Copenhagen Business School
HANDELSHØJSKOLEN

Department of Economics

Copenhagen Business School

Working paper 20-2020

Forecasting Mid-Price Movement of Bitcoin Futures Using Machine Learning

Erdinc Akyildirim
Oguzhan Cepni
Shaen Corbet
Gazi Salah Uddin

Department of Economics – Porcelænshaven 16A, 1. DK-2000 Frederiksberg

Forecasting Mid-Price Movement of Bitcoin Futures Using Machine Learning

Erdinc Akyildirim[a,b,c*], Oguzhan Cepni[d,e], Shaen Corbet[f,g], Gazi Salah Uddin[h]

[a] *Department of Mathematics, ETH, Zurich, Switzerland*

[b] *University of Zurich, Department of Banking and Finance, Zurich, Switzerland*

[c] *Department of Banking and Finance, Burdur Mehmet Akif Ersoy University, Burdur, Turkey*

[d] *Copenhagen Business School, Department of Economics, Porcelænshaven 16A, Frederiksberg DK-2000, Denmark*

[e] *Central Bank of the Republic of Turkey, Haci Bayram Mah. Istiklal Cad. No:10 06050, Ankara, Turkey*

[f] *DCU Business School, Dublin City University, Dublin 9, Ireland*

[g] *School of Accounting, Finance and Economics, University of Waikato, New Zealand*

[h] *Department of Management and Engineering, Linköping University, 581 83 Linköping, Sweden*

[*] *Corresponding Author: erdinc.akyildirim@math.ethz.ch*

Abstract

In the aftermath of the global financial crisis and on-going COVID-19, investors face challenges in understanding price dynamics across assets. In this paper, we explore the applicability of a large scale comparison of machine learning algorithms (MLA) to predict mid-price movement for bitcoin futures prices. We use high-frequency intra-day data to evaluate the relative forecasting performances across various time-frequencies, ranging between 5-minutes and 60-minutes. The empirical analysis is based on six different specifications of MLA methods during periods of pandemic. The empirical results show that MLA outperforms the random walk and ARIMA forecasts in Bitcoin futures markets, which may have important implications in the decision-making process of predictability.

Keywords: Cryptocurrency; Bitcoin Futures, Machine Learning; Covid-19; k-Nearest Neighbors; Logistic Regression; Naive Bayes; Random Forest; Support Vector Machine; Extreme Gradient Boosting

---

1. Introduction

In this research, we use high frequency Bitcoin pricing data, and by using machine learning algorithms, we attempt to predict mid price movement for bitcoin futures price series across a variety of time-frequencies, ranging between 5-minutes and 60-minutes. The novelty of our research surrounds the 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 a number of 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, whether designed explicitly [Corbet and Cumming, 2020], or implicitly [Akyildirim et al., 2020] that have been present in recent years. Each of these steps had provided additional evidence of a developing and maturing product, to which we must examine as to whether predictability, as identified across a number of other spot currency markets can also be identified through similar futures market prices [Akyildirim 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 previous identified across several assets such as foreign exchange markets [Plakandaras et al., 2015] and a number of 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

[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.

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.

We contribute to the literature by evaluating the application of high frequency bitcoin future prices using six machine learning algorithms during to 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 indicates 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 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. This indicates that COVID-19 may become a new source of suitability of machine learning algorithms. 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 Section 2. Section 3 presents a thorough explanation of the wide variety of data used in our analyses, while Section 4 presents a concise overview of the methodologies utilised. Section 5 presents a concise overview of the results and their relevance for policy-makers and regulatory authorities, while Section 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 a number of 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]. Ghoddusi 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 analyzed 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 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 in order 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,b] and Hu et al. [2020].

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 time 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. specially in the Bitcoin, Ether and Litecoin cryptocurrencies. 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 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. Zoumpekas 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 prediction 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] 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.

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 minutely level which have the earliest starting points at different dates. In order to have enough number of observations to draw meaningful and robust conclusions we use only 1-month to 5-month 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 the day at CME after 23:00 PM on Sunday till 22:00 PM on Friday (because of 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). 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-minute frequency during 2020. For instance, while there are 4345 observations at the 60-min frequency, this number increases to 51733 at 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.

4. Classification Algorithms

4.1. Machine learning models

We apply six different machine learning algorithms (k-Nearest Neighbors, 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 Neighbors Classifier

The k-nearest neighbors algorithm (kNN) is a commonly used, simple yet successful classification method which has been applied in a large number of classification and regression problems such as handwritten digits and satellite image scenes. The kNN is a supervised machine learning model where the model learns from the labeled 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 labeled ‘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 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. 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 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:

$$P\widehat{P}(Y = k \mid X_1, ..., X_P) = \frac{\pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)}{\sum_{k=1}^{K} \pi(Y = k) \prod_{j=1}^{P} P(X_j \mid Y = k)} \tag{1}$$

where Y is the random variable corresponding to the class index of an observation, $X_1, ..., X_P$ 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. In logistic regression, the outcome of 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 features 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. $x = (x_1, ..., x_d)$, and parameter vector w, the logistic regression with the penalty term minimizes the following optimization problem:

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

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 individual 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 $(x_1, x_2, ..., x_5)$ with corresponding labels $(y_1, y_2, ..., y_5)$ 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 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 $\theta_1, ..., \theta_{k-1}$ 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 a number of 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 $h(x, \theta_k), k = 1, ...$ 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. 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 $i = 1, 2, ..., N$ 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 $\xi_i \geq 0, i = 1, 2, ..., N$. 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 $i = 1, 2, ..., N$, 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 $K(x_i, x_j) = \phi(x_i)^T \phi(x)$ 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

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

The optimization problem in Equation 3 can be solved globally using the Karush-Kuhn-Tucker (KKT) conditions. Clearly, 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 in order to produce one optimal predictive model. 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 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

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

where

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

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)

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

and represents the maximum achievable return assuming perfect forecast.

- The Total Return is computed in the following way

$$TotalReturn = \sum_{j=1}^{M} sign(Y_j') * h_j \tag{9}$$

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).

$$IPR = \frac{TotalReturn}{MaxReturn} \tag{10}$$

5. Empirical Results

As explained in Section 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 these classification methods is the sign of the mid-price return of bitcoin futures with different maturities at different frequencies. We consider three different divisions of the dataset as train and test sets, which are also called the 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.

Table 1 shows the number of observations for the mid-price returns of Bitcoin futures at different time frequencies for different train/test set divisions. For instance, while there are 51,733 mid-price returns in total at the 5-minute scale, there are only 4,345 mid-price returns at the 60-minute scale. 0.7/0.3 hold-out at 5-minute 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. Table 2 presents the descriptive statistics for mid-price returns of Bitcoin futures at different time scales with different maturities. As it is clear from the Table 2, 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 one time scale to another within the same Bitcoin future (except the 1-month futures). As expected, standard deviations increase both with time to maturity and time-frequency within the same futures.

As we described in the previous section, we employ two key metrics to measure the performance of the different machine learning algorithms. First of all, 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 fully random then the correct sign prediction ratio would be 50%. However, in our case, it is important 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. In addition to the sign prediction ratio, we also apply the ideal profit ratio to measure the performance of the related classification algorithm. As it is formalized in Section 4, the ideal profit ratio is the ratio of the return generated by a given algorithm to the perfect sign forecast.

Table 3-8 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 neighborhood 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. It is also possible to observe that it is almost always the case that as the time-frequency decreases, the accuracy rate increases for the same maturity futures under the kNN method. It is also evident from Table 3 that for most of the cases, it is possible to 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 minutes 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 one can observe that for most of the cases it is possible to attain a positive ideal profit ratio with this prediction method. 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. Although the max accuracy rate can reach even up to 60% for 1-month futures at 5 minutes frequency for the first hold-out, the average accuracy rate is only 45% across the different cases. This result also holds true for 0.8/0.2 and 0.9/0.1 divisions. We also have similar results for the ideal profit ratios. We receive 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 lower than the in-sample fits. This designates the high variance in the random forest classification with high in sample fit to the noisy data but lower out-of-sample performance. For all of the divisions of data, the average success rate is around 56% with a maximum value of 67%, which is attained for 5-months futures at 5- minutes 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.

When we apply a nonlinear classification method by choosing radial kernel in the support vector machine, Table 7 shows the best out-of-sample forecasting results across different maturities and frequencies. 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 for the ideal profit ratios. It is evident from the results given in Table 8 that XGBoost method provides very close 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, where results are presented in Table 9. For a small sized sample, 1% of difference may not mean 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-minute frequency (0.7/0.1 hold-out) means 1,552 more correct prediction of the target variable. In our case, 4% of difference corresponds to 6,208 more accurate predictions of the target variable at 5 minutes level, which is not a negligible number. 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.

6. Conclusion

We examine the forecast performance of the midpoint movement of CME Bitcoin futures prices ranging from 2nd January 2020 to 10th September 2020 across a variety of time-frequencies, varying between 5-minutes and 60-minutes, by utilizing various machine learning algorithms (MLA). The core objective of this work is to evaluate whether the more advanced MLAs are able to outperform a simple parsimonious framework during periods of turmoil. To compare the forecast accuracy of the employed MLA frameworks, we utilize ARIMA model as a benchmark for our models. Furthermore, we utilize the sign prediction rate or accuracy rate and ideal profit ratio to evaluate the forecasting accuracy of the employed frameworks. In addition, we evaluate the robustness of our obtained estimates by varying the train and test (hold-out) sample.

Our empirical analysis is as follows. Firstly, our findings indicate that the k-nearest neighbor (kNN) approach and the random forest algorithm yields the highest in- and out-of-sample accuracy rate at varying frequencies. Among these MLAs, the random forest algorithm provides the highest in-sample success rate that can reach up to 87% for the 0.3 hold-out sample, 83% for the 0.s hold-out sample, and 87% for the 0.1 hold-out sample. However, the out-of-sample average performance is significantly lower than the in-sample fits, designated to the high variance in the random forest classification with high in-sample fit to the noisy data but lower out-of-sample performance. The average out-of-sample performance of the random forest algorithm provides an accuracy rate of around 56%. Secondly, the logistic regression algorithm, the Naïve Bayes classification approach, the nonlinear classification approach, and the XGBoost algorithm yields a relatively stable results across different maturities with an average out-of-sample accuracy rate of 54%, 45%, 56%, and 55%, respectively. Thirdly, the benchmark model (ARIMA) provides an average out-of-sample accuracy rate of around 52%. In general, our findings indicate that most of the MLA significantly outperforms a simple parsimonious model, both in- and out-of-sample forecasting accuracy. This highlights the importance and relevancy of MLAs to forecast bitcoin futures prices during periods of turmoil.

References

Adcock, R. and N. Gradojevic (2019). Non-fundamental, non-parametric bitcoin forecasting. *Physica A: Statistical Mechanics and its Applications 531*.

Akhtaruzzaman, M., S. Boubaker, and A. Sensoy (2020). Financial contagion during covid–19 crisis. *Finance Research Letters*, 101604.

Akyildirim, E., S. Corbet, D. Cumming, B. Lucey, and A. Sensoy (2020). Riding the wave of crypto-exuberance: The potential misusage of corporate blockchain announcements. *Technological Forecasting and Social Change 159*, 120191.

Akyildirim, E., S. Corbet, P. Katsiampa, N. Kellard, and A. Sensoy (2020). The development of bitcoin futures: Exploring the interactions between cryptocurrency derivatives. *Finance Research Letters 34*, 101234.

Akyildirim, E., S. Corbet, B. Lucey, A. Sensoy, and L. Yarovaya (2020). The relationship between implied volatility and cryptocurrency returns. *Finance Research Letters 33*, 101212.

Akyildirim, E., S. Corbet, A. Sensoy, and L. Yarovaya (2020). The impact of blockchain related name changes on corporate performance. *Journal of Corporate Finance*, 101759.

Akyildirim, E., A. Goncu, and A. Sensoy (2020). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*.

Alameer, Z., A. Fathalla, K. Li, H. Ye, and Z. Jianhua (2020). Multistep-ahead forecasting of coal prices using a hybrid deep learning model. *Resources Policy 65*.

Alexander, C., J. Choi, H. Park, and S. Sohn (2020). Bitmex bitcoin derivatives: Price discovery, informational efficiency, and hedging effectiveness. *Journal of Futures Markets 40*(1), 23–43.

Alonso-Monsalve, S., A. Suárez-Cetrulo, A. Cervantes, and D. Quintana (2020). Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. *Expert Systems with Applications 149*.

Atsalakis, G., I. Atsalaki, F. Pasiouras, and C. Zopounidis (2019). Bitcoin price forecasting with neuro-fuzzy techniques. *European Journal of Operational Research 276*(2), 770–780.

Auld, T. and O. Linton (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). Random forests. *Machine learning 45*(1), 5–32.

Chai, J., L.-M. Xing, X.-Y. Zhou, Z. Zhang, and J.-X. Li (2018). Forecasting the wti crude oil price by a hybrid-refined method. *Energy Economics 71*, 114–127.

Chen, W., H. Xu, L. Jia, and Y. Gao (2021). Machine learning model for bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting 37*(1), 28–43.

Chen, Y., X. Xie, T. Zhang, J. Bai, and M. Hou (2020). A deep residual compensation extreme learning machine and applications. *Journal of Forecasting 39*(6), 986–999.

Chowdhury, R., M. Rahman, M. Rahman, and M. Mahdy (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., S. Corbet, and R. J. McGee (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., D. Cumming, B. Lucey, M. Peat, and S. Vigne (2020). The destabilising effects of cryptocurrency cybercriminality. *Economics Letters 191*.

Corbet, S. and D. J. Cumming (2020). *The Wild West of ICOs*, pp. 113–130. Berlin, Boston: De Gruyter.

Corbet, S., Y. Hou, Y. Hu, C. Larkin, and L. Oxley (2020). Any port in a storm: Cryptocurrency safe-havens during the covid-19 pandemic. *Economics Letters 194*.

Corbet, S., Y. Hou, Y. Hu, and L. Oxley (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., Y. Hou, Y. Hu, L. Oxley, and D. Xu (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., C. Larkin, and B. Lucey (2020). The contagion effects of the covid-19 pandemic: Evidence from gold and cryptocurrencies. *Finance Research Letters*, 101554.

Corbet, S., C. Larkin, B. Lucey, A. Meegan, and L. Yarovaya (2020). The impact of macroeconomic news on bitcoin returns. *European Journal of Finance*.

Corbet, S., B. Lucey, M. Peat, and S. Vigne (2018a). Bitcoin futures—what use are they? *Economics Letters 172*, 23–27.

Corbet, S., B. Lucey, M. Peat, and S. Vigne (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., B. Lucey, A. Urquhart, and L. Yarovaya (2019). Cryptocurrencies as a financial asset: A systematic analysis. *International Review of Financial Analysis 62*, 182–199.

de la Horra, L., G. de la Fuente, and J. Perote (2019). The drivers of bitcoin demand: A short and long-run analysis. *International Review of Financial Analysis 62*, 21–34.

De Spiegeleer, J., D. Madan, S. Reyners, and W. Schoutens (2018). Machine learning for quantitative finance: fast derivative pricing, hedging and fitting. *Quantitative Finance 18*(10), 1635–1643.

El Shazly, M. and H. El Shazly (1999). Forecasting currency prices using a genetically evolved neural network architecture. *International Review of Financial Analysis 8*(1), 67–82.

Faghih Mohammadi Jalali, M. and H. Heidari (2020). Predicting changes in bitcoin price using grey system theory. *Financial Innovation 6*(1).

Fan, L., S. Pan, Z. Li, and H. Li (2016). An ica-based support vector regression scheme for forecasting crude oil prices. *Technological Forecasting and Social Change 112*, 245–253.

Fang, Y., B. Guan, S. Wu, and S. Heravi (2020). Optimal forecast combination based on ensemble empirical mode decomposition for agricultural commodity futures prices. *Journal of Forecasting 39*(6), 877–886.

Ftiti, Z., K. Tissaoui, and S. Boubaker (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., G. Creamer, and N. Rafizadeh (2019). Machine learning in energy economics and finance: A review. *Energy Economics 81*, 709–727.

Giudici, P. and G. Polinesi (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., Y. Hou, L. Oxley, and S. Corbet (2020). Does blockchain patent-development influence bitcoin risk? *Journal of International Financial Markets, Institutions and Money*.

Huang, J.-Y. and J.-H. Liu (2020). Using social media mining technology to improve stock price forecast accuracy. *Journal of Forecasting 39*(1), 104–116.

Hubáček, O., G. Šourek, and F. Železný (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., S. Corbet, and B. Lucey (2019a). High frequency volatility co-movements in cryptocurrency markets. *Journal of International Financial Markets, Institutions and Money 62*, 35–52.

Katsiampa, P., S. Corbet, and B. Lucey (2019b). Volatility spillover effects in leading cryptocurrencies: A bekk-mgarch analysis. *Finance Research Letters 29*, 68–74.

Kim, A., Y. Yang, S. Lessmann, T. Ma, M.-C. Sung, and J. Johnson (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., C. Liang, Y. Ma, and M. Wahab (2020). Cryptocurrency volatility forecasting: A markov regime-switching midas approach. *Journal of Forecasting 39*(8), 1277–1290.

Matyjaszek, M., P. Riesgo Fernández, A. Krzemień, K. Wodarski, and G. Fidalgo Valverde (2019). Forecasting coking coal prices by means of arima models and neural networks, considering the transgenic time series theory. *Resources Policy 61*, 283–292.

Mensi, W., A. Sensoy, X. V. Vo, and S. H. Kang (2020). Impact of covid-19 outbreak on asymmetric multifractality of gold and oil prices. *Resources Policy 69*, 101829.

Nakano, M., A. Takahashi, and S. Takahashi (2018). Bitcoin technical trading with artificial neural network. *Physica A: Statistical Mechanics and its Applications 510*, 587–609.

Ntakaris, A., M. Magris, J. Kanniainen, M. Gabbouj, and A. Iosifidis (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., T. Papadimitriou, and P. Gogas (2015). Forecasting daily and monthly exchange rates with machine learning techniques. *Journal of Forecasting 34*(7), 560–573.

Saad, M., J. Choi, D. Nyang, J. Kim, and A. Mohaisen (2020). Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions. *IEEE Systems Journal 14*(1), 321–332.

Samitas, A., E. Kampouris, and D. Kenourgios (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., C. Aloui, and L. Yarovaya (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. and W. Zhu (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. and R. Cont (2019). Universal features of price formation in financial markets: perspectives from deep learning. *Quantitative Finance 19*(9), 1449–1459.

Sánchez Lasheras, F., F. de Cos Juez, A. Suárez Sánchez, A. Krzemień, and P. Riesgo Fernández (2015). Forecasting the comex copper spot price by means of neural networks and arima models. *Resources Policy 45*, 37–43.

Wang, M., L. Tian, and P. Zhou (2018). A novel approach for oil price forecasting based on data fluctuation network. *Energy Economics 71*, 201–212.

Wittkemper, H.-G. and M. Steiner (1996). Using neural networks to forecast the systematic risk of stocks. *European Journal of Operational Research 90*(3), 577–588.

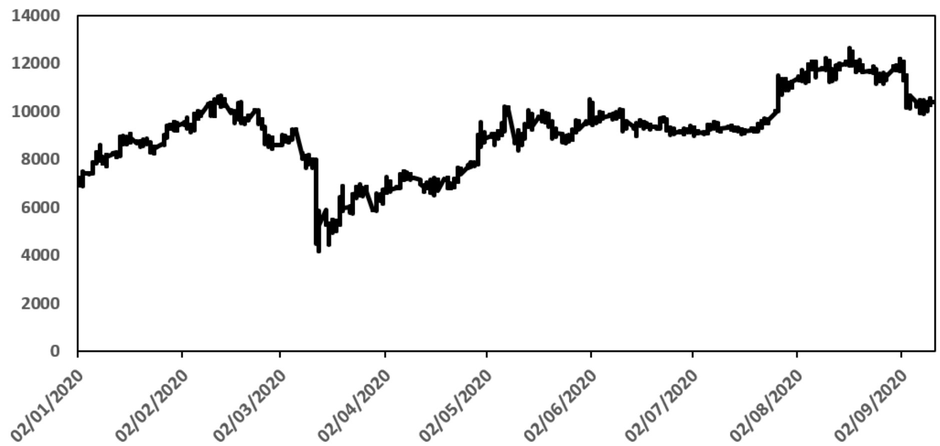
Yarovaya, L., N. Mirza, J. Abaidi, and A. Hasnaoui (2020). Human capital efficiency and equity funds' performance during the covid-19 pandemic. *International Review of Economics & Finance 71*, 584–591.

Zhang, G., B. Eddy Patuwo, and M. Y. Hu (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting 14*(1), 35–62.

Zimmermann, H., R. Neuneier, and R. Grothmann (2001). Multiagent modeling of multiple fx-markets by neural networks. *IEEE Transactions on Neural Networks 12*(4), 735–743.

Zoumpekas, T., E. Houstis, and M. Vavalis (2020). Eth analysis and predictions utilizing deep learning. *Expert Systems with Applications 162*.

Figure 1: 1-month futures mid-price at 5-minute frequency



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

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

		5-min	10-min	15-min	30-min	60-min
	total	51,733	25,885	17,269	8,653	4,345
0.7/0.3	train	36,213	18,119	12,088	6,057	3,041
	test	15,520	7,766	5,181	2,596	1,304
0.8/0.2	train	41,386	20,708	13,815	6,922	3,476
	test	10,347	5,177	3,454	1,731	869
0.9/0.1	train	46,559	23,296	15,542	7,787	3,910
	test	5,174	2,589	1,727	866	435

Note: 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.

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

1-month futures	5-min	10-min	15-min	30-min	60-min
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
Std.Dev.	0.003	0.005	0.006	0.008	0.011
2-month futures	5-min	10-min	15-min	30-min	60-min
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
Std.Dev.	0.005	0.007	0.008	0.012	0.016
3-month futures	5-min	10-min	15-min	30-min	60-min
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
Std.Dev.	0.007	0.010	0.012	0.017	0.024
4-month futures	5-min	10-min	15-min	30-min	60-min
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
Std.Dev.	0.010	0.014	0.017	0.023	0.033
5-month futures	5-min	10-min	15-min	30-min	60-min
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
Std.Dev.	0.018	0.024	0.028	0.038	0.054

Note: 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.

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.

		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
target variable	time-scale	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample
1-month future	5-min	0.86	0.56	-0.002	0.86	0.55	0.009	0.86	0.54	0.033
	10-min	0.83	0.54	0.004	0.83	0.55	0.024	0.69	0.56	0.091
	15-min	0.81	0.54	-0.004	0.82	0.53	0.016	0.66	0.54	0.100
	30-min	0.66	0.54	0.074	0.66	0.54	0.022	0.62	0.57	0.090
	60-min	0.62	0.53	0.065	0.62	0.53	0.083	0.64	0.52	0.032
2-month future	5-min	0.84	0.55	0.000	0.85	0.54	-0.002	0.85	0.54	0.020
	10-min	0.81	0.55	0.028	0.82	0.55	0.034	0.72	0.56	0.084
	15-min	0.79	0.54	-0.006	0.68	0.53	0.041	0.71	0.54	0.038
	30-min	0.64	0.55	0.107	0.69	0.54	0.040	0.61	0.55	0.147
	60-min	0.61	0.53	0.051	0.60	0.55	0.115	1.00	0.52	0.083
3-month future	5-min	0.84	0.55	-0.030	0.85	0.54	-0.016	0.84	0.54	-0.080
	10-min	0.81	0.54	0.043	0.81	0.54	0.063	0.72	0.55	0.101
	15-min	0.80	0.54	0.021	0.62	0.54	0.101	0.66	0.55	0.222
	30-min	0.63	0.55	0.168	0.63	0.55	0.177	0.63	0.55	0.295
	60-min	0.61	0.53	0.231	0.60	0.54	0.279	0.62	0.53	0.311
4-month future	5-min	0.84	0.56	-0.109	0.85	0.56	-0.114	0.85	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
	60-min	0.60	0.55	-0.056	0.61	0.53	-0.070	0.61	0.52	0.072
5-month future	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
	60-min	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

Note: 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.

		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
target variable	time-scale	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample
1-month future	5-min	0.54	0.54	0.038	0.54	0.54	0.065	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
	15-min	0.54	0.53	0.019	0.54	0.53	0.040	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
	60-min	0.54	0.56	0.061	0.53	0.56	0.066	0.54	0.55	0.050
2-month future	5-min	0.54	0.54	0.064	0.54	0.55	0.069	0.54	0.55	0.093
	10-min	0.54	0.52	0.012	0.53	0.53	0.017	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
	30-min	0.55	0.55	0.048	0.55	0.54	0.082	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
3-month future	5-min	0.54	0.54	0.081	0.54	0.55	0.092	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
	15-min	0.55	0.54	0.151	0.54	0.54	0.176	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
	60-min	0.54	0.56	0.291	0.54	0.57	0.319	0.55	0.56	0.349
4-month future	5-min	0.54	0.52	0.011	0.54	0.53	0.020	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
	15-min	0.54	0.52	-0.055	0.51	0.51	0.049	0.51	0.50	0.000
	30-min	0.56	0.55	0.045	0.55	0.54	0.151	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
5-month future	5-min	0.53	0.53	-0.070	0.53	0.53	-0.068	0.53	0.53	0.023
	10-min	0.52	0.51	-0.167	0.52	0.52	-0.160	0.53	0.51	0.090
	15-min	0.54	0.53	-0.066	0.54	0.53	-0.072	0.54	0.50	0.145
	30-min	0.55	0.56	0.028	0.56	0.54	0.020	0.56	0.54	-0.119
	60-min	0.51	0.57	-0.011	0.53	0.55	-0.008	0.53	0.53	0.035
mean		0.54***	0.54***	0.04	0.54***	0.54***	0.06	0.54***	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	0.11
max		0.56	0.57	0.29	0.56	0.57	0.32	0.56	0.56	0.35
min		0.51	0.51	-0.17	0.51	0.51	-0.16	0.51	0.50	-0.24

Note: 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.

		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
target variable	time-scale	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample
1-month future	5-min	0.51	0.60	0.015	0.51	0.57	0.024	0.51	0.58	0.045
	10-min	0.50	0.43	0.001	0.50	0.45	-0.007	0.50	0.44	-0.031
	15-min	0.50	0.45	0.008	0.50	0.46	0.005	0.50	0.45	-0.031
	30-min	0.51	0.47	0.020	0.51	0.47	0.022	0.51	0.46	-0.040
	60-min	0.51	0.54	-0.011	0.51	0.52	-0.010	0.52	0.53	0.069
2-month future	5-min	0.50	0.41	0.014	0.50	0.44	0.009	0.50	0.43	-0.006
	10-min	0.50	0.44	0.009	0.50	0.46	0.002	0.50	0.45	-0.038
	15-min	0.50	0.46	0.014	0.50	0.47	0.007	0.50	0.46	-0.042
	30-min	0.51	0.47	0.015	0.51	0.48	0.016	0.51	0.47	-0.050
	60-min	0.51	0.48	0.051	0.51	0.49	0.014	0.50	0.48	-0.081
3-month future	5-min	0.50	0.42	0.022	0.53	0.56	0.070	0.52	0.57	0.005
	10-min	0.50	0.45	0.026	0.50	0.46	0.017	0.50	0.45	-0.044
	15-min	0.51	0.46	0.032	0.50	0.47	0.022	0.50	0.46	-0.041
	30-min	0.51	0.48	0.037	0.51	0.48	0.038	0.51	0.47	-0.042
	60-min	0.52	0.50	0.359	0.52	0.53	0.402	0.51	0.47	0.262
4-month future	5-min	0.50	0.41	0.002	0.50	0.44	0.014	0.50	0.42	-0.005
	10-min	0.50	0.44	-0.001	0.50	0.45	-0.006	0.50	0.44	-0.029
	15-min	0.51	0.46	-0.076	0.51	0.47	-0.095	0.51	0.45	-0.032
	30-min	0.51	0.47	-0.096	0.51	0.47	-0.123	0.51	0.46	-0.027
	60-min	0.53	0.54	0.257	0.53	0.54	0.289	0.51	0.49	-0.187
5-month future	5-min	0.51	0.33	-0.045	0.51	0.36	-0.049	0.50	0.30	-0.057
	10-min	0.51	0.37	-0.042	0.51	0.38	-0.046	0.51	0.31	0.004
	15-min	0.51	0.38	-0.042	0.51	0.39	-0.047	0.51	0.32	-0.032
	30-min	0.52	0.41	-0.058	0.52	0.41	-0.067	0.52	0.34	-0.088
	60-min	0.52	0.43	-0.094	0.52	0.43	-0.116	0.52	0.34	-0.061
mean		0.51***	0.45***	0.02	0.51***	0.47***	0.02	0.51***	0.44***	-0.02
t-stat		5.67	-4.43		5.06	-3.34		5.94	-4.03	
std		0.01	0.06	0.10	0.01	0.05	0.11	0.01	0.07	0.08
max		0.53	0.60	0.36	0.53	0.57	0.40	0.52	0.58	0.26
min		0.50	0.33	-0.10	0.50	0.36	-0.12	0.50	0.30	-0.19

Note: 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.

		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
target variable	time-frequency	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample
1-month futures	5-min	0.98	0.55	-0.002	0.98	0.54	0.004	0.99	0.53	0.030
	10-min	0.99	0.54	0.031	0.99	0.54	0.060	0.99	0.53	0.065
	15-min	0.98	0.54	0.026	0.71	0.55	0.052	0.70	0.54	0.049
	30-min	0.74	0.55	0.053	0.73	0.56	0.066	0.73	0.56	0.062
	60-min	0.84	0.54	0.062	0.80	0.55	0.042	0.79	0.54	0.034
2-month futures	5-min	0.69	0.55	0.091	0.69	0.55	0.069	0.68	0.54	0.087
	10-min	0.98	0.54	0.025	0.99	0.54	0.024	0.98	0.53	0.018
	15-min	0.68	0.53	0.037	0.72	0.53	0.053	0.71	0.53	0.054
	30-min	0.73	0.56	0.089	0.69	0.55	0.078	0.69	0.56	0.109
	60-min	0.82	0.53	0.069	0.80	0.54	0.096	0.79	0.52	0.056
3-month futures	5-min	0.68	0.55	0.141	0.68	0.54	0.165	0.99	0.54	-0.066
	10-min	0.98	0.54	0.120	0.71	0.53	0.215	0.68	0.53	0.208
	15-min	0.72	0.54	0.166	0.70	0.55	0.214	0.69	0.55	0.101
	30-min	0.73	0.55	0.137	0.71	0.56	0.142	0.71	0.55	0.049
	60-min	0.77	0.55	0.291	0.82	0.58	0.299	0.79	0.56	0.362
4-month futures	5-min	0.98	0.55	-0.157	0.98	0.54	-0.138	0.99	0.53	-0.128
	10-min	0.98	0.53	-0.006	0.98	0.53	0.013	1.00	0.53	-0.090
	15-min	1.00	0.54	-0.201	0.73	0.54	-0.126	0.71	0.54	-0.268
	30-min	0.74	0.54	-0.057	0.70	0.55	-0.062	0.69	0.55	0.054
	60-min	0.80	0.56	0.135	0.79	0.56	-0.071	0.78	0.54	0.055
5-month futures	5-min	0.99	0.62	0.042	0.98	0.62	0.061	0.98	0.67	0.100
	10-min	0.99	0.60	0.051	1.00	0.61	0.130	0.99	0.66	0.051
	15-min	0.99	0.60	0.130	0.99	0.61	0.013	0.99	0.67	0.174
	30-min	0.99	0.60	-0.022	1.00	0.61	0.009	1.00	0.66	0.144
	60-min	1.00	0.56	0.017	0.99	0.58	0.017	1.00	0.63	-0.006
mean		0.87***	0.55***	0.05	0.83***	0.56***	0.06	0.84***	0.56***	0.05
t-test		14.38	11.25		12.57	10.42		12.27	6.59	
std		0.13	0.02	0.10	0.13	0.03	0.10	0.14	0.05	0.12
max		1.00	0.62	0.29	1.00	0.62	0.30	1.00	0.67	0.36
min		0.68	0.53	-0.20	0.68	0.53	-0.14	0.68	0.52	-0.27

Note: 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.

		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
target variable	time-scale	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample
1-month future	5-min	0.53	0.59	0.020	0.52	0.57	0.026	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
	15-min	0.55	0.57	0.023	0.54	0.57	0.028	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
	60-min	0.56	0.56	0.021	0.56	0.55	0.004	0.52	0.54	0.122
2-month future	5-min	0.53	0.59	0.043	0.52	0.57	0.026	0.52	0.57	0.052
	10-min	0.53	0.56	0.005	0.54	0.55	0.002	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
	30-min	0.57	0.55	0.064	0.57	0.54	0.059	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
3-month future	5-min	0.53	0.58	0.041	0.54	0.56	0.043	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
	15-min	0.55	0.56	0.095	0.55	0.56	0.126	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
	60-min	0.56	0.57	0.296	0.55	0.58	0.312	0.56	0.55	0.322
4-month future	5-min	0.55	0.59	0.019	0.52	0.57	0.006	0.52	0.58	0.031
	10-min	0.54	0.56	-0.076	0.52	0.55	-0.015	0.52	0.56	0.013
	15-min	0.54	0.56	-0.136	0.54	0.55	-0.069	0.53	0.56	-0.093
	30-min	0.57	0.56	-0.064	0.57	0.54	-0.086	0.57	0.55	0.038
	60-min	0.55	0.56	0.055	0.55	0.56	-0.054	0.54	0.56	-0.325
5-month future	5-min	0.58	0.59	0.094	0.57	0.64	0.072	0.57	0.71	0.122
	10-min	0.58	0.61	-0.092	0.58	0.63	0.015	0.57	0.69	0.148
	15-min	0.57	0.60	-0.084	0.58	0.61	-0.093	0.57	0.62	0.122
	30-min	0.57	0.54	0.098	0.57	0.52	0.061	0.57	0.52	-0.177
	60-min	0.54	0.44	-0.064	0.55	0.45	-0.152	0.55	0.40	0.056
mean		0.55***	0.56***	0.02	0.55***	0.56***	0.02	0.55***	0.56***	0.04
t-test		13.97	9.99		12.73	8.43		11.78	5.91	
std		0.02	0.03	0.09	0.02	0.04	0.09	0.02	0.06	0.13
max		0.58	0.61	0.30	0.58	0.64	0.31	0.57	0.71	0.32
min		0.52	0.44	-0.14	0.52	0.45	-0.15	0.52	0.40	-0.33

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

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.

		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
target variable	time-scale	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample	in-sample	out-sample	out-sample
1-month future	5-min	0.57	0.56	0.049	0.57	0.55	0.052	0.57	0.52	0.029
	10-min	0.58	0.50	-0.011	0.58	0.51	-0.008	0.58	0.49	-0.063
	15-min	0.59	0.53	0.007	0.58	0.54	0.043	0.59	0.54	0.046
	30-min	0.62	0.56	0.063	0.61	0.56	0.094	0.61	0.55	0.055
	60-min	0.64	0.54	0.006	0.64	0.55	0.014	0.63	0.53	0.005
2-month future	5-min	0.58	0.56	0.076	0.58	0.54	0.065	0.58	0.54	0.059
	10-min	0.58	0.52	0.014	0.58	0.51	-0.017	0.57	0.52	-0.010
	15-min	0.59	0.53	0.049	0.59	0.54	0.064	0.58	0.53	0.028
	30-min	0.61	0.55	0.105	0.60	0.55	0.126	0.60	0.55	0.085
	60-min	0.65	0.55	0.087	0.63	0.55	0.092	0.63	0.54	0.065
3-month future	5-min	0.57	0.54	0.150	0.58	0.54	0.167	0.57	0.54	0.083
	10-min	0.58	0.53	0.128	0.59	0.53	0.207	0.58	0.53	0.198
	15-min	0.59	0.54	0.156	0.59	0.56	0.220	0.58	0.55	0.104
	30-min	0.61	0.56	0.170	0.61	0.55	0.166	0.60	0.55	0.053
	60-min	0.66	0.55	0.246	0.64	0.57	0.298	0.65	0.55	0.359
4-month future	5-min	0.58	0.54	-0.026	0.58	0.53	0.029	0.58	0.52	0.115
	10-min	0.59	0.52	-0.216	0.59	0.52	-0.318	0.58	0.51	-0.422
	15-min	0.59	0.53	-0.122	0.59	0.54	-0.139	0.59	0.53	-0.129
	30-min	0.62	0.55	-0.052	0.61	0.55	-0.071	0.60	0.55	0.052
	60-min	0.63	0.55	0.054	0.64	0.54	-0.105	0.63	0.53	-0.124
5-month future	5-min	0.66	0.58	0.087	0.66	0.59	0.113	0.65	0.63	0.067
	10-min	0.67	0.56	0.063	0.66	0.58	-0.034	0.65	0.63	0.137
	15-min	0.67	0.55	0.097	0.66	0.58	0.115	0.65	0.63	0.228
	30-min	0.69	0.57	0.107	0.67	0.59	0.028	0.67	0.64	0.120
	60-min	0.69	0.53	-0.109	0.67	0.55	-0.275	0.67	0.61	-0.004
mean		0.62***	0.54***	0.05	0.61***	0.55***	0.04	0.61***	0.55***	0.05
t-stat		14.75	13.25		16.46	11.63		16.51	6.22	
std		0.04	0.02	0.10	0.03	0.02	0.14	0.03	0.04	0.14
max		0.69	0.58	0.25	0.67	0.59	0.30	0.67	0.64	0.36
min		0.57	0.50	-0.22	0.57	0.51	-0.32	0.57	0.49	-0.42

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

Table 9: ARIMA method: in-sample and out-of-sample accuracy results and ideal profit ratios for different train/test combinations for the static analysis.

		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
3-month future	5-min	0.53	0.53	0.53	0.53	0.53	0.55
	10-min	0.52	0.50	0.52	0.51	0.51	0.51
	15-min	0.52	0.51	0.52	0.51	0.52	0.51
	30-min	0.54	0.53	0.54	0.52	0.53	0.53
	60-min	0.56	0.54	0.55	0.55	0.55	0.55
4-month future	5-min	0.54	0.51	0.54	0.52	0.53	0.51
	10-min	0.53	0.51	0.52	0.51	0.53	0.50
	15-min	0.52	0.52	0.52	0.51	0.52	0.51
	30-min	0.54	0.54	0.54	0.52	0.54	0.53
	60-min	0.55	0.56	0.56	0.53	0.55	0.51
5-month future	5-min	0.50	0.52	0.50	0.54	0.51	0.55
	10-min	0.50	0.53	0.50	0.54	0.51	0.54
	15-min	0.51	0.54	0.51	0.54	0.51	0.51
	30-min	0.51	0.54	0.51	0.52	0.52	0.52
	60-min	0.51	0.55	0.51	0.55	0.52	0.55
mean		0.53***	0.52***	0.53***	0.52***	0.53***	0.52***
t-stat		8.30	6.20	8.28	7.33	9.05	7.30
std		0.02	0.02	0.02	0.02	0.01	0.02
max		0.56	0.56	0.56	0.55	0.56	0.55
min		0.50	0.50	0.50	0.50	0.51	0.50

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