

Article

Enhancing Bitcoin Price Fluctuation Prediction Using Attentive LSTM and Embedding Network

Yang Li ^{1,2} , Zibin Zheng ^{1,2,*}  and Hong-Ning Dai ^{3,*} 

¹ School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China; liyang99@mail2.sysu.edu.cn

² National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou 510275, China

³ Faculty of Information Technology, Macau University of Science and Technology, Macau 999078, China

* Correspondence: zhizibin@mail.sysu.edu.cn (Z.Z.); hndai@ieee.org (H.-N.D.)

Received: 2 June 2020; Accepted: 8 July 2020; Published: 16 July 2020



Abstract: Bitcoin has attracted extensive attention from investors, researchers, regulators, and the media. A well-known and unusual feature is that Bitcoin's price often fluctuates significantly, which has however received less attention. In this paper, we investigate the Bitcoin price fluctuation prediction problem, which can be described as whether Bitcoin price keeps or reversals after a large fluctuation. In this paper, three kinds of features are presented for the price fluctuation prediction, including basic features, traditional technical trading indicators, and features generated by a Denoising autoencoder. We evaluate these features using an Attentive LSTM network and an Embedding Network (ALEN). In particular, an attentive LSTM network can capture the time dependency representation of Bitcoin price and an embedding network can capture the hidden representations from related cryptocurrencies. Experimental results demonstrate that ALEN achieves superior state-of-the-art performance among all baselines. Furthermore, we investigate the impact of parameters on the Bitcoin price fluctuation prediction problem, which can be further used in a real trading environment by investors.

Keywords: Bitcoin; price fluctuation prediction; deep learning; embedding network; long short term memory

1. Introduction

Bitcoin has attracted extensive attention from both investors and researchers since it was first proposed by Nakamoto [1] in 2008. Bitcoin is a decentralized digital currency that uses encryption schemes, decentralized consensus, and other mechanisms to verify transactions and ensure the security. Bitcoin can be directly exchanged between two individuals, using a private key and a public key. Moreover, users can transfer Bitcoin currencies to different countries with minimal processing fees, avoiding huge cost charged by traditional financial institutions [2]. The technology behind Bitcoin is called blockchain [3,4]. New Bitcoins (BTC) were created through a consensus process known as mining. Recently, more than 2200 different cryptocurrencies are created and traded publicly (CoinMarketCap.com).

Bitcoin is not controlled by any central authority, removing central banks from managing the currency supply. An unusual feature of Bitcoin price is the large fluctuation in contrast to traditional financial assets (such as gold, stock indexes and commodities). For example, Bitcoin price often fluctuates by more than 10% (or even more than 30%) at some times. More specifically, as shown in Figure 1, the 30-day historical volatility of Bitcoin is much greater than gold, HS300, and S&P500. These unusual and large price fluctuations have attracted significant research interests from investors and researchers.

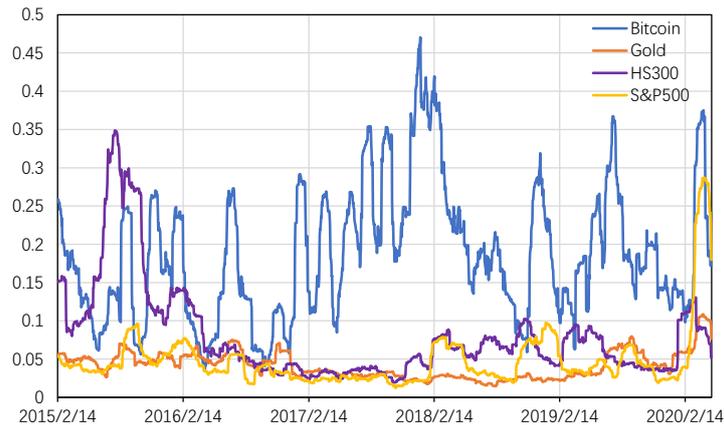


Figure 1. Historical volatility over 30 days.

1.1. Motivation

The large price fluctuation of Bitcoin was caused by many factors, which can be roughly divided into two categories. First, Bitcoin market is a newly developing market. There is no physical representation linked with this kind of virtual asset. Meanwhile, a large number of individual investors can be easily affected by market manipulation [5], consequently making unreasonable decisions. All these issues (fake news, manipulation or other reasons) lead to a large price fluctuation of Bitcoin. Second, Bitcoin market lacks government regulation. Regulators in traditional financial markets are basically missing in the field of cryptocurrencies. For instance, fake news frequently affects the decisions of individual investors. BSV (Bitcoin Satoshi's Vision (BSV) is a variant of Bitcoin since 2018) increased from \$125 to \$251 on 2019.05.29 with the news that BSV would be backed by the Binance currency exchange. However, the price fell back to \$130 when the CEO of Binance clarified that this announcement was a fake news. In addition, Bitcoin is a global product that is affected by regulation around the world. For example, the sharp reduction in Bitcoin price by almost 50% in early 2018 was mostly caused by government regulations in South Korea and China, who forbid the initial coin offerings (ICOs).

We investigate the large fluctuation problem of Bitcoin price in this paper. In particular, we present a Bitcoin price fluctuation prediction problem since many investors care more about whether the sudden rise or fall is worth following. This problem can be simply described as the different behaviors of the Bitcoin price after a certain percentage change (i.e., rise or fall). For instance, if the Bitcoin price reverses after a rapid rising, following the rising price is harmful to investors since investors would make a loss in this case. Otherwise, if the price keeps rising after a rapid rising, it is good for investors since they can make more profits when following the rapid rising.

However, the historical data for this problem is limited since Bitcoin and other related cryptocurrencies have been active in trading since 2017 for most investors. It is difficult to use large interval data (e.g., day-level, and month-level data) because we can only obtain a few training samples. Therefore, we adopt the minute-level data in this paper. It implies that we can ignore the mining information (e.g., miners' revenue, mining difficulties, hash rate, transactions, fees, and so on) of Bitcoin. For instance, we use 1-min level data but the block of Bitcoin (e.g., transactions, fees, miners' revenue) generated approximately every ten minutes. Moreover, the change time of mining difficulties is much longer. Moreover, we do not consider the news information since it is hard to determine the authenticity of a news or predict the occurrence of emergencies. Therefore, we only adopt the minute-level price-based data. Price-based data can also reveal some manipulation behaviors. For instance, they show the same basic characteristics [6]: (1) Manipulators will gradually sell their cryptocurrencies when prices are rising. Otherwise, if they sell them when prices are falling, the prices will fall quickly. Hence, the trading prices will be quite low to them. (2) Manipulators will gradually purchase their cryptocurrencies when the prices are falling. Otherwise, if they purchase them when

prices are rising, it will lead to a high trading cost for them. These basic characteristics are reflected in the changes of Bitcoin price.

1.2. Contributions

To solve the aforementioned problems, we propose an attentive LSTM and embedding network (ALEN) model. More specifically, an attentive LSTM network is used to capture the time dependency representation of Bitcoin price and an embedding network is adopted to incorporate the hidden representations from related cryptocurrencies. We adopt the attentive LSTM because it achieves outstanding performance in time series prediction as shown in [7,8]. We also leverage the embedding network to further capture the relations between Bitcoin and multiple cryptocurrencies since most of the previous studies typically treat Bitcoin as an independent cryptocurrency and ignore the relations with other cryptocurrencies. The rich relations between cryptocurrencies may contain valuable clues for Bitcoin price fluctuation prediction. The work [9] shows the effectiveness in stock prediction by integrating stock relations into prediction. Therefore, we devise an embedding network to incorporate the hidden representations from different related cryptocurrencies into Bitcoin price prediction. Figure 3b shows an illustration of ALEN model. In particular, we first feed the features (see Section 3.2) into an attentive LSTM model. Then, the hidden representations of Bitcoin and other related cryptocurrencies are integrated into the final hidden representation. Finally, a fully connected layer is applied to the final hidden representation to obtain the results.

Three key contributions of the paper are summarized as follows.

- In this paper, we investigate the Bitcoin price fluctuation prediction problem. Meanwhile, we give a clear problem definition and introduce three kinds of features. This work can enrich the research in Bitcoin and provide investors with more tools for investment analyses.
- A novel model namely ALEN is proposed to solve the fluctuation prediction of Bitcoin price. In particular, an attentive LSTM network is used to capture the time-dependency features of Bitcoin price and an embedding network is proposed to capture the hidden representations from related cryptocurrencies.
- We empirically demonstrate the effectiveness of our ALEN on the real-world cryptocurrency market. Moreover, ALEN achieves superior state-of-the-art performance among all baselines.

The remainder of this paper is organized as follows. Section 2 introduces the related work. Section 3 introduces the problem definition and features designation. Section 4 presents our ALEN model, which includes an attentive LSTM network and an embedding network. Section 5 describes experimental results. We conclude the paper in Section 6.

2. Related Work

2.1. Bitcoin

Bitcoin as a decentralized cryptocurrency, is not controlled by any central authority, thereby mitigating the necessity of central banks for managing the currency supply. Therefore, many studies focus on this new emerging technology [10,11]. The work [12] investigates the volatility of Bitcoin, gold, and the dollar. They reveal the similarity among these assets and show that Bitcoin has a larger return than that of the others. It is also shown that a GARCH method can estimate the volatility of Bitcoin. The similar studies focusing on the estimation for Bitcoin volatility include [13–15]. Meanwhile, [16] shows that the Bitcoin mining protocol is not incentive-compatible, indicating the vulnerability of Bitcoin mining. The work [17] shows that the price increment is likely driven by manipulations, as indicated by the evidence from Mt.Gox Bitcoin currency exchange. Moreover, [18] presents an empirical investigation into the fundamental value of Bitcoin, showing speculative bubbles in Bitcoin. Recently, [2] shows a systematic analysis of Bitcoin as a financial asset through reviewing a substantial

amount of academic literature in finance. In this paper, we also investigate the large price fluctuation of Bitcoin.

2.2. Bitcoin Price

In the existing literature, researchers have extensively studied the price prediction of Bitcoin. Many previous studies treat the price prediction of Bitcoin as a time series prediction. The studies [19–21] used traditional machine learning including Random Forest, XGBoost, Support Vector Machine to predict the Bitcoin price. However, traditional machine learning methods cannot capture the time dependency of time series. Recently, deep learning methods [22–24] such as Recurrent Neural Networks (RNN) [25] can handle the issue of the time dependency. However, RNN is struggling to learn long-term dependencies due to the vanishing gradient. LSTM and Gated Recurrent Unit (GRU) that are the most commonly used variants of RNN can solve the vanishing gradient problem. For instance, the study [26] used the LSTM network to predict the price direction of Bitcoin. Wu et al. [27] used two various LSTM models including the conventional LSTM model and LSTM with AR model to predict Bitcoin price. Meanwhile, the work [28] used GRU to predict Bitcoin price, which performs better than RNN and LSTM models. However, all these studies have only simply applied common machine learning methods used in stock price prediction for Bitcoins while failing to capture the unique characteristics of Bitcoin. Moreover, some studies use specific characteristics of Bitcoin that are different from stocks to make a prediction. In particular [29], used Bitcoin transaction graph to predict Bitcoin price. All Bitcoin transactions are available on a public ledger, authors used the transaction ID, sender, recipient, value and a timestamp included in a transaction to construct features to predict Bitcoin price. The work [30] designed a complex method that uses the transaction network's most frequent edges to predict the future price of Bitcoin. The study [31] used various variables including Blockchain data (e.g., transactions per block, median confirmation time, hash rate, difficulty) and macroeconomic variables (e.g., S&P500, gold) to predict the evolution of Bitcoin price.

Our work mainly leverages the large price fluctuation Bitcoin. To the best of our knowledge, the presented problem in this paper is firstly investigated in both Bitcoin and stocks. Our work can also enrich the research in the field of Bitcoin and provide investors with more effective tools for investment analyses.

3. Problem Definition and Features Designation

This section defines the Bitcoin price fluctuation prediction problem, and formally introduces three types of features for modeling Bitcoin price fluctuation prediction problem.

3.1. Problem Definition

We then provide a clear definition of the Bitcoin price fluctuation prediction problem, which is proposed for the first time in this paper, as follows.

First, we introduce some notations and definitions. We divide time length T into N intervals of sub-length τ ($\tau = T/N$), and τ can be n -second, n -minute or other intervals. We also define the maximal intervals N_{\max} ($N \leq N_{\max}$). Second, suppose that Bitcoin price rises over η from T_0 to $T_0 + T$, as shown Figure 2a. Specifically, η is the change of Bitcoin price in time T . Third, the goal is to determine whether the following price will decrease or not in the next max intervals N'_{\max} from $T_0 + T + 1$. The time length is computed as $T' = N' \times \tau$ ($N' \leq N'_{\max}$).

For the following Bitcoin price behaviors after a sudden large fluctuation, we define the labels as follows. If the following Bitcoin price keeps rising from $T_0 + T + 1$ to $T_0 + T + 1 + T'$ compared to the price at $T_0 + T$, it can be formulated as $\frac{1}{T'} \sum_{t=T_0+T+1}^{T_0+T+1+T'} p_t \geq p_{T_0+T}$. In this case, we define the incremental Bitcoin price as a positive sample and label it as 1, as shown in Figure 2b. Otherwise, we label it as -1 . Similarly, if the following Bitcoin price keeps dropping from $T_0 + T + 1$ to $T_0 + T + 1 + T'$ compared to the price at $T_0 + T$, it can be formulated as $\frac{1}{T'} \sum_{t=T_0+T+1}^{T_0+T+1+T'} p_t < p_{T_0+T}$. In this case, we define the cryptocurrency price fluctuation of a cryptocurrency decrease as a positive sample and

label it as 1, as shown in Figure 2b. Otherwise, we label it as -1 . The consideration of the average price used here is to alleviate the stochastic property of Bitcoin price.

For investors, label -1 means that they should not follow the Bitcoin price trend. For instance, they should short Bitcoin if the current Bitcoin price has been in an up trend. Similarly, they should long Bitcoin price if current Bitcoin in a down trend. Label 1 means that they should follow the current Bitcoin price trend to make more profits.

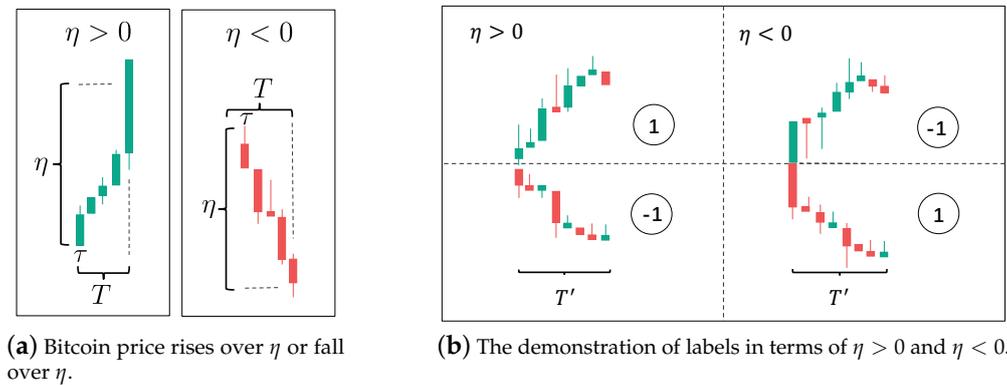


Figure 2. The demonstration of Bitcoin price fluctuation.

3.2. Features for Bitcoin Price Fluctuation Prediction

This section focuses on the features used in Bitcoin price fluctuation prediction with supervised learning algorithms. Features can be roughly categorized as follows: (1) features are manually extracted from a fluctuation period (e.g., the change in price, the speed of change, the change in turnover and the change in volatility); (2) features are formed by the traditional technical trading indicators (e.g., the simple moving average (SMA), the Bollinger Band (BB), moving average convergence divergence (MACD)), which are always used by individual investors; and (3) features are extracted from DAEs, which may obtain some implied representations.

3.2.1. Basic Features

Features extracted from the large price fluctuation period are as basic features. The details are described as follows.

The change in price, f_{1-1} is equal to or greater than η . The larger the value of f_{1-1} is, the stronger the large price fluctuation signal is.

The speed of change, f_{1-2} is an important feature. For instance, some time series data needs four intervals to achieve η , but some time series data only needs one interval to achieve η . The difference shows the degree of the change in price. The smaller the f_{1-2} is, the stronger the large price fluctuation signal is.

The change in turnover, f_{1-3} that was adopted can be explained as follows. On the one hand, if we fix the trading amount, a larger price needs more turnovers to achieve the same price change rate compared with a smaller price. It also indicates that manipulators need more money to achieve this price change rate. On the other hand, if we fix the price change rate, a larger turnover means that the buyers and sellers have some different opinions regarding this price change rate. In addition, the turnover is difficult to compare between a “bull” market (i.e., an increment in the market) and a “bear” market (a decline in the market). Therefore, we adopt the change in turnover as the feature. Formally, it can be calculated as:

$$f_{1-3} = \frac{\sum_{t=1}^T (P_t \times V_t) - \sum_{t=1}^{T_{\text{long}}} (P_t \times V_t)}{\sum_{t=1}^{T_{\text{long}}} (P_t \times V_t)}, \quad (1)$$

where T_{long} is a previous long period, T is a current short period P_t and V_t are the price and volume at time t , respectively.

The change in volatility, f_{1-4} is an important indicator. It is a measure of the uncertainty and is used to reflect the risk level of Bitcoin price. The larger the value of f_{1-4} is, the stronger the signals are. Formally, the change in volatility between current period (cur) and previous period (pre) is computed as:

$$f_{1-4} = \frac{\sigma_{cur} - \sigma_{pre}}{\sigma_{pre}}, \tag{2}$$

where σ is the standard deviation of the rate of return.

3.2.2. Technical Trading Indicators

Technical trading indicators such as SMA, BB, and MACD [32] used in Bitcoin price fluctuation prediction are based on their widespread usage among individual investors. We make a brief introduction as follows.

SMA, f_{2-1} averages the last N intervals (e.g., seconds, minutes, or days) of Bitcoin price. Formally, it is computed as:

$$SMA = \frac{\sum_{i=1}^N P_i}{N}, \tag{3}$$

where P_i represents the price on i , $SMA/P_N - 1$ is regarded as our feature. Meanwhile, we use different intervals N to calculate various features.

BB, f_{2-2} is a technical indicator that creates two bands around a moving average. These bands are computed by the standard deviation of Bitcoin price. The lower and upper BBs can be calculated as:

$$L = M - 2 \times \sigma_m \tag{4}$$

$$U = M + 2 \times \sigma_m, \tag{5}$$

where σ_m stands for the volatility of the moving average M , $U/L - 1$ is regarded as the feature. Moreover, we use different values of m to construct various features.

MACD, f_{2-3} is a technical indicator that subtracts two exponential averages from each other, namely the short and the long interval exponential average. The mathematical formula for the MACD is:

$$D_i = E[\text{short}]_i - E[\text{long}]_i, \tag{6}$$

where E is the exponential moving average, computed as $E_i = \frac{2}{N+1} * (P_i - E_{i-1}) + E_{i-1}$. The value D_i is as the feature. Meanwhile, we use different short and long periods to form various features.

3.2.3. Denoising Autoencoders Features

The features summarized above are limited, and many features are hidden in the time series financial data. To address the problems mentioned above, we adopt denoising autoencoders (DAEs) [33], which have the ability to extract deep features and eliminate the noise from the input time series with a deep neural network architecture. The DAEs as a type of unsupervised learning structures include three components: encoder, decoder, and training module.

The main process of DAEs is described in Algorithm 1. In particular, first, we add low Gaussian noise $q = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(o_t-\mu)^2}{2\sigma^2}}$ to the initial time series o_t and obtain \tilde{o} . Then, the encoder network f_θ maps \tilde{o} to s , and the decoder network $g_{\theta'}$ maps s to z . Both encoder and decoder networks adopt LSTM networks since LSTM can better capture time dependency of the input time series data. The reconstruction error is measured by the loss $L_2(o, z) = \|o - z\|^2$, and the parameters θ and

θ' are updated through minimizing the error $L_2(o, z)$. Finally, the high-level hidden representation s is regarded as the hidden features.

Algorithm 1 DAEs

Require:

- time series o_t .
- encoder parameters θ .
- decoder parameters θ' .

Ensure:

- hidden features s_t .
 - for** each t in $\{1, \dots, n\}$ **do**
 - adds tiny noise to o_t , obtain \tilde{o} ;
 - Encoder:** $s_t \sim f_\theta(\tilde{o})$;
 - Decoder:** $z_t \sim g_{\theta'}(s_t)$;
 - Training:** $\operatorname{argmin}_{\theta, \theta'} \|s_t - z_t\|^2$;
 - end for**
 - return s_t ;
-

4. Our Approach

As shown in Figure 3, the proposed model ALEN consists of two blocks: an attentive LSTM network and an embedding network. The attentive LSTM network is used to capture the sequential dependency representation of Bitcoin price. The embedding network is devised to capture the hidden representations from related cryptocurrencies.

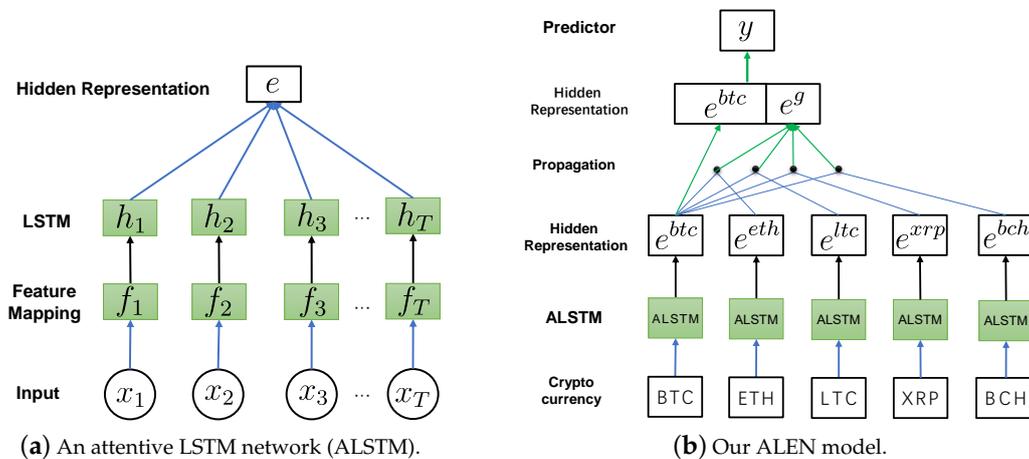


Figure 3. Overview of our ALEN model.

4.1. Attentive LSTM

LSTM [34] is one of the most representative variations of recurrent neural network (RNN) architectures. It was widely used in time series modelling since it can overcome the problem of vanishing gradients and better capture long-term dependencies of time series [7,35]. In particular, it is common used to the prediction of stock price, Bitcoin price [26,27], and so on. As shown in Figure 3a, before applying LSTM, we first leverage a fully connected layer to learn features from the inputs x_t . The learnt features are represented by a latent representation f_t , which is $f_t = \tanh(W_m x_t + b_m)$, where W_m and b_m are parameters to be learnt. After this process, an LSTM layer is applied to map $[f_1, f_2, \dots, f_T]$ to a hidden representation $h_t \in [h_1, h_2, \dots, h_T]$. More specifically, h_t is computed by combining the latent

representation f_t and previous hidden representation h_{t-1} . It can be formulated as $h_t = \text{LSTM}(f_t, h_{t-1})$. Please note that each cell in the LSTM structure can be computed as follows:

$$\begin{bmatrix} \tilde{c}_t \\ o_t \\ i_t \\ \bar{f}_t \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} T_{A,b} \begin{bmatrix} f_t \\ h_{t-1} \end{bmatrix} \tag{7}$$

$$c_t = \tilde{c}_t \odot i_t + c_{t-1} \odot \bar{f}_t \tag{8}$$

$$h_t = o_t \odot \tanh(c_t), \tag{9}$$

where i_t represents an input gate, \bar{f}_t represents a forget gate, o_t represents an output gate, c_t represents a memory cell and h_t is a hidden state. $T_{A,b}$ is an affine transformation, which depends on parameters of the network A and b , σ denotes the logistic sigmoid function and \odot denotes the element-wise multiplication.

Although the LSTM layer can learn long-term dependencies, it cannot detect the important time instance for financial time series prediction. It is important to capture the significance of different time slots for stock movement prediction, as shown in [7,8]. Attention mechanism is much more important, especially in our Bitcoin price fluctuation prediction problem, since our problem mainly focuses on the large fluctuation of Bitcoin price. Formally, the hidden representation h_t learnt from LSTM is transformed as follows:

$$\mathbf{g} = v_g^T \tanh(W_g h_t + b_g), \tag{10}$$

$$\gamma_i = \frac{\exp(g_i)}{\sum_{j=1}^T \exp(g_j)}, \tag{11}$$

$$e^{\text{btc}} = \sum_{i=1}^T \gamma_i h_i, \tag{12}$$

where W_g, b_g, v_g are parameters to be learnt, $\gamma_{1:T}$ are the temporal attention weights, and e^{btc} is the final representation obtained through the attentive LSTM.

4.2. Embedding Network

Figure 4 shows a cumulative return compared to the first trading day. It shows that Bitcoin and Ethereum exhibit quite similar trends in terms of the price change in both daily interval and 15-min interval, revealing the relationship of the Ethereum price and Bitcoin price. For instance, in some cases, if Bitcoin price rises over η , Ethereum price always rises over η at the same time. In addition to the similarity, the relationship between Bitcoin price and Ethereum also shows many other complicated relationships, as depicted as dotted box in Figure 4. The main reason is that the strength of the relation between Bitcoin and other cryptocurrencies is continuously evolving. Therefore, we apply a time-aware embedding network to aggregate the hidden representation of other cryptocurrencies. More specifically, we select four cryptocurrencies (ETH, LTC, XRP, BCH) which fall into the top10 of total cryptocurrencies as the relational cryptocurrencies. Each cryptocurrency is fed to ALSTM to obtain a hidden representation e . All hidden representations are aggregated in our embedding network as follow:

$$e_g = \sum_{j \in \{\text{eth}, \text{lrc}, \text{xrp}, \text{bch}\}} \frac{g(e^{\text{btc}}, e^j)}{4} e^j, \tag{13}$$

where $g(e^{\text{btc}}, e^j)$ represents the weight that contributes to the Bitcoin.

Weight $g(e^{btc}, e^j)$ is computed as follows:

$$g(e^{btc}, e^j) = \alpha \underbrace{e^{btc} e^j}_{\text{similarity}} + (1 - \alpha) \phi(w^T [e^{btc^T}, e^{j^T}]^T + b), \tag{14}$$

where w and b are the parameters in the embedding network to be learned, ϕ is an activation function used to normalize the outputs. Weight g consists of two components and α is the parameter used to control the importance of each component. Specifically, the first component measures the similarity between Bitcoin and other cryptocurrency at the current prediction time steps. Please note that we use inner product here to estimate the similarity. The second component is a nonlinear regression model on the two hidden representations used to learn more complex relations.

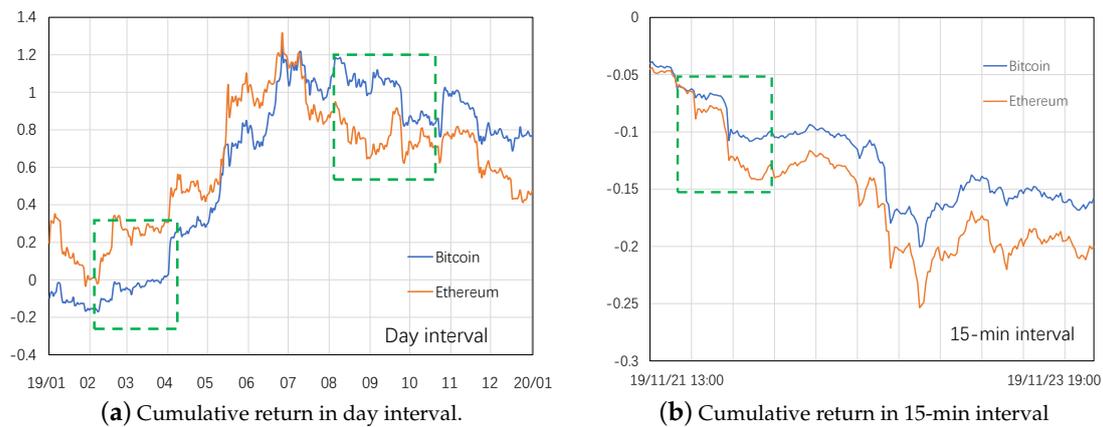


Figure 4. The relationship between Bitcoin and Ethereum.

4.3. Prediction Layer

We first concatenate the hidden representation e^{btc} with e^g into a final latent representation e , which is represented by $e = [e^{btc^T}, e^{g^T}]^T$. Then, we use a fully connected layer as the predictive function to estimate the classification $y = \phi(W_p e + b_p)$, where W_p and b_p are the parameters to be learnt, ϕ is an element-wise nonlinear transformation function.

5. Experiments

We present the experimental results to evaluate performance of the proposed ALLEN model.

5.1. Datasets

Bitcoin is not controlled by any authority, and it is traded on at least 200 cryptocurrency exchanges (e.g., [CoinMarketCap.com](https://www.coinmarketcap.com)). Therefore, the construction of dataset is as follows. The historical trading data of Bitcoin is collected from five bitcoin exchanges (i.e., Huobi, Coinbase, Binance, Bitstamp, Bitfinex) that trade the most bitcoins, and we use the average price of five bitcoin exchanges weighted by the trading volume. The same process is adopted when collecting the other relevant cryptocurrencies including Ethereum (ETH), Bitcoin Cash (BCH), Litecoin (LTC), and XRP. The consideration of the selected cryptocurrencies is that they have a large market capitalisation at their sub-field, and the market capitalisation of all these four related cryptocurrencies falls into the top10 in the cryptocurrency market. Meanwhile, all the datasets are collected from August 2017 to May 2020 with an interval of 1 min (i.e., 1 min). The data of other intervals (e.g., 5 min, 15 min, 1 h, 4 h) are extracted from the data of 1 min. The consideration of starting from 2017 is that Bitcoin and other related cryptocurrencies have been active in trading since 2017 for most investors.

5.2. Experimental Settings

The parameter settings and implementation details are presented as follows. We implement ALEN on TensorFlow2.0 and optimize it by using Adam [36] with an initial learning rate of 0.001. The parameters of Adam are fixed to 0.9 and 0.999. We fix the batch size in the range of [16, 512]. Meanwhile, we find the best η falling into the range of [0, 5%], which is the fluctuation percentage of Bitcoin price. The terms N_{max} and $\max N'_{max}$ are searched from the range of [1, 10]. We let the length of an interval be 1 min, 5 min, 15 min, 1 h, and 4 h. The parameter α is used to balance the importance of the weight, and we let the best value of α falling into the range [0, 1].

5.3. Baselines

First, we evaluate a simple method that always predicts true; this method is called ATrue with a simplest RAND predictor. The RAND predictor makes random guess (up or down) that each direction has an equal probability. Then, we evaluate the hand-crafted (the first two kinds of features) and generated features (the third kind of features) using several classic baselines, Random Forests (RF) [37], Support Vector Machine with RBF kernel (SVM) [38], XGBoost (XGB) [39]. Meanwhile, LSTM and GRU models [28] with recurrent dropout are also applied. All these models are baselines. Moreover, to make an elaborate analysis of all the primary components within our proposed ALEN, we further evaluate the following variations of ALEN, ALUE, and LSTM+Attention (ALSTM) [7]. More specifically, ALEN [2] represents only using technical trading indicators. ALEN [3] represents only using the generated features. ALEN [2, 3] represents using the last two kinds of features. ALUE represents incorporating uniform representation of all related cryptocurrencies rather than aggregated by embedding network. Formally, uniform representation of all related cryptocurrencies is formulated as $e_g = \sum_{j \in \{eth, ltc, xrp, bch\}} \frac{g(e^{btc}, e^j)}{4} e^j$.

5.4. Training Protocol

The training protocol of our model ALEN and baselines are described as follows. First, we construct a 15-min ($\tau = 15$ min) time series for Bitcoin and other related cryptocurrencies. Second, we select the above time series using the parameters $N_{max} = 5$ and $\eta = 2.2\%$, the main process is described in Section 3. The input features of each sample are described in Table 1. More specifically, all features can be fed into traditional machine learning methods (RF, SVM, XGB) and the input dimension is 30 of each sample. However, f_{1-s} cannot be fed into LSTM based methods and GRU since it is the same for each time step. Therefore, the input dimension is 35×26 in each sample, representing 35 time steps fed to LSTM. Third, we label the selected time series using the parameter $N'_{max} = 3$. The main process is also described in Section 3. Moreover, the percentage of positive samples is close to 51.00%. Finally, we split the dataset into training set (the first 60%), validating set (the next 20%), and testing set (the last 20%). The training set is fed into our proposed model ALEN, and we tune parameters through the performance of ALEN model on validating set. For instance, we tune the parameter α in the range of [0, 1], which is used to balance the importance of the weight that contributes to Bitcoin price prediction.

Table 1. Features Description.

Features	Features' Dimension	Details
f_{1-*}	4	$f_{1-1}, f_{1-2}, f_{1-3}, f_{1-4}$
f_{2-1}	8	$N = 2, 3, 5, 7, 10, 15, 20, 25$
f_{2-2}	4	$m = 5, 10, 15, 20$
f_{2-3}	4	short $\in [5, 7, 13, 15]$, long $\in [10, 14, 26, 30]$
hr	10	hidden representation

5.5. Experimental Results

Following the experimental settings and training protocol, we evaluate baselines, our ALEN as well as its variants. The performance comparison is shown in Table 2. We have several observations. Please note that the performance metrics used for evaluation are accuracy, precision, recall, and F1 score, which are standard measurements for classification tasks.

- The accuracy of ATrue predictor is related to the percentage of positive samples. The RAND predictor achieves accuracy around 50/50 percent as we expect. Meanwhile, classic machine learning classifiers perform better than random guess. Meanwhile, LSTM and GRU models [28] perform better than traditional machine learning methods, such as RF and SVM, since they can use the time dependency features well. However, XGB outperforms LSTM and GRU, the main reason may be the sufficient use of our designed features. It is not surprising that ALSTM outperforms LSTM since ALSTM can capture the importance of each timestamps; this observation was shown in [7,8]. Notably, ALEN achieves the best result compared to all baselines with accuracy, precision, recall, and F1 score except ATrue. ALEN outperforms the best baseline ALSTM 6.0% on accuracy and 7.0% on F1 score except ATrue.
- It is clear that the technical trading indicators are better than the features generated by DAEs when comparing the results of ALEN [2] with those of ALEN [3]. The reason may own to the fact that the technical trading indicators can capture the inherent regular of the specific time series. Moreover, all three kinds of features can contribute to the results since ALEN [2,3] achieves the best performance compared to ALEN [2] and ALEN [3].
- It demonstrates that incorporation with other related cryptocurrencies can contribute to the prediction of Bitcoin price fluctuation prediction problem when compared ALUE [2,3] with ALSTM. It also demonstrates the effectiveness of the embedding network when compared ALEN [2,3] with ALUE [2,3]. ALEN uses network embedding method but ALUE uses uniform embedding method. In particular, ALEN [2,3] outperforms ALUE [2,3] 3.3% on accuracy and 3.2% on F1 score.

Table 2. Experimental Results.

Models	Accuracy %	Precision %	Recall %	F1 %
ATrue	51.00	51.00	1.0	67.55
RAND	49.11	50.03	49.48	49.75
RF	54.66	54.48	54.13	54.39
SVM	53.21	53.32	53.01	53.16
XGB	56.65	56.28	57.01	56.64
LSTM	55.19	55.16	55.44	55.30
GRU	55.10	55.06	55.31	55.18
ALSTM	57.58	57.14	57.02	57.07
ALEN [2]	60.68	60.58	60.33	60.45
ALEN [3]	60.46	60.17	60.31	60.23
ALUE [2,3]	59.15	59.16	59.03	59.09
ALEN [2,3]	61.15	61.10	60.93	61.01

5.6. Further Analyses

To the best of our knowledge, our work is the first Bitcoin price fluctuation prediction in the field of cryptocurrency. Therefore, in this section, we conduct experiments to answer the following research questions:

- RQ1: What is the difference between fluctuation prediction problem and traditional price movement prediction problem?
- RQ2: Are the parameters in fluctuation prediction problem sensitive?
- RQ3: How our research could be used by practitioners in the field?

5.6.1. Study of RQ1

Traditional price movement prediction problem can be described as the process of using the past market data or events to predict the price rising or descending at the next time slot. When we set $\eta = 0$, our problem (BFD) degenerates into a traditional price movement prediction problem (SMP). Here, we design an experiment that reveals a difference between our problem and traditional price movement prediction. First, we construct a training set BFD-train and a testing set BFD-test for BFD problem. Second, we use the same parameters as BFD problem by letting η be 0. We construct a new training set SMP-train and testing set SMP-test for the SMP problem. In particular, if we use the BFD-test dataset as testing set, the SMP problem is called SMP1. To provide a fairness comparison, we use the same model ALEN to evaluate the different problems. The result is shown in Table 3.

Table 3. Comparison of different problems.

Problem	Accuracy %	Precision %	Recall %	F1 %
SMP1	52.34	52.16	52.31	52.23
SMP	54.34	54.44	54.22	54.33
BFD	61.15	61.10	60.93	61.01

As observed in Table 3, SMP outperforms SMP1 as expected since the distribution between training set and testing set in SMP1 is different. The BFD outperforms the SMP 12.53% on accuracy and 12.30% on F1 score. Moreover, the performance of SMP with ALEN model even performs worse than BFD with RF model. It indicates the effectiveness of our defined problem compared to traditional price movement prediction. The main difference between BFD and SMP problem is that BFD selects a subset samples from SMP problem. The intuition behind this effect is that when more data is introduced to the model, not only the signals are introduced, but also more noises are added. Therefore, our defined problem provides a new way to enhance the accuracy for price movement prediction through selecting a new dataset to learn.

5.6.2. Study of RQ2

We further discuss the effect of hyper-parameters on our proposed problem. We first investigate the impact of the fluctuation percentage of Bitcoin price η . We vary its value in the range of [1.3%, 5%] while fixing the other parameters. As shown in Figure 5a, the result indicates that too small or too large η cannot achieve the best result. Too small η will make our problem close to traditional price movement prediction problem and cannot achieve a better result. Too large η will make the selected samples too small and also cannot achieve the compelling result, as shown in Figure 5b.

Then, we investigate the impact of α . The parameter α is used to balance the weight of similarity and a nonlinear regression in embedding network. We vary its value within [0, 1] while fixing the other parameters. The result is demonstrated in Figure 6a, indicating that too small or too large α cannot achieve a better result. It also evidences the effectiveness of the combination of similarity and a nonlinear regression that measure the importance of related cryptocurrencies.

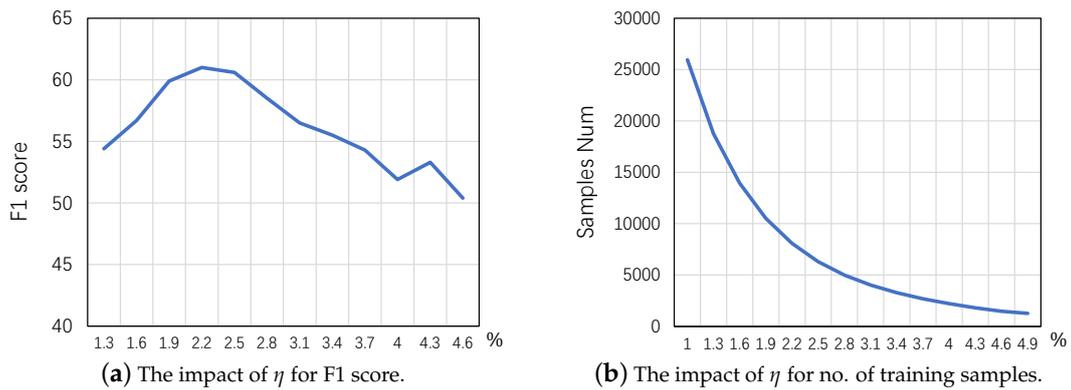


Figure 5. The impact of η .

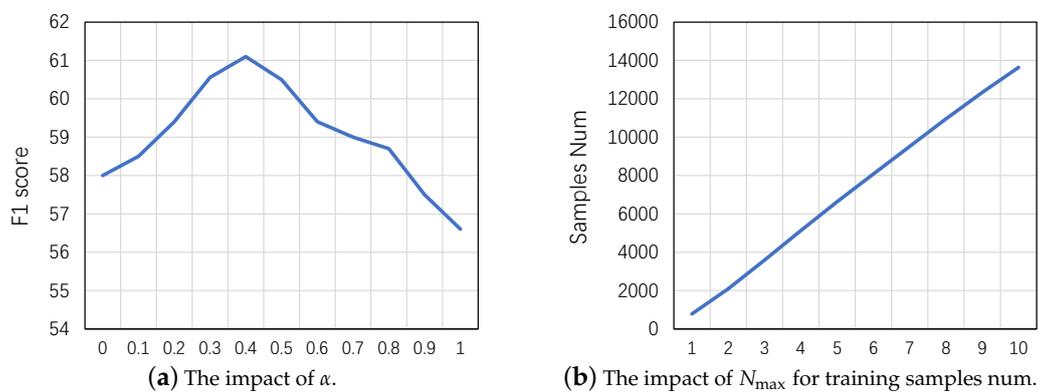


Figure 6. The impact of α , N_{\max} .

We next investigate the impact of N_{\max} and N'_{\max} , where N_{\max} represents our strictness of selecting fluctuant time series. Results are shown in Figure 6b. First, we vary the value of N_{\max} within [1, 10] and fix the other parameters. In particular, we set $N'_{\max} = 3$. Results are shown in Figure 7a. It can be seen that F1 score achieves the best when $N_{\max} = 5$. Too large or too small values of N_{\max} cannot achieve a better performance. Moreover, the change of F1 score is small with the change of N_{\max} . Second, we vary the value of N'_{\max} within [1, 10] and fix the other parameters. In particular, we set $N_{\max} = 5$. Results are shown in Figure 7b. It can be seen that F1 score achieves the best when $N'_{\max} = 3$. Besides, the value of N'_{\max} performs more sensitively when comparing Figure 7a with Figure 7b. The main reason is that the larger value of N'_{\max} means more noise introduced.

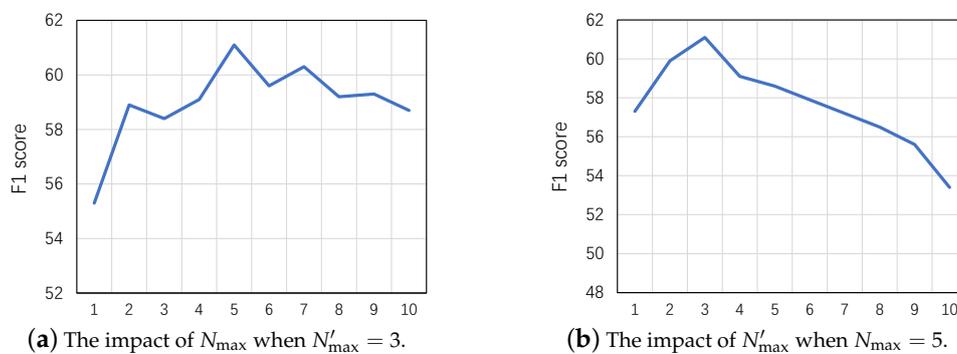


Figure 7. The impact of N_{\max} , N'_{\max} .

Finally, we investigate the parameter τ , which is the length of intervals. Here, we adopt six different intervals (1 min, 5 min, 15 min, 30 min, 1 h, 4 h) to evaluate our model ALEN. All the other

parameters are tuned to the best. The results are shown in Table 4. It demonstrates that the 15 min interval is better than the others and the best η is 2.2%. In particular, the best η at 15 min, 30 min, 1 h and 4 h is not much different.

Table 4. Comparison of Different τ .

Interval	η %	Accuracy %	Precision %	Recall %	F1 %
1 min	0.64	56.03	56.25	56.47	56.36
5 min	1.4	59.15	60.15	58.01	59.06
15 min	2.2	61.15	61.10	60.93	61.01
30 min	2.5	59.86	59.04	59.99	59.51
1 h	2.7	57.86	57.64	57.11	57.37
4 h	2.9	57.75	57.24	58.24	57.73

5.6.3. Study of RQ3

We next provide a detailed analysis of our proposed method for practitioners in the field. In particular, we implement two strategies to evaluate our results: namely the buy-only strategy and the long-short strategy. The consideration of the buy-only strategy is that most Bitcoin exchanges do not allow short the Bitcoins. The details of the buy-only strategy is described as follows. If current Bitcoin price (at time t) increases (or decrease) over η compared to the price at time $t - N$ and we use the model ALEN to predict true (or false) at the same time. We buy n Bitcoins at next open time. Then, we will sell the Bitcoins through next N'_{\max} time steps and we sell $\frac{n}{N'_{\max}}$ Bitcoins every time, which is corresponding to our label definition. The details of the long-short strategy are as follows. First, buying bitcoins is the same as buy-only strategy. In addition, if the price increase (or decrease) over η and we use the model ALEN to predict false (or true) at the same time. We will short n Bitcoins at next open price. Then, we will buy the bitcoins through N'_{\max} time steps. More specifically, we buy $\frac{n}{N'_{\max}}$ Bitcoins at each time step. The simulated results are shown in Figure 8.

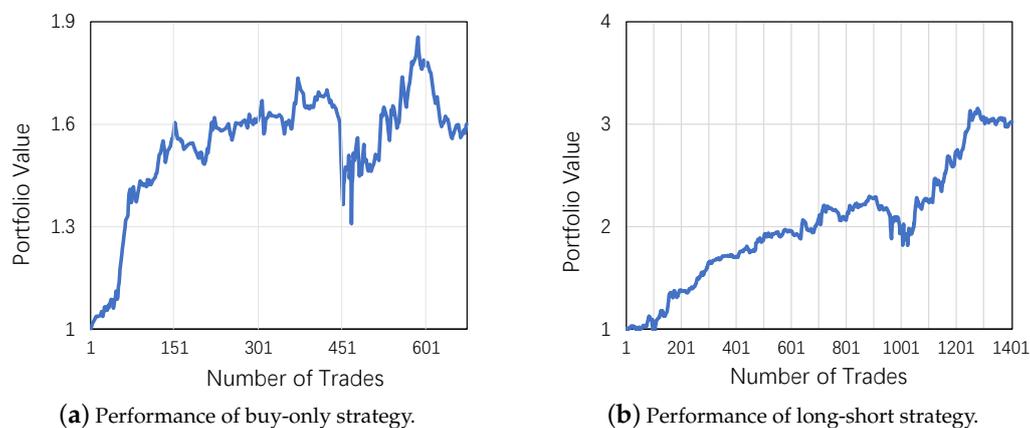


Figure 8. Performance of two strategies.

As shown in Figure 8, horizontal axis represents the number of trades while the vertical axis represents the portfolio value. An initial portfolio value is set to 1. At each trade, portfolio value adds the profits ratio which is calculated by (sell-price - buy-price)/buy-price. The performance of the long-short strategy is better than the buy-only strategy when compared Figure 8b with Figure 8a. The main reason is that the buy-only strategy misses the profit from shorting the Bitcoins. In addition, Figure 8 also shows that the profit of shorting Bitcoins contributes more than the profits of longing bitcoins. Overall, it demonstrates the effectiveness of the proposed model ALEN on real-world cryptocurrency market.

6. Conclusions

In this paper, we investigate the Bitcoin price fluctuation prediction problem in the field of cryptocurrency. The problem can be formally described as follows: will the price keep or reverse if the Bitcoin price anomaly changes (i.e., suddenly rising or falling)? To solve this problem, three kinds of features are carefully designed: basic features, traditional technical trading indicators, and features generated by DAEs. The novel proposed model ALEN cannot only capture the representation of Bitcoin but also aggregate the hidden representation from the related cryptocurrencies. The experimental results show that the designed features are suitable for our presented problem and our proposed ALEN model achieves the state-of-the-art superior performance among all baselines. Moreover, we evaluate the impact of various parameters on our problems. Our study may potentially pave the way toward a real trading environment by investors in the future.

Author Contributions: Conceptualization, Y.L., H.-N.D.; methodology, Y.L.; validation, H.-N.D, Y.L.; formal analysis, Y.L., Z.Z. H.-N.D.; investigation, Y.L.; data curation, Y.L.,Z.Z.; writing—original draft preparation, Y.L., H.-N.D.; writing—review and editing, Y.L., H.-N.D.; supervision, Y.L., H.-N.D., Z.Z.; project administration, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper was supported by the National Key Research and Development Program (2016YFB1000101) , the National Natural Science Foundation of China (U1811461, 61722214) and Macao Science and Technology Development Fund under Macao Funding Scheme for Key R & D Projects (0025/2019/AKP).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakamoto, S. Bitcoin: A Peer-To-Peer Electronic Cash System. Bitcoin. 2008. Available online: <https://git.dhimmel.com/bitcoin-whitepaper/> (accessed on 15 March 2018).
2. Corbet, S.; Lucey, B.; Urquhart, A.; Yarovaya, L. Cryptocurrencies as a financial asset: A systematic analysis. *Int. Rev. Financ. Anal.* **2019**, *62*, 182–199. [[CrossRef](#)]
3. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; pp. 557–564.
4. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [[CrossRef](#)]
5. Chen, W.; Wu, J.; Zheng, Z.; Chen, C.; Zhou, Y. Market Manipulation of Bitcoin: Evidence from Mining the Mt. Gox Transaction Network. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 964–972.
6. Wang, L.X. Dynamical models of stock prices based on technical trading rules part I: The models. *IEEE Trans. Fuzzy Syst.* **2014**, *23*, 787–801. [[CrossRef](#)]
7. Qin, Y.; Song, D.; Cheng, H.; Cheng, W.; Jiang, G.; Cottrell, G.W. A dual-stage attention-based recurrent neural network for time series prediction. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 2627–2633.
8. Feng, F.; Chen, H.; He, X.; Ding, J.; Sun, M.; Chua, T.S. Enhancing stock movement prediction with adversarial training. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; AAAI Press: Menlo Park, CA, USA, 2019; pp. 5843–5849.
9. Feng, F.; He, X.; Wang, X.; Luo, C.; Liu, Y.; Chua, T.S. Temporal relational ranking for stock prediction. *ACM Trans. Inf. Syst. (TOIS)* **2019**, *37*, 1–30. [[CrossRef](#)]
10. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An Overview on Smart Contracts: Challenges, Advances and Platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [[CrossRef](#)]
11. Xie, S.; Zheng, Z.; Chen, W.; Wu, J.; Dai, H.N.; Imran, M. Blockchain for cloud exchange: A survey. *Comput. Electr. Eng.* **2020**, *81*, 106526. doi:10.1016/j.compeleceng.2019.106526. [[CrossRef](#)]
12. Dyhrberg, A.H. Bitcoin, gold and the dollar—A GARCH volatility analysis. *Financ. Res. Lett.* **2016**, *16*, 85–92. [[CrossRef](#)]

13. Katsiampa, P. Volatility estimation for Bitcoin: A comparison of GARCH models. *Econ. Lett.* **2017**, *158*, 3–6. [[CrossRef](#)]
14. Baur, D.G.; Dimpfl, T.; Kuck, K. Bitcoin, gold and the US dollar—A replication and extension. *Financ. Res. Lett.* **2018**, *25*, 103–110. [[CrossRef](#)]
15. Chu, J.; Chan, S.; Nadarajah, S.; Osterrieder, J. GARCH modelling of cryptocurrencies. *J. Risk Financ. Manag.* **2017**, *10*, 17. [[CrossRef](#)]
16. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 3–7 March 2014; pp. 436–454.
17. Gandal, N.; Hamrick, J.; Moore, T.; Oberman, T. Price manipulation in the Bitcoin ecosystem. *J. Monet. Econ.* **2018**, *95*, 86–96. [[CrossRef](#)]
18. Cheah, E.T.; Fry, J. Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. *Econ. Lett.* **2015**, *130*, 32–36. [[CrossRef](#)]
19. Chen, Z.; Li, C.; Sun, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *J. Comput. Appl. Math.* **2020**, *365*, 112395. [[CrossRef](#)]
20. Alessandretti, L.; ElBahrawy, A.; Aiello, L.M.; Baronchelli, A. Anticipating cryptocurrency prices using machine learning. *Complexity* **2018**, *2018*, 8983590. [[CrossRef](#)]
21. Jain, A.; Tripathi, S.; DharDwivedi, H.; Saxena, P. Forecasting Price of Cryptocurrencies Using Tweets Sentiment Analysis. In Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2–4 August 2018; pp. 1–7.
22. Kumar, D.; Rath, S. Predicting the Trends of Price for Ethereum Using Deep Learning Techniques. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 103–114.
23. Livieris, I.E.; Pintelas, E.; Stavroyiannis, S.; Pintelas, P. Ensemble Deep Learning Models for Forecasting Cryptocurrency Time-Series. *Algorithms* **2020**, *13*, 121. [[CrossRef](#)]
24. Ji, S.; Kim, J.; Im, H. A comparative study of Bitcoin price prediction using deep learning. *Mathematics* **2019**, *7*, 898. [[CrossRef](#)]
25. Huisu, J.; Lee, J.; Ko, H.; Lee, W. Predicting bitcoin prices by using rolling window lstm model. In Proceedings of the KDD Data Science in Fintech Workshop, London, UK, 19–23 August 2018.
26. McNally, S.; Roche, J.; Caton, S. Predicting the price of bitcoin using machine learning. In Proceedings of the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Cambridge, UK, 21–23 March 2018; pp. 339–343.
27. Wu, C.H.; Lu, C.C.; Ma, Y.F.; Lu, R.S. A new forecasting framework for bitcoin price with LSTM. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 168–175.
28. Dutta, A.; Kumar, S.; Basu, M. A Gated Recurrent Unit Approach to Bitcoin Price Prediction. *J. Risk Financ. Manag.* **2020**, *13*, 23. [[CrossRef](#)]
29. Greaves, A.; Au, B. *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*; Stanford University: Stanford, CA, USA, 2015.
30. Kurbucz, M.T. Predicting the price of Bitcoin by the most frequent edges of its transaction network. *Econ. Lett.* **2019**, *184*, 108655. [[CrossRef](#)]
31. Jang, H.; Lee, J. An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *IEEE Access* **2017**, *6*, 5427–5437. [[CrossRef](#)]
32. Achelis, S.B. *Technical Analysis from A to Z*; McGraw Hill: New York, NY, USA, 2001.
33. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408.
34. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
35. Li, Y.; Zheng, W.; Zheng, Z. Deep Robust Reinforcement Learning for Practical Algorithmic Trading. *IEEE Access* **2019**, *7*, 108014–108022. [[CrossRef](#)]
36. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

37. Khaidem, L.; Saha, S.; Dey, S.R. Predicting the direction of stock market prices using random forest. *arXiv* **2016**, arXiv:1605.00003.
38. Huang, W.; Nakamori, Y.; Wang, S.Y. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.* **2005**, *32*, 2513–2522. [[CrossRef](#)]
39. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).