University of Jyväskylä, Finland.

References

- Adcock, R., & Gradojevic, N. (2019). Non-fundamental, non-parametric bitcoin forecasting. *Physica A: Statistical Mechanics and its Applications* 531.
- Akhtaruzzaman, M., Boubaker, S., & Sensoy, A. (2020). Financial contagion during covid-19 crisis. *Finance Research Letters*, 101604.
- Akyildirim, E., Corbet, S., Cumming, D., Lucey, B., & Sensoy, A. (2020). Riding the wave of crypto-exuberance: The potential misuse of corporate blockchain announcements. *Technological Forecasting and Social Change*, 159, 120191.
- Akyildirim, E., Corbet, S., Katsiampa, P., Kellard, N., & Sensoy, A. (2020). The development of bitcoin futures: Exploring the interactions between cryptocurrency derivatives. *Finance Research Letters*, 34, 101234.
- Akyildirim, E., Corbet, S., Lucey, B., Sensoy, A., & Yarovaya, L. (2020). The relationship between implied volatility and cryptocurrency returns. *Finance Research Letters*, 33, 101202.
- Akyildirim, E., Corbet, S., Sensoy, A., & Yarovaya, L. (2020). The impact of blockchain related name changes on corporate performance. *Journal of Corporate Finance*, 101759.
- Akyildirim, E., Goncu, A., & Sensoy, A. (2020). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*.
- Alameer, Z., Fathalla, A., Li, K., Ye, H., & Jianhua, Z. (2020). Multistep-ahead forecasting of coal prices using a hybrid deep learning model. *Resources Policy* 65.
- Alexander, C., Choi, J., Park, H., & Sohn, S. (2020). Bitmex bitcoin derivatives: Price discovery, informational efficiency, and hedging effectiveness. *Journal of Futures Markets*, 40(1), 23–43.
- Alonso-Monsalve, S., Suárez-Cetrulo, A., Cervantes, A., Quintana, D. (2020). Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. *Expert Systems with Applications* 149.
- Atsalakis, G., Atsalaki, I., Pasiouras, F., & Zopounidis, C. (2019). Bitcoin price forecasting with neuro-fuzzy techniques. *European Journal of Operational Research*, 276(2), 770–780.
- Auld, T., & Linton, O. (2019). The behaviour of betting and currency markets on the night of the eu referendum. *International Journal of Forecasting*, 35(1), 371–389.
- Breiman, L. (2001). Machine learning. *Random Forests*, 45(1), 5–32.
- Chai, J., Xing, L.-M., Zhou, X.-Y., Zhang, Z., & Li, J.-X. (2018). Forecasting the wti crude oil price by a hybrid-refined method. *Energy Economics*, 71, 114–127.
- Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, 37(1), 28–43.

- Chen, Y., Nasrabadi, N., & Tran, T. (2013). Hyperspectral image classification via kernel sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1), 217–231.
- Chen, Y., Xie, X., Zhang, T., Bai, J., & Hou, M. (2020). A deep residual compensation extreme learning machine and applications. *Journal of Forecasting*, 39(6), 986–999.
- Chowdhury, R., Rahman, M., Rahman, M., & Mahdy, M. (2020). An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning. *Physica A: Statistical Mechanics and its Applications* 551.
- Conlon, T., Corbet, S., & McGee, R. J. (2020). Are cryptocurrencies a safe haven for equity markets? An international perspective from the covid-19 pandemic. *Research in International Business and Finance*, 101248.
- Corbet, S., Cumming, D., Lucey, B., Peat, M., & Vigne, S. (2020). The destabilising effects of cryptocurrency cybercriminality. *Economics Letters* 191.
- Corbet, S., Hou, Y., Hu, Y., Larkin, C., & Oxley, L. (2020). Any port in a storm: Cryptocurrency safe-havens during the covid-19 pandemic. *Economics Letters* 194.
- Corbet, S., Hou, Y., Hu, Y., & Oxley, L. (2020). The influence of the covid-19 pandemic on asset-price discovery: Testing the case of chinese informational asymmetry. *International Review of Financial Analysis* 72.
- Corbet, S., Hou, Y., Hu, Y., Oxley, L., & Xu, D. (2021). Pandemic-related financial market volatility spillovers: Evidence from the chinese covid-19 epicentre. *International Review of Economics and Finance*, 71, 55–81.
- Corbet, S., Larkin, C., & Lucey, B. (2020). The contagion effects of the covid-19 pandemic: Evidence from gold and cryptocurrencies. *Finance Research Letters*, 101554.
- Corbet, S., Larkin, C., Lucey, B., Meegan, A., & Yarovaya, L. (2020). The impact of macroeconomic news on bitcoin returns. *European Journal of Finance*.
- Corbet, S., Lucey, B., Peat, M., & Vigne, S. (2018a). Bitcoin futures-what use are they? *Economics Letters*, 172, 23–27.
- Corbet, S., Lucey, B., Peat, M., & Vigne, S. (2018b). What sort of asset? Bitcoin analysed. In *International workshop on enterprise applications, markets and services in the finance industry*, pp. 52–65. Springer.
- Corbet, S., Lucey, B., Urquhart, A., & Yarovaya, L. (2019). Cryptocurrencies as a financial asset: A systematic analysis. *International Review of Financial Analysis*, 62, 182–199.
- de la Horra, L., de la Fuente, G., & Perote, J. (2019). The drivers of bitcoin demand: A short and long-run analysis. *International Review of Financial Analysis*, 62, 21–34.
- De Spiegeleer, J., Madan, D., Reyner, S., & Schoutens, W. (2018). Machine learning for quantitative finance: Fast derivative pricing, hedging and fitting. *Quantitative Finance*, 18(10), 1635–1643.
- El Shazly, M., & El Shazly, H. (1999). Forecasting currency prices using a genetically evolved neural network architecture. *International Review of Financial Analysis*, 8(1), 67–82.
- Faghih Mohammadi Jalali, M., & Heidari, H. (2020). Predicting changes in bitcoin price using grey system theory. *Financial Innovation* 6(1).
- 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*, 112, 245–253.
- Fang, Y., Guan, B., Wu, S., & Heravi, S. (2020). Optimal forecast combination based on ensemble empirical mode decomposition for agricultural commodity futures prices. *Journal of Forecasting*, 39(6), 877–886.
- Fauvel, M., Benediktsson, J., Chanussot, J., & Sveinsson, J. (2008). Spectral and spatial classification of hyperspectral data using svms and morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 46(11), 3804–3814.
- 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.
- Ftiti, Z., Tissaoui, K., & Boubaker, S. (2020). On the relationship between oil and gas markets: a new forecasting framework based on a machine learning approach. *Annals of Operations Research*.
- Ghoddusi, H., Creamer, G., & Rafizadeh, N. (2019). Machine learning in energy economics and finance: A review. *Energy Economics*, 81, 709–727.
- Giudici, P., & Polinesi, G. (2019). Crypto price discovery through correlation networks. *Annals of Operations Research*.
- Goodell, J. W. (2020). Covid-19 and finance: Agendas for future research. *Finance Research Letters*, 101512.
- Hu, Y., Hou, Y., Oxley, L., & Corbet, S. (2020). Does blockchain patent-development influence bitcoin risk? *Journal of International Financial Markets, Institutions and Money*.
- Huang, J.-Y., & Liu, J.-H. (2020). Using social media mining technology to improve stock price forecast accuracy. *Journal of Forecasting*, 39(1), 104–116.
- Hubáček, O., Šourek, G., & Železný, F. (2019). Exploiting sports-betting market using machine learning. *International Journal of Forecasting*, 35(2), 783–796.

- Huck, N. (2019). Large data sets and machine learning: Applications to statistical arbitrage. *European Journal of Operational Research*, 278(1), 330–342.
- Katsiampa, P., Corbet, S., & Lucey, B. (2019a). High frequency volatility co-movements in cryptocurrency markets. *Journal of International Financial Markets, Institutions and Money*, 62, 35–52.
- Katsiampa, P., Corbet, S., & Lucey, B. (2019b). Volatility spillover effects in leading cryptocurrencies: A bekk-mgarch analysis. *Finance Research Letters*, 29, 68–74.
- 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*, 283(1), 217–234.
- Koutmos, D. (2020). Market risk and bitcoin returns. *Annals of Operations Research*, 294(1–2), 453–477.
- Lajbcygier, P. (2004). Improving option pricing with the product constrained hybrid neural network. *IEEE Transactions on Neural Networks*, 15(2), 465–476.
- Ma, F., Liang, C., Ma, Y., & Wahab, M. (2020). Cryptocurrency volatility forecasting: A markov regime-switching midas approach. *Journal of Forecasting*, 39(8), 1277–1290.
- Matyjaszek, M., Riesgo Fernández, P., Krzemień, A., Wodarski, K., & Fidalgo Valverde, G. (2019). Forecasting coking coal prices by means of arima models and neural networks, considering the transgenic time series theory. *Resources Policy*, 61, 283–292.
- Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8), 1778–1790.
- Mensi, W., Sensoy, A., Vo, X. V., & Kang, S. H. (2020). Impact of covid-19 outbreak on asymmetric multifractality of gold and oil prices. *Resources Policy*, 69, 101829.
- Munder, S., & Gavriila, D. (2006). An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11), 1863–1868.
- Nakano, M., Takahashi, A., & Takahashi, S. (2018). Bitcoin technical trading with artificial neural network. *Physica A: Statistical Mechanics and its Applications*, 510, 587–609.
- Ntakaris, A., Magris, M., Kannianen, J., Gabbouj, M., & Iosifidis, A. (2018). Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 37(8), 852–866.
- Philip, R. (2020). Estimating permanent price impact via machine learning. *Journal of Econometrics*, 215(2), 414–449.
- Plakandaras, V., Papadimitriou, T., & Gogas, P. (2015). Forecasting daily and monthly exchange rates with machine learning techniques. *Journal of Forecasting*, 34(7), 560–573.
- Saad, M., Choi, J., Nyang, D., Kim, J., & Mohaisen, A. (2020). Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions. *IEEE Systems Journal*, 14(1), 321–332.
- Samitas, A., Kampouris, E., & Kenourgios, D. (2020). Machine learning as an early warning system to predict financial crisis. *International Review of Financial Analysis* 71.
- Sensoy, A. (2019). The inefficiency of bitcoin revisited: A high-frequency analysis with alternative currencies. *Finance Research Letters*, 28, 68–73.
- Sharif, A., Aloui, C., & Yarovaya, L. (2020). Covid-19 pandemic, oil prices, stock market, geopolitical risk and policy uncertainty nexus in the us economy: Fresh evidence from the wavelet-based approach. *International Review of Financial Analysis*, 101496.
- Shu, M., & Zhu, W. (2020). Real-time prediction of bitcoin bubble crashes. *Physica A: Statistical Mechanics and its Applications* 548.
- Sirignano, J. (2019). Deep learning for limit order books. *Quantitative Finance*, 19(4), 549–570.
- Sirignano, J., & Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, 19(9), 1449–1459.
- Suykens, J., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3), 293–300.
- Sánchez Lasheras, F., de Cos Juez, F., Suárez Sánchez, A., Krzemień, A., & Riesgo Fernández, P. (2015). Forecasting the comex copper spot price by means of neural networks and arima models. *Resources Policy*, 45, 37–43.
- Wang, M., Tian, L., & Zhou, P. (2018). A novel approach for oil price forecasting based on data fluctuation network. *Energy Economics*, 71, 201–212.
- Weldegebriel, H., Liu, H., Haq, A., Buggingo, E., & Zhang, D. (2020). A new hybrid convolutional neural network and extreme gradient boosting classifier for recognizing handwritten ethiopian characters. *IEEE Access*, 8, 17804–17818.
- Wittkemper, H.-G., & Steiner, M. (1996). Using neural networks to forecast the systematic risk of stocks. *European Journal of Operational Research*, 90(3), 577–588.

- Yarovaya, L., Mirza, N., Abaidi, J., & Hasnaoui, A. (2020). Human capital efficiency and equity funds' performance during the covid-19 pandemic. *International Review of Economics & Finance*, 71, 584–591.
- Zhang, G., Eddy Patuwo, B., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1), 35–62.
- Zhang, K., Zhang, L., & Yang, M.-H. (2014). Fast compressive tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10), 2002–2015.
- Zimmermann, H., Neuneier, R., & Grothmann, R. (2001). Multiagent modeling of multiple fx-markets by neural networks. *IEEE Transactions on Neural Networks*, 12(4), 735–743.
- Zoumpikas, T., Houstis, E., & Vavalis, M. (2020). Eth analysis and predictions utilizing deep learning. *Expert Systems with Applications* 162.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.