Contents lists available at ScienceDirect

Decision Support Systems

Bitcoin price forecasting: A perspective of underlying blockchain transactions

Haizhou Guo [a], Dian Zhang [a,*], Siyuan Liu [b], Lei Wang [c], Ye Ding [d]

[a] *College of Computer Science and Software Engineering, Shenzhen University, 3688 Nanhai Avenue, Nanshan District, Shenzhen, China*
[b] *Department of Supply Chain and Information System, 423 Business Building, University Park, PA 16802, United States*
[c] *Department of Supply Chain and Information System, 426 Business Building, University Park, PA 16802, United States*
[d] *School of Cyberspace Security, Dongguan University of Technology, 1 Daxue Road, Songshan Lake District, Dongguan, Guangdong, China*

ARTICLE INFO

ABSTRACT

Cryptocurrency price forecasting plays an important role in financial markets. Traditional approaches face two challenges: (1) it is difficult to ascertain the influential factors related to price forecasting; and (2) due to the 24/7 trading policy, cryptocurrencies' prices face very large fluctuations, thus weakening the forecasting power of traditional models. To address these issues, we focus on Bitcoin and identify the influential factors related to its price forecasting from the perspective of underlying blockchain transactions. We then propose a price forecasting model WT-CATCN, which leverages Wavelet Transform (WT) and Casual Multi-Head Attention (CA) Temporal Convolutional Network (TCN), to forecast cryptocurrency prices. Our model can capture important positions of input sequences and model the correlations among different data features. Using real-world Bitcoin trading data, we test and compare WT-CATCN with other state-of-the-art price forecasting models. The experiment results show that our model improves the price forecasting performance by 25%.

1. Introduction

Cryptocurrency is a new type of digital asset that uses decentralized networks based on blockchain technology and cryptography to facilitate, secure, and verify transactions. As a representative cryptocurrency, Bitcoin is based on blockchain technology and a payment system [1,2] that is decentralized and based on peer-to-peer transactions [3,4]. It has gained a lot of attention from worldwide businesses, consumers and investors because of the uniqueness of Bitcoin's payment protocol and its growing popularity. In May 2015, the New York Stock Exchange launched a Bitcoin index (NYXBT). In December 2017, the market capitalization of all Bitcoins in the world surged to US $275 billion, with the Bitcoin price peaking at $19,500. Major retailers, such as Microsoft, Overstock, and AT&T, accept Bitcoin as a mode of payment.

After the boom and bust of its price, Bitcoin has been recognized as an investment asset [3]. Due to its highly volatile nature, there is a need for good forecasting on Bitcoin prices to facilitate investment decisions [5,6]. A heated discussion has arisen in response to one question: what determines the monetary value of Bitcoin [4]? Finding the factors that influence Bitcoin's monetary value is very important for both

practitioners and researchers. Identifying proper price predictors helps investors to forecast future price fluctuations and estimate expected return [7]. Retailers and large corporations need to understand the trend of price movement before they decide to adopt Bitcoin as a payment method. Researchers have also shown great interest in Bitcoin and its underlying Blockchain technology. Bitcoin is viewed as a thriving Fin-Tech innovation that disrupts existing payment and monetary systems [7,3]. Thus, to better understand Bitcoin and facilitate its theory development, scholars need to identify the influential factors in the dispersion of new financial technologies, and Bitcoin in particular.

When forecasting the Bitcoin price, it is natural to ask what factors should be taken into account. Though some factors (user comments, blockchain features, order book, etc.) have been leveraged and tested [8–11], previous studies usually depend on the researchers' domain knowledge and did not consider the features from the transactions in the blockchain. Various models have been applied to forecast Bitcoin's price. McNally et al. [12] used Long Short-Term Memory (LSTM) and historical price data to classify price movement patterns. Kristjanpoller et al. [13] tested the ANN-GARCH model for price forecasting, and the ANN part was used to capture the nonlinear effect of price fluctuation.

Guo et al. [10] proposed a probabilistic temporal mixture model to forecast the Bitcoin price. However, due to the high price fluctuation, these models have limited performance in forecasting accuracy and have failed to forecast price movement trends. The importance of forecasting the Bitcoin price has been documented in previous studies. However, research on how to rigorously forecast this price and its movement pattern is still lacking. Traditional approaches face the following challenges: (1) how to identify the potentially important factors that affect Bitcoin price changes; and (2) how to design accurate forecasting models to deal with Bitcoin's high price fluctuation.

Our solution to resolve the first challenge comes from an interesting observation in the cryptocurrency market. Fig. 1[1] demonstrates the trading volumes in different types of exchanges and the correlations between trading volumes and prices. We observe that when the Bitcoin price is falling, the big exchange (e.g., Huobi) has higher trading volumes than the small exchange (e.g., Bleutrade), as shown in column (a). In contrast, when the Bitcoin price is rising, the small exchange (Bleutrade) has higher trading volumes than the big exchange (Huobi), as shown in column (b). Some previous work has shown the effect of exchanges on prices. Giudici et al. [14] found that the big and small exchanges have different partial correlations to the Bitcoin price, and some exchanges are the price setters. Makarov et al. [15] found that Bitcoin price is related to the arbitrage between exchanges. Griffin et al. [16] found that the flow of Tether (i.e., one of the cryptocurrencies) between exchanges is correlated with the return of the Bitcoin price. Thus, the flows between exchanges are more likely to affect the price of cryptocurrencies and should be adopted in the price forecasting analysis.

To enhance the above insight, we denoted the total volumes of exchange when the price rose as *rising volume* and the sum of an exchange's volume when the price fell as *falling volume*. We calculated the ratio of the rising volume to the falling volume for each exchange. The results showed that the average ratio of big exchanges was 1.22, and the average ratio of small exchanges was 1.42. That is, the transaction volumes of small exchanges usually are larger than big exchanges when prices rise. This statistical result validates our insight. Therefore, such volume differences between big and small exchanges may correlate with the Bitcoin price, and we can utilize such correlations to forecast the

Bitcoin price. We then consider the impact of social interest (i.e., by leveraging Google Trends search volume data) as an important measure of investor attention and media hype.

We address the second challenge by proposing a new high-dimensional forecasting model, named WT-CATCN. WT-CATCN is based on the Wavelet Transform (WT) [17] and a well-designed Casual Multi-Head Attention Temporal Convolutional Network (CATCN). CATCN is based on the Causal Multi-Head Attention [18] and Temporal Convolutional Network (TCN) layers [19]. In our model, wavelet transform is utilized to alleviate the non-linear and non-stationary prices. Causal Multi-Head Attention in TCN is leveraged to capture important positions of input data sequences by dynamically calculating the weights according to different data inputs.

To demonstrate the effectiveness of our model, we use Bitcoin as the target of our research and obtain its transaction data from *WalletExplore*,[2] which records the Bitcoin blockchain and merges addresses to different entities. We collect the inter-exchange transaction data from 2016-01-01 to 2018-12-31, including the underlying blockchain transaction data. The experiment results show that the correlations between the volumes of different exchanges indeed contribute to the price forecasting by 19%. Moreover, our WT-CATCN model outperforms the state-of-the-art models by at least 25% in terms of forecasting accuracy at different time scales. The price trends, such as rising and falling, can also be captured by our model.

Within the information systems (IS) discipline, current research on blockchain technologies and cryptocurrencies is still in the nascent stage [20]. Therefore, our work builds upon and extends this area and addresses the call for research that facilitates better price forecasting and modeling for cryptocurrencies in general and Bitcoin in particular. Our paper contributes to the IS and Bitcoin literature in the following ways. First, we analyze the features which may affect the Bitcoin prices from the perspective of transactions in the blockchain, and we find that the volume difference between big and small exchanges can significantly contribute to price forecasting. By demonstrating the predictive power of the volume difference, we help firms develop deeper insights into the specific factors that contributes to cryptocurrency price prediction. To the best of our knowledge, the present paper is one of the pioneer works to utilize such features in cryptocurrency price forecasting. Second, we propose and evaluate a new price forecasting model, WT-CATCN. Combining Wavelet Transform (WT) and Casual Multi-Head Attention with the Temporal Convolutional Network (TCN) helps us to improve the forecasting accuracy for cryptocurrencies with high fluctuation, such as Bitcoin. Third, our WT-CATCN model can clearly detect Bitcoin's price movement (e.g., rising, falling, concave) within 14 days, which is almost impossible for traditional methodologies. Traditional price forecasting methods often adopt an end-to-end approach in which predictions about future prices mainly focus on price itself instead of the time-frequency features of prices. However, due to the high fluctuation, how to capture the trend of future cryptocurrency prices becomes a vital question, and firms can benefit tremendously from accurately detecting such movement trends. The basic intuition of the proposed WT-CATCN method is that we combine both wavelet transform and a well-designed deep learning model to boost the performance of the cryptocurrency price forecasting. The managerial implications of our research can lead to more effective price forecasting, thereby enabling financial firms to improve their ability to handle problems related to the prediction of highly fluctuating prices and achieve a sustainable competitive advantages.

The rest of this paper is organized as follows. Section 2 discusses the related studies on price forecasting in the cryptocurrency market and traditional market. Section 3 provides the preliminaries of the blockchain transaction and problem formulation. Section 4 demonstrates our

Fig. 1. The normalized inter-exchange transaction volumes of Huobi (one of the big exchanges) and Bleutrade (one of the small exchanges). Column (a) shows that the trading volumes of Huobi are larger than those of Bleutrade when the price drops. Column (b) shows that the trading volumes of Bleutrade are larger than Huobi when the price rises. This indicates that the trading volume difference between big and small exchanges could be considered in price forecasting.

---

[1] We use the Min-Max normalization method to scale the volumes into the range [0, 1] to better show the change and trend.

[2] Bitcoin block explorer with address grouping and wallet labeling: https://www.walletexplorer.com/.

system and provides the details of the proposed model. Section 5 describes the evaluation results. Section 6 investigates the scalability of our model. Section 7 concludes this work with a discussion of possible future works.

2. Related work

In this section, we survey the price forecasting not only in the cryptocurrency market but also in the traditional market.

2.1. Price forecasting in the cryptocurrency market

Several methods have been proposed to forecast the price of Bitcoin and other cryptocurrencies. For example, Katsiampa et al. [21] used daily closing prices to evaluate GARCH-type models and explain the volatility of Bitcoin's prices. They found that AR-GARCH is one of the best models based on the goodness-of-fit measure. Mallqui et al. [22] proposed machine learning ensemble algorithms to predict the Bitcoin exchange rate. Combining Artificial Neural Network and AutoRegressive Conditional Heteroskedasticity, an ANN-GARCH framework was chosen from twelve different combinations of models and used to forecast Bitcoin price volatility based on the series of Bitcoin prices [13]. McNally et al. [12] used Bayesian-optimized Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) based on price data to predict three states of price changes (price up, down, and no change) and found that the deep learning models have better performance than the traditional ARIMA model for time-series prediction. Alonso-Monsalve et al. [23] investigated different neural network architectures and concluded that CNN can be used in Bitcoin prediction. The above methods using only past price data cannot achieve high performance in Bitcoin price forecasting.

Recent works have incorporated multiple data sources, such as sentiment-based data sources, to achieve better forecasting performance. Kim et al. [8] analyzed user comments on cryptocurrency online communities for sentiment tagging and used Average One-Dependent Estimator (AODE) to predict price. Stenqvist et al. [24] first classified the sentiment regarding Bitcoin on Twitter into three categories: Positive, Neutral, and Negative. Then they used the three sentiment categories to predict three states of price and showed that the proposed prediction model achieved great prediction accuracy. Similarly, Li et al. [25] used sentiment data on Twitter to forecast Bitcoin price fluctuations. They suggested that using semantics for forecasting results in limited performance improvement because some language schema (such as sarcasm) are difficult to classify. Mohapatra et al. [26] built a cryptocurrency price prediction platform based on Twitter sentiments, which support real-time prediction. Opinion information from Twitter is explored to predict Bitcoin prices [27]. Jain et al. [28] suggested that other features, such as mining cost and economic factors, may also affect Bitcoin price fluctuations.

In addition to the sentiment data, other data sources have also been introduced. Rebane et al. [29] showed that Seq2Seq has better forecasting performance than ARIMA using additional input sources such as Google Trends, Altcoin data, and Bitcoin prices. However, if the Bitcoin price faces falls sharply, the performance of the Seq2Seq model drops significantly. Jang et al. [9] used Bayesian Neural Networks (BNN) to forecast the Bitcoin price based on features extracted from the Blockchain, and they identified a collinearity problem between blockchain features and macroeconomic variables. Their experiments showed that BNN has better predictive performance than the Linear Regression (LR) and Support Vector Regression (SVR) models in both fluctuation and volatility. After extracting the generic properties, such as bid and ask prices of Bitcoin, Amjad et al. [30] used simple, real-time machine learning models to predict three states of price and simulate the trading strategy, which outperformed both EC and ARIMA models. Guo et al. [10] proposed probabilistic temporal mixture models to forecast short-term volatility based on the Bitcoin prices and order book data.

The order book data consists of bid orders (the maximum price to buy) and ask orders (the minimum price to sell), which are stored in the exchange database. Their models showed fewer errors than the statistic models and machine learning baselines.

In this paper, we investigate a new set of features generated from the transactions between exchanges across the underlying blockchain, and we examine their roles in Bitcoin price forecasting. To the best of our knowledge, this paper is one of the pioneer works to use such information to forecast Bitcoin prices.

2.2. Price forecasting in traditional market

The cryptocurrency price forecasting problem is similar to the ones in traditional financial markets, including foreign exchange, stocks, commodities, and so on. We have investigated some methods that might be used to forecast cryptocurrency price changes. For instance, autoregression models have been adopted to study predictability and stock returns [31]. Due to the non-linear and non-stationary features of stock prices, the traditional autoregression models have limited power to make an accurate forecast. To solve the non-linear problems in stock prices, many machine learning techniques [32–35] have been applied. Hassan et al. [34] presented a Hidden Markov Model (HMM)-based adaptive fuzzy inference system to analyze price trends. Nayak et al. [35] proposed a hybrid framework which combined the Support Vector Machine (SVM) and K-Nearest Neighbor approach to balance each model's complexity and errors, achieving well forecasting performance for the stock market indices.

Deep learning is a powerful method, that has revolutionized various industries, including the financial sectors [36]. Multiple studies have demonstrated that Artificial Neural Networks (ANN) models achieve better performance than the ARIMA model. Jiao et al. [37] used the standard machine learning models, including ANN models, to forecast the stock price changes based on 463 stocks in the S&P 500. Faccini et al. [38] proposed as a new predictor the representative investor's implied relative risk aversion. Neely et al. [39] showed that technical indicators (relying on past price and volume patterns) and macroeconomic variables (such as dividend-price ratio) can boost forecasting performance. To extract multi-frequency trading patterns, State Frequency Memory (SFM) [40], inspired by the Discrete Fourier Transform, has been developed. Sul et al. [41] utilized social sentiments and news to forecast stock prices. However, they only concatenated vectors into one super feature vector; thus such methods will lose the intrinsic links among features. Hence, Li et al. [42] used stock data to compare the tensor-based approach with Support Vector Regression (SVR), PCA+SCR, and ISOMAP+SVR, and they showed that the tensor-based approach outperformed other models.

Unlike traditional financial markets, Bitcoin, as a decentralized digital currency, can be traded 24/7 without closing. Moreover, any offline information and events could influence the price of cryptocurrency immediately rather than when the market (e.g., the stock market) opens. We analyzed the stocks dataset from the work [40] and found that the average value of the standard deviation of the 50 stock prices is 23.97. However, the Bitcoin prices' standard deviation is 3923.15. Hence, Bitcoin face greater fluctuations than most of the traditional financial assets, indicating an urgent need for a new forecasting model to address such high price fluctuation.

3. Preliminaries

In this section, we provide preliminary information, including data collection and processing, and we then define the problem related to cryptocurrency price forecasting.

3.1. Cryptocurrency transaction

A blockchain-based cryptocurrency is issued by sending transactions

from senders to receivers. The sender or receiver represents a unique and anonymous address on the blockchain. The sender and the receiver of a transaction could each be a regular account, an institution (bank, exchange, managed fund, etc.), or the blockchain itself (in this case, the amount of the transaction is considered a bonus to the miners). When a transaction is verified by the blockchain, Bitcoin is transferred from the sender's address to the recipient's address. For privacy, the addresses of a transaction are anonymous, while other information such as transaction ID, amount, and fee are public. Furthermore, a single transaction often consists of multiple senders and receivers; thus, it is not easy to identify the particular sender and receiver in a specific cryptocurrency transaction. A cryptocurrency transaction is defined as follows:

Definition 1. (*A Transaction*) A cryptocurrency transaction is a tuple $\xi = \langle h, \delta, P, R, S, V, e \rangle$, where h is the hash ID of the transaction, δ is the timestamp, $P = \{p_1, p_2, ..., p_m\}$ is the list of senders, $R = \{r_1, r_2, ..., r_n\}$ is the list of receivers, $S = \{s_1, s_2, ..., s_m\}$ is the corresponding amounts sent by the senders, $V = \{v_1, v_2, ..., v_n\}$ is the corresponding amounts received by the receivers, and e is the fee of the transaction, such that $|P| = |S|$, $|R| = |V|$, and $\sum(S) = \sum(V) + f$.

Hence, an *inter-exchange transaction* that is, the transaction between two cryptocurrency exchanges, is a subset of cryptocurrency transactions where $|P| = |S| = 1$ and $|R| = |V| = 1$. This is formally defined as:

Definition 2. (*Inter-exchange Transaction*) An inter-exchange transaction is defined as a tuple $\tau = \langle h, \delta, p, r, s, v, e \rangle$, where h is the hash ID of the transaction, δ is the timestamp, p is the sender, r is the receiver, s is the amount sent by the sender, v is the amount received by the receiver, and e is the fee of the transaction. Thus, $s = v + e$.

In the cryptocurrency market, inter-exchange transactions illustrate the flow of cryptocurrency between exchanges. Inside a cryptocurrency exchange, there are also inner-exchange transactions, which illustrate the flow of cryptocurrency between trading accounts inside the exchange. Differing from inter-exchange transactions, inner-exchange transactions are not recorded in the blockchain and are hard to access. Thus, inter-exchange transactions represent trustworthy data which may help in forecasting prices. It is worth noting that, inter-exchange transactions are one of the unique features of cryptocurrency. Such transactions are hard to obtain from standard currencies.

Similar to traditional financial markets, based on the inner-exchange transactions, an exchange provides the market price in many forms. In this paper, based on the suggestions from domain experts, we use Open-High-Low-Close (OHLC)[3] as the indicator of the market price. OHLC illustrates the price movement of a financial instrument over time by summarizing the opening price, highest price, lowest price, and closing price in a given time period. The time period (i.e., *time frame*) is often available by intervals of year, month, week, day, 60 min, 30 min, 15 min, 5 min, 1 min, and so on. In this paper, we use day as the time frame for OHLC. Formally, an inner-exchange market price is defined as:

Definition 3. (*Inner-exchange Market Price*) An inner-exchange market price is defined as a tuple $m_t = \langle o_t, h_t, l_t, p_t, v_t \rangle$, where o_t, h_t, l_t, and p_t are the opening price, highest price, lowest price, and closing price of the date time t, respectively; v_t is the cumulative trading volume in the current time frame.

3.2. Social interest

In this paper, social interest refers to people's interest in cryptocurrencies. Previous literature has shown that social interest affects Bitcoin prices [7]. Representing the main way for people to understand cryptocurrency, social behavior data play an important role in price

forecasting. We select *Google Trends*[4] as the most representative data source for our social interest data [43]. Formally, we denote g_t^κ as the Google Trends result for keyword κ on date t, indicating what people are searching around the world. Calculated by Google, g_t^κ is a continuous variable from 0 to 100 that captures the trending searches over a period of time. The closer g_t^κ is to 100, the more people are interested in keyword κ on date t.

3.3. Data integration

Since the above inter-exchange transactions, inner-exchange market prices, and social interest data often vary over time, we employ the dataset with length T before a specific timestamp d as our input data.

- $X_d^T = \{\tau_i | d - T + 1 < \tau_i(\delta) <= d\}$ is the inter-exchange transactions set consisting of the transactions received from $d - T + 1$ to d;
- $M_d^T = \{m_t | t = \{d - T + 1, ..., d\}\}$ is the past daily market prices before $d + 1$;
- $B_d^T = \{g_t^\kappa | t = \{d - T + 1, ..., d\}\}$ is the past social interest data before $d + 1$.

3.4. Problem definition

In this study, our goal is to forecast the Bitcoin prices within a given time period. Hence, we use the input *T-point* temporal data X_d^T, M_d^T and B_d^T to forecast the future *n-step* prices $\widehat{P_d^n} = \{\widehat{p}_{d+i} | i = 1, 2, ..., n\}$ after timestamp d. The *step* indicates the time granularity (can be set to day, hour, etc.), and n indicates the length of the forecasting, which is from timestamp $d + 1$ to $d + n$. For example, we can forecast the daily prices within the next 7 days (here n is 7, and a step is set as per-day) using the data from the past 60 days (here T is 60, and each point represents one day). The Cryptocurrency Price Forecasting Problem (CPFP) can be formally defined as:

Definition 4. (*Cryptocurrency Price Forecasting Problem*) Minimize the error $\mathcal{O}$ between $\widehat{P_d^n}$ and true prices:

$$\mathcal{O} = \sqrt{\frac{\sum_{i=1}^{n} (\widehat{p}_{d+i} - p_{d+i})^2}{n}} \tag{1}$$

Such that:

$$\widehat{P_d^n} = f(X_d^T; M_d^T; B_d^T) \tag{2}$$

where $\widehat{P_d^n} = \{\widehat{p}_{d+i} | i = 1, 2, ..., n\}$, $\{p_{d+i} | i = 1, 2, ..., n\}$ are the real prices, and f is a non-linear function represented by our model that we aim to learn.

4. System design

In this section, we introduce our forecasting system, which extracts features from the inter-exchange transactions, market prices, and social interest data to forecast prices.

4.1. System overview

To solve the cryptocurrency price forecasting problem, we propose a cryptocurrency price forecasting system as shown in Fig. 2. The system contains two major parts: the feature extraction module and the forecasting model. The first part of our system involves extracting representative features from three categories of datasets (X_d^T, M_d^T and B_d^T). Details are discussed in Section 4.2. In the second part, we propose a
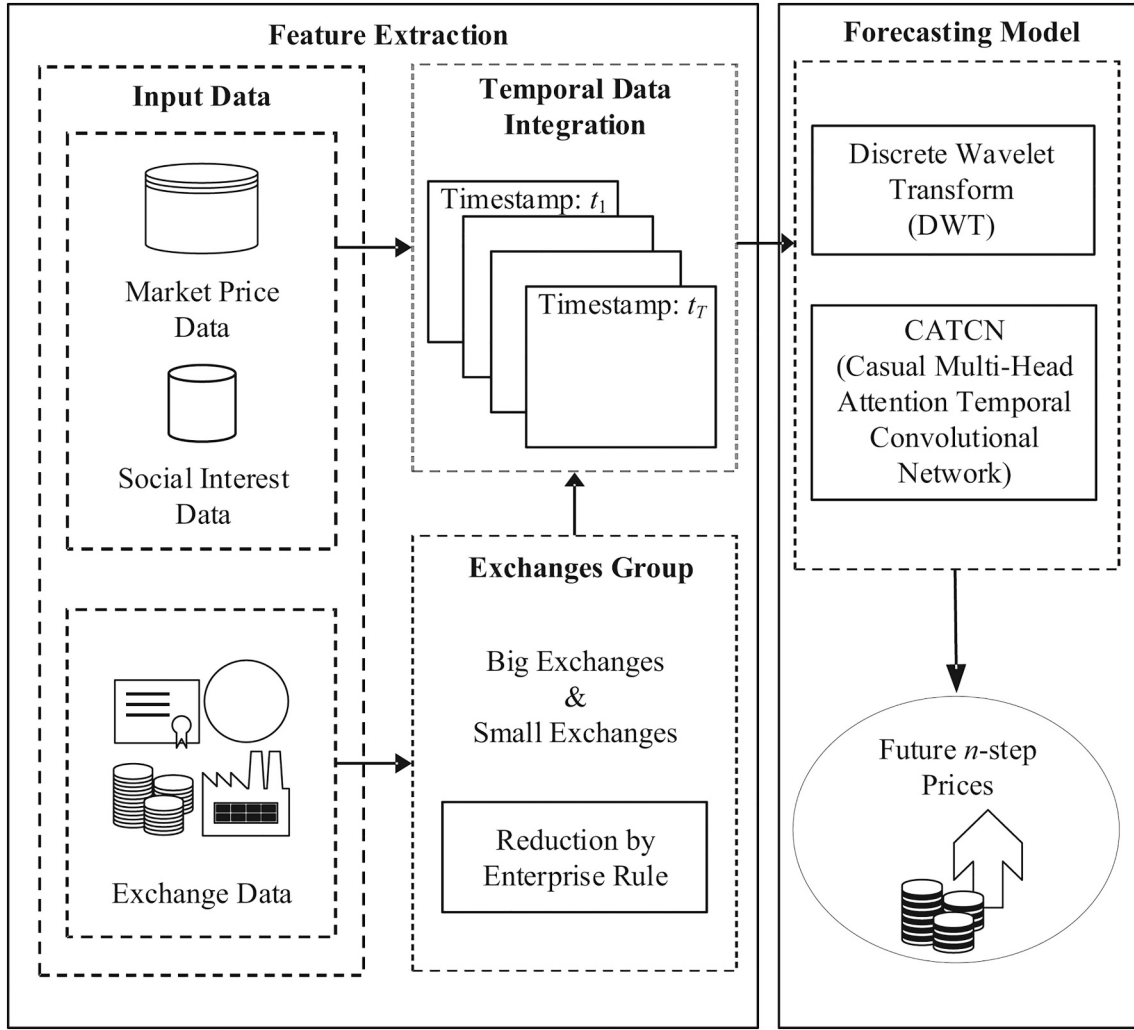

Fig. 2. A new cryptocurrency price forecasting system.

forecasting model called WT-CATCN, which is based on Wavelet Transform (WT) and Casual Multi-Head Attention Temporal Convolutional Network (CATCN). Wavelet transform is employed to alleviate the impact of the non-linear and non-stationary input data, and CATCN is used to address the relationships among input data by adding casual multi-head attention in traditional TCN. The details are provided in Section 4.3.

4.2. Feature extraction

In this section, we present the method used to extract features from the input data. According to the definition in the last section, our input data consist of inter-exchange transactions data X_d^T, inner-exchange market prices data M_d^T and social interest data B_d^T.

As shown in Fig. 1, the volume differences between big and small exchanges may have a potential impact on the Bitcoin price. Therefore, we start by cleansing the data and producing inter-exchange transaction τ_i. Then we divide τ_i into two transaction sets: big exchanges and small exchanges. Here, we adopt the industry rule which utilizes the top-k exchanges in terms of volume as the threshold to differentiate big and small exchanges. The details are as follows.

We collect and preprocess the cryptocurrency transactions ξ from WalletExplorer. Since the collected data come from the underlying blockchain, these underlying transaction data are reliable. As the transactions have multiple senders and receivers, we construct a

cartesian transaction set of blockchain transactions according to the senders and receivers. Then the transaction fee is also recorded for all transactions. The last step is to select inter-exchange transactions τ_i from the cartesian transaction set. WalletExplorer has grouped and labeled the addresses; thus, we can exclude the transactions for which the sender and receiver do not have labels.

The next step is to divide τ_i; we eliminate the redundancy of the $\{\tau_i(p), \tau_i(r)|\tau_i \in X_d^T\}$, then form a unique element set $\mathcal{U}$, which consists of the receiver or sender of τ_i.

Next, we denote the transaction set of exchange u_j generated during the period from time $t-1$ to t, as $\mathcal{H}_t^{u_j} = \{\tau_i|\tau_i(p) = u_j \text{ and } t-1 < \tau_i(\delta) \leqslant t\}$, where $t = \{d-T+1, ..., d\}$.

In order to choose the top-k exchanges, we calculate the volume of each exchange and rank exchanges accordingly as suggested by the widely accepted website CoinMarketCap.[5] For exchange u_j, we define Vol_{u_j} of the exchange as:

$$\text{Vol}_{u_j} = \sum_{t=d-T+1}^{d} \sum_{\tau_i \subseteq \mathcal{H}_t^{u_j}} (\tau_i(v) + \tau_i(s))v_t \tag{3}$$

We define the top-k exchanges as big exchanges, whose transactions

---

[5] CoinMarketCap's service provides information about all digital currencies that are traded in at least one public exchange and have a non-zero trading volume. https://coinmarketcap.com/rankings/exchanges/.

account for nearly half of the total transactions, and the trading volume of these exchanges is always among the top-k. Then, we define the rest as small exchanges. As Fig. 3 shows, the top-4 exchanges dominate around 50% of the transactions in the market. According to the industry rule, we set k as 4.

To explore the trading behavior between big and small exchanges, we record all the transactions between them. We ignore the transactions between big exchanges, since they are considered as internal transactions. Similarly, the trading between small exchanges is likewise not considered.

The inflow of small exchanges is the outflow of big exchanges, and vice versa. For these symmetric links, we only need to calculate once for the transactions' outflow $\tau_i(s)$ and the corresponding transaction fees. We define the extracted features as $\mathcal{X}_t^z$, where $t = \{d - T + 1, ..., d\}$, and z is the vector consisting of the features shown in Table 1. The last four features are the Bitcoin prices transferred into USD in the real world.

4.3. Forecasting model

In this section, we aim to forecast n-step prices $\widehat{P_d^n}$ of cryptocurrency using the time series features $\mathcal{X}_t^z$. Our forecasting model WT-CATCN is based on Wavelet Transform (WT) and Casual Multi-Head Attention Temporal Convolutional Network (CATCN). In order to solve the non-linear and non-stationary problem of the input data, we utilize Wavelet Decomposition of Discrete Wavelet Transform (DWT) to decompose $\mathcal{X}_t^z$ into time-frequency features (as detailed in Section 4.3.1).

Next, we utilize CATCN, which stacks multiple CATCN blocks to forecast the time-frequency features of $\widehat{P_d^n}$. Each CATCN block contains casual multi-head attention and the TCN layer.

Casual multi-head attention (detailed in Section 4.3.2) can make the block focus on the important positions of the data while ensuring the causality (the way of calculation, is defined in [19]). As for the TCN layer (detailed in Section 4.3.3), its convolutions are causal and have long-term memory similar to LSTM. Each CATCN block (detailed in Section 4.3.4) is defined by stacking the causal self-attention layer and TCN layer naturally. Finally, we reconstruct the time-frequency prices to the final prices $\widehat{P_d^n}$ by Wavelet Reconstruction (detailed in Section 4.3.5).

4.3.1. Discrete wavelet decomposition

Wavelet transform is adopted to alleviate the impact of the non-linear and non-stationary input data. Its main property is variable time-frequency resolution, which allows wavelet transform to have a higher-frequency solution (local behavior) with a lower time solution, or a lower-frequency solution (general behavior) with a higher time solution. The use of WT also makes it convenient to train neural networks [44]. Thus, we decompose $\mathcal{X}_t^z$ into multiple time-frequency features of z to explore different hidden patterns in our input data. The details are shown below.

Wavelet is used to decompose input sequences into different scale components. The discrete wavelets [45] ψ are as follows:

$$\psi_{n_1, n_2}(t) = a^{-\frac{n_1}{2}} \psi(a^{-n_1} t - bn_2), \qquad n_1, n_2 \in \mathbb{Z} \tag{4}$$

where a is the scale factor and b is the translation factor. Scale factor a controls the stretch or shrink of the wavelet. Translation factor b can shift the wavelet to move on input sequence $\mathcal{X}_t^z$. $\mathbb{Z}$ represents the integer set.

Discrete wavelet transform is defined as:

$$\mathrm{DWT}_{n_1, n_2} = a^{-\frac{n_1}{2}} \sum_{t=d-T+1}^{d} \mathcal{X}_t^z \psi^*(a^{-n_1} t - bn_2) \tag{5}$$

where * denotes the complex conjugate. The reconstruction of $\mathcal{X}_t^z$ can only be performed when the discrete wavelet satisfies certain conditions.

The common way for the reconstruction is to build a discrete dyadic wavelet transform as follows:

$$\mathrm{DWT}_{i,j} = 2^{-\frac{i}{2}} \sum_{t=d-T+1}^{d} \mathcal{X}_t^z \psi^*(2^{-m} t - j) \qquad n_1, n_2 \in \mathbb{Z} \tag{6}$$

where n_1 and n_2 are set to 1, $a = 2^i$, $b = j2^i$.

Then, $\mathcal{X}_t^z$ can be decomposed and reconstructed by employing Stephane Mallat's Multiresolution Analysis (MRS) [46]. After the decomposition of MRS, the function of discrete wavelet decomposition can be defined as:

$$\Gamma(\mathcal{X}_t^c) = \{A_J^z, D_1^z, D_2^z, ..., D_J^z\} \tag{7}$$

where J is the decomposition level, A_J^z is the approximation components (i.e., low-frequency components used to capture the general behavior) for each feature of z, and $\{D_j^z, |j = 1, 2, ..., J\}$ are detail components (i.e., high-frequency components used to capture the local behavior) for each feature of z. Then, time-series features $\mathcal{X}_t^z$ can be decomposed as $A_J^z, D_1^z, D_2^z, ..., D_J^z$, as Fig. 4 shows.

4.3.2. Casual multi-head attention

Attention mechanism [18] can selectively screen out important information from a large amount of input sequence. Multi-head attention [18] is used to dynamically calculate the attention value for each position in the input sequence, such that different input positions have different importance levels. Since our input time series data are correlated with each other at different timestamps, we use casual multi-head attention to examine the attention values for the input sequence and keep causality within the data.

Casual multi-head attention, by our design, is the first layer of each CATCN block, and there are multiple CATCN blocks in our system. Thus, there are two types of input of casual multi-head attention:

(1) If the layer is in the first CATCN block, the input comprises the corresponding time-frequency features $\{A_J^z, D_1^z, D_2^z, ..., D_J^z\}$. Without loss of generality, we define x as either A_J^z or D_i^z, $i \in [1, J]$.

(2) If the layer is in the rest of the CATCN blocks, the input is the output of its previous CATCN block.

The casual multi-head attention layer combines multiple causal self-attention layers similar to [18], as shown in Fig. 5. Each causal self-attention layer can calculate attention values for x. Multiple causal self-attention layers can calculate multiple attention values and extract more information from input data. We denote the number of causal self-attention layers as N_m.

The causal self-attention layer is calculated by three matrices, Q, K, and V, that represent the *query*, *key*, and *value* respectively [18]. Q, K and V are calculated by multiplying the input x by three parameter matrices having the same size. Their values are updated after the matrices are trained. Attention values are calculated by taking the dot product of the Q and K.

Since the future price data are not available in advance, we need to design a mask[6] to replace the corresponding attention values to negative infinity and to cause the attention values related to future data to be zero after the softmax function. Thus, the self-attention becomes causal by adding the mask. The specific calculation process of the casual self-attention layer is defined as follows.

$$\mathrm{Mask}(X) = \begin{cases} X_{(i,j)} = X_{(i,j)}, & \text{if } i \le j \\ X_{(i,j)} = -\inf, & \text{otherwise.} \end{cases} \tag{8}$$

---

[6] A mask is a matrix to select the places of interest. In our work, it represents the matrix in Eq. (8).
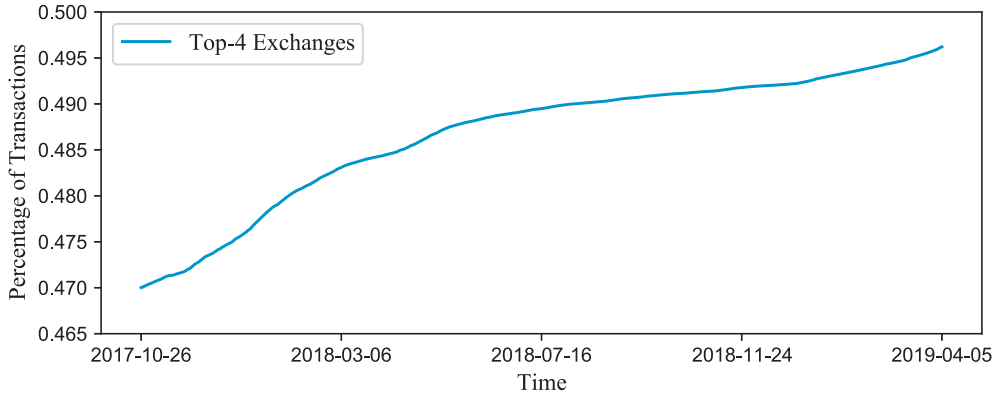
Fig. 3. The top-4 exchanges dominate the whole Bitcoin transactions.

Table 1
The features of $\mathcal{X}_t^z$.

Data source		Features
Market prices data	z_m	The market price m_t as defined in Definition 3.
Social interest data	z_g	The social behavior data g_t^c as defined in Section 3.2.
	z_1	The average number of Bitcoin a_s flowing from big exchanges to small exchanges.
	z_2	The average transaction fee e_s flowing from big exchanges to small exchanges.
	z_3	The average number of Bitcoin a_l flowing from small exchanges to big exchanges.
	z_4	The average transaction fee e_b flowing from small exchanges to big exchanges.
Inter-exchange transaction data	z_5	The average number of USD $a_s^p = a_s p_t$ flowing from big exchanges to small exchanges.
	z_6	The average transaction fee (USD) $e_s^p = e_s p_t$ flowing from big exchanges to small exchanges.
	z_7	The average number of USD $a_b^p = a_b p_t$ flowing to from small exchanges to big exchanges.
	z_8	The average transaction fee (USD) $e_b^p = e_b p_t$ flowing from small exchanges to big exchanges.

can be defined as:

$$\texttt{MultiHead}(x) = \texttt{Concat}(\{head_i\})W_m \qquad (10)$$

where $head_i = \texttt{Attention}(Q_i, K_i, V_i)$, $i = \{1, 2, ..., N_m\}$. Q_i, K_i and V_i are generated by multiplying the input x by different parameter matrices. W_m is a parameter matrix to project different attention values, and the final attention values have the same dimension of x.

Next, we calculate the multi-head attention values $\texttt{MultiHead}(x)$ for x, and the input of the TCN layer (introduced in the next subsection) is calculated as $x' = \texttt{MultiHead}(x) = \{x_i'|i = 1, 2, ..., l_b\}$, where l_b is the length of input x.

4.3.3. Temporal convolutional network layer

Temporal Convolutional Network (TCN) can be used to achieve long memory[7] and simplify the neural networks [19]. It not only can alleviate the problems of vanishing gradient and exploding gradient[8] but also can be computed in a parallel approach.[9]

TCN Layer. The TCN layer is used to build the CATCN block in Section 4.3.4. The TCN layer is composed of a stack of two identical sub-layers. To achieve long memory, each TCN layer has twice the dilation
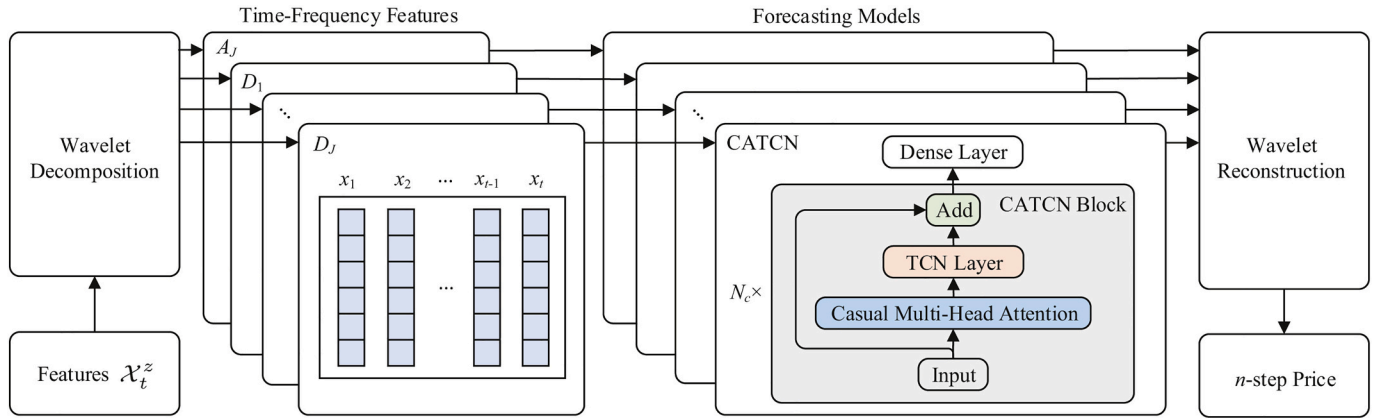

Fig. 4. WT-CATCN architecture.

where $X_{(i,j)} = QK^T$ is a square matrix.

$$\texttt{Attention}(Q, K, V) = \texttt{softmax}\left(\frac{\texttt{Mask}(QK^T)}{\sqrt{d_x}}\right)V \qquad (9)$$

where d_x is the dimension of x, and the $\texttt{softmax}$ function is used to calculate the distribution of attention values. After we calculate the attention values $\texttt{Attention}(Q, K, V)$, the causal multi-head attention

factor (detailed later in the Dilated Convolutions section) of the previous

---

[7] Long memory means that a model is capable of learning and remembering over long sequences of inputs.
[8] After the activation function of the multi-layer neural network, the gradient exponential decay/increase may causes vanishing/exploding gradient.
[9] Neural networks usually run on the GPU (Graphics Processing Unit), convolutional architecture can be better optimized and run parallel in GPU.
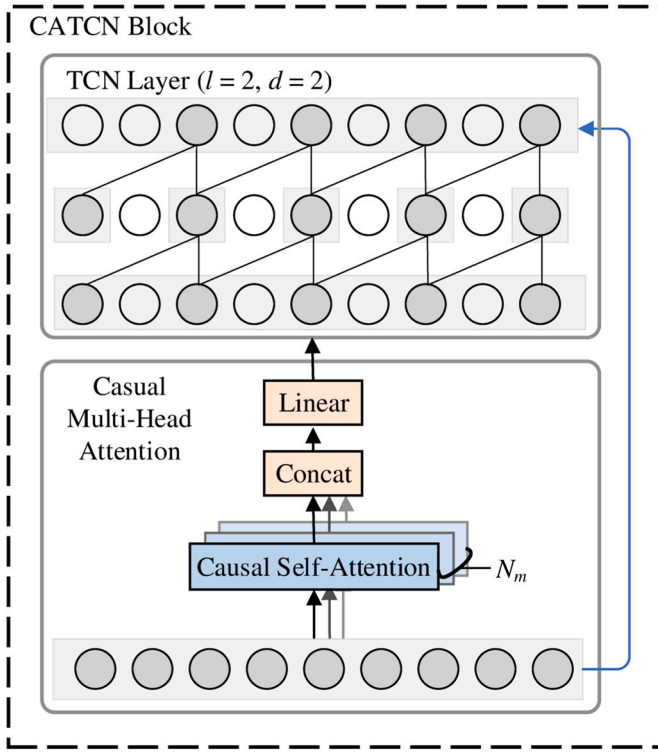
Fig. 5. The CATCN block consists of the TCN layer and multiple causal self-attention layers. Here is an example in which the kernel size l is 2 and the dilated factor d is 2.

layer to increase the receptive field. The dilation factor exponentially increases with the number of the TCN layers, which makes the memory of CATCN exponentially increase. Building the TCN layer in this way guarantees that our model can process long sequence data more efficiently. The sub-layer of the TCN layer is composed of causal and dilated convolutions, which are implemented as follows.

Causal convolutions: Models with causal convolutions can form a function, which is similar to that of recurrent connections of RNNs.

Causal convolution needs the convolution operation to convolve the data before time x_i'. Causal convolutions can be defined as:

$$y_i' = (x' * W)(i) = \sum_{\eta=0}^{l_w - 1} x'_{i-\eta} W(i), \tag{11}$$

where $i = \{1, 2, ..., l_b\}$, W represents a matrix (usually called the kernel) multiplication for causal convolution, and l_w is the kernel size of W. The model with casual convolutions is casual and efficient, but it does not have a large enough receptive field to address a long sequence dataset. To address this concern, we adopt the following dilated convolutions.

Dilated convolutions: Dilated convolutions are used to expand the kernels to increase the receptive field and maintain the size of the kernels, which means that holes are added to the kernel at the same interval controlled by dilation factor d. Dilated convolution can be converted from Eq. (11) as:

$$y_i' = (x' *_d W)(i) = \sum_{\eta=0}^{l_w - 1} v_{i-d\cdot\eta} W(i), \tag{12}$$

where W represents a matrix multiplication for dilated convolution. The receptive field can be increased exponentially by 2 times d with each additional layer.

4.3.4. CATCN block

By stacking causal multi-head attention and the TCN layer, the

proposed CATCN (Casual Multi-Head Attention TCN) block is shown in Fig. 5. The CATCN block has the following characteristics:

Causality. To guarantee the sequence of each step of the time series, the CATCN block is used to processes the input in a comprehensive causality way. In the causal multi-head attention layer, we use the casual mask to invalid the attention values which involve future data. At the same time, the TCN layer uses causal convolutions to address the data from the causal multi-head attention layer. Stacking these two layers allows the CATCN block to focus on important positions while processing the data in a causality approach.

Residual connection. The number of layers in the model increases if the input is a very long sequence. This may cause the degradation of model performance.

Inspired by the gate controls of LSTM, He et al. [47] propose the residual block, which adds an input to the nonlinear transformation of the input. After using the residual block, models with an increasing number of layers show significant improvement. Hence, the residual connection is implemented in our block. The residual block can be defined as:

$$y'' = A(x + \mathcal{F}(x)) \tag{13}$$

where $i = \{1, 2, ..., l_b\}$, x is the input of the residual block, A is activation function and $\mathcal{F}$ is the nonlinear transformation which consists of causal multi-head attention and TCN layers.

The CATCN block can leverage dilated convolutions to enlarge the receptive field; thus, the receptive field of CATCN increases exponentially with the increasing number of CATCN blocks. To prevent a very large receptive field, the number of CATCN blocks needs to be controlled. We denote the number of CATCN blocks as N_c. The receptive field l_r of the last block can be calculated as:

$$l_r = 1 + 2(l_w - 1)(2^{N_c} - 1) \tag{14}$$

In general, the receptive field should cover all the available data; thus, l_r should cover the input of the first block. By fixing l_r, N_c can be calculated according to Eq. (14).

4.3.5. Wavelet reconstruction

For each CATCN, the components of their output can be integrated as $\{\widehat{A_J^p}, \widehat{D_1^p}, ..., \widehat{D_J^p}\}$, which are the components of $\widehat{P_d^n}$. We can use the inverse transform of Eq. (7) to reconstruct $\widehat{P_d^n}$, which is defined as:

$$\Gamma^{-1}(\{\widehat{A_J^p}, \widehat{D_1^p}, ..., \widehat{D_J^p}\}) = \widehat{P_d^n} \tag{15}$$

5. Evaluation and result

In this section, we conduct a series of experiments to evaluate our WT-CATCN model proposed in Section 4.3.

5.1. Experiment setup

5.1.1. Datasets

- **Inter-exchange Transactions.** The inter-exchange transaction data are collected from *WalletExplorer*, one of the most well-known websites monitoring Bitcoin transactions between exchanges. This dataset contains 87 exchanges' transactions from 2016-01-01 to 2018-12-31. The total number of transactions is 63,748,894, and the number of transactions between exchanges is 2,668,529.
- **Inner-exchange Market Prices.** To get the fairest market prices, we collect Bitcoin price data from *CoinMarketCap*, because it prices Bitcoin by obtaining the price of exchange and convert it using the existing reference prices. The market price data contain the daily OHLC and volume from 2016-01-01 to 2018-12-31.

- **Google Trends Data.** Google provides flexible time granularity for its Trends data. In our experiment, we set κ as *Bitcoin* (also used in [29]) to obtain the daily search data from 2016-01-01 to 2018-12-31.

In our experiment, our input data covers almost one year before the forecast starting point. Such one year data can be used as the training dataset. During the training, we randomly select time points to forecast, set T as 60 from the training dataset as input, and forecast the price of the next n-step (n is set as 14 in our experiment). The reason why we set n as 14 is that 14-step is large enough to provide a clear trend of price while remaining small enough to achieve good performance throughout the forecasting sequence [29]. The test results on different time periods forecasting are reported in Tables 6 and 7.

5.1.2. Baselines

In this paper, we compare our proposed **WT-CATCN** with the following baseline and state-of-the-art methods.

- **ARIMA.** The Autoregressive Integrated Moving Average (ARIMA) model is one of the most widely used statistical models in time-series analysis. ARIMA is widely used in the Bitcoin prediction [30,29,12, 10], which takes Bitcoin price as input and predicts the future price.
- **ARIMAX.** This model incorporates additional explanatory variables into ARIMA, which makes it more suitable for non-stationary and multi-variable data. Compared with ARIMA, ARIMAX takes all the features in Table 1 as input.
- **CNN.** Convolutional Neural Networks (CNN) performs well not only in computer vision but also in the financial market, such as in stock forecasting.
- **MLP.** Multilayer Perceptron (MLP) is a typical feedforward artificial neural network, which can be applied in time-series forecasting.
- **LSTM.** Long Short-Term Memory (LSTM) is a variant of a recurrent neural network (RNN) that is widely used in time-series forecasting [10,40].
- **Seq2Seq.** A Sequence to Sequence (Seq2Seq) network consists of an encoder and a decoder. We follow the Seq2Seq network suggested in [29].
- **BNN.** Bayesian Neural Networks (BNN) have been applied in many fields, such as price forecasting [9].
- **SFM.** State-Frequency Memory (SFM) Recurrent Neural Networks have been applied in stock price prediction [40].

5.1.3. Implementation details

In the following experiments, the default setting is as follows. For the wavelet transform part, we use $db4$ wavelet, a typical Daubechies Wavelet, to decompose the $\mathcal{X}_t^z$. Kernel size l_w is set to 2, and the decomposition level J is set to 1 so that there are two components in the output $\{A_1, D_1\}$ with the same length, which makes it easier to choose the same parameters for the models. For the datasets, we use all the datasets by default. The T and n are set to 60 and 14, respectively.

With the length of input T and l_w, we can calculate the N_c as 4 according to Eq. (14). Thus the N_m is set to 4. In general, the learning rate is set to 0.003. We keep the training epochs[10] as 50 and batch size[11] as 8. The model is optimized by performing mini-batch stochastic gradient descent using Adam Optimizer[12] and trained on two NVIDIA GTX 1080Ti GPUs. All the parameters are optimized by using mean square error (MSE) as the loss function.

---

[10] One epoch means the algorithm runs all the samples in the training dataset once.

[11] The number of samples fed into the neural network at one time.

[12] Optimizers are algorithms or methods used to change the attributes of neural networks such as weights and learning rate in order to reduce the losses.

5.1.4. Evaluation metrics

Closeness and direction metrics are used to measure the forecasting of the next price, with the closeness metric capturing the deviation of the forecasting price and the direction metric capturing the trend of the forecasting price [42]. Our aim is to forecast n-step prices; therefore, we need more appropriate metrics of closeness and direction to measure our forecasting.

For closeness metrics, we use Root Mean Square Error(RMSE) to measure the deviation of forecasting prices. Used in many works [10,29, 9] as the standard deviation of the forecasting errors, RMSE can measure the closeness of the forecasting curve, but not the direction of the curve.

Typical closeness metrics to measure the deviation of forecasting prices are listed as follow:

- **MAE.** Mean Absolute Error (MAE) represents the average of the absolute errors between the forecasting result and the true price.
- **MSE.** Mean Squared Error (MSE) measures the average of the squares of the errors between the forecasting result and the true price.
- **RMSE.** Root Mean Square Error (RMSE) measures the deviation of forecasting prices. RMSE has been used in many studies [10,29,9] and is the standard deviation of the forecasting errors.

For the reason that there is no standard metric to measure the trend of n-step forecasting, we consider the following metrics as direction metrics:

- **Fréchet Dist**. Discrete Fréchet Distance is a measure of similarity between curves. It consider the location and order of the points.
- **DTW**. Dynamic Time Warping distance is used to measure the similarity between two sequences.

5.2. Experiment results

5.2.1. Performance

We forecast the Bitcoin price using our WT-CATCN model based on the testing dataset from 2017-06-15 to 2018-09-21 as a randomly selected time period. The performance of our model, traditional models, and deep learning models are presented in Tables 2 and 3.

As shown in Table 2, between the linear models, ARIMA forecasts better than ARIMAX. Although ARIMAX considers more features as input, it achieves worse performance (about 54% in RMSE). One possible explanation is that ARIMAX cannot model the interaction between any two input variables. The price predicted by the linear models is like a straight line (shown in the next section), which is apt to have a small error in most cases, but it cannot reflect the price trend.

The traditional models have limited ability to capture useful information and make precise forecasts. For models like CNN and MLP, their closeness and direction metrics can only reach about 3045 and 2675 on

Table 2

Comparison with baseline methods.

Model	Closeness metric			Direction metric	
	MSE	MAE	RMSE	Fréchet dist	DTW
ARIMA	3.658e+06	1212.321	1392.607	2294.416	5210.660
ARIMAX	9.056e+06	1895.164	2147.586	3371.873	8035.529
CNN	1.570e+07	2925.147	3044.914	4114.364	11393.024
MLP	1.188e+07	2557.760	2675.220	3857.817	10009.758
LSTM	7.865e+06	1995.959	2207.373	3187.193	8259.233
Seq2Seq	7.244e+06	1887.248	2085.580	2986.248	7803.527
BNN	4.551e+06	1518.051	1702.973	2621.925	6371.943
SFM	3.747e+06	1434.790	1608.113	2475.281	6017.006
WT-CATCN	**2.018e+06**	**1044.375**	**1204.091**	**1903.501**	**4505.296**

Bold values are our experimental results (WT-CATCN is our algorithm).

Table 3
Comparison with baseline methods (increased ratio compare to ARIMA).

Model	Closeness metric			Direction metric	
	MSE	MAE	RMSE	Fréchet dist	DTW
ARIMA	1.00	1.00	1.00	1.00	1.00
ARIMAX	2.47	1.56	1.54	1.47	1.54
CNN	4.29	2.41	2.18	1.79	2.19
MLP	3.25	2.11	1.92	1.68	1.92
LSTM	2.15	1.65	1.58	1.39	1.59
Seq2Seq	1.98	1.56	1.50	1.30	1.49
BNN	1.24	1.25	1.22	1.14	1.22
SFM	1.02	1.18	1.15	1.08	1.15
WT-CATCN	**0.55**	**0.86**	**0.86**	**0.83**	**0.86**

Bold values are our experimental results (WT-CATCN is our algorithm).

RMSE, respectively. It is quite challenging for them to forecast prices, since they are not designed for time-series forecasting. For deep learning models, LSTM, BNN, and SFM obtain better forecasting results, with SFM achieving the best performance.

When comparing our model with the representative deep learning models, it is clear that our model significantly outperforms them for both the closeness and direction metrics. For the closeness metrics, our model has a nearly 25% improvement on RMSE than the state-of-the-art SFM

model. The main reason is that SFM cannot model well the multiple frequency components underlying very high fluctuation which is the advantage of our method. For the direction metrics, our model yields the lowest Fréchet dist and DTW measure. In other words, our model not only obtains the smallest deviation but also captures a clear price trend.

5.2.2. Case study

In this subsection, we show our experimental results in different price trend scenarios: **Rising** (prices keep increasing over time), **Falling** (prices keep decreasing over time), **Convex** (prices first increase then decrease over time), **Concave** (prices first decrease then increase over time), and **Flat** (prices are stable over time).

In order to clearly visualize the forecasting prices, we compare our results with the three models among the baselines (ARIMA, SFM, and BNN). The result of each trend is presented in Fig. 6. We find that WT-CATCN can accurately forecast most of the trends. The details are discussed below.

For the rising trend (subfigure (a)), WT-CATCN has the best forecasting result, and its RMSE is about 716. The SFM predicts the right trend curve only at the beginning. BNN is able to forecast the rising trend, but its curves are not as smooth as WT-CATCN's curve. The ARIMA's curve is very flat, which indicates that this model usually cannot capture the rising trend. WT-CATCN is the only model that can
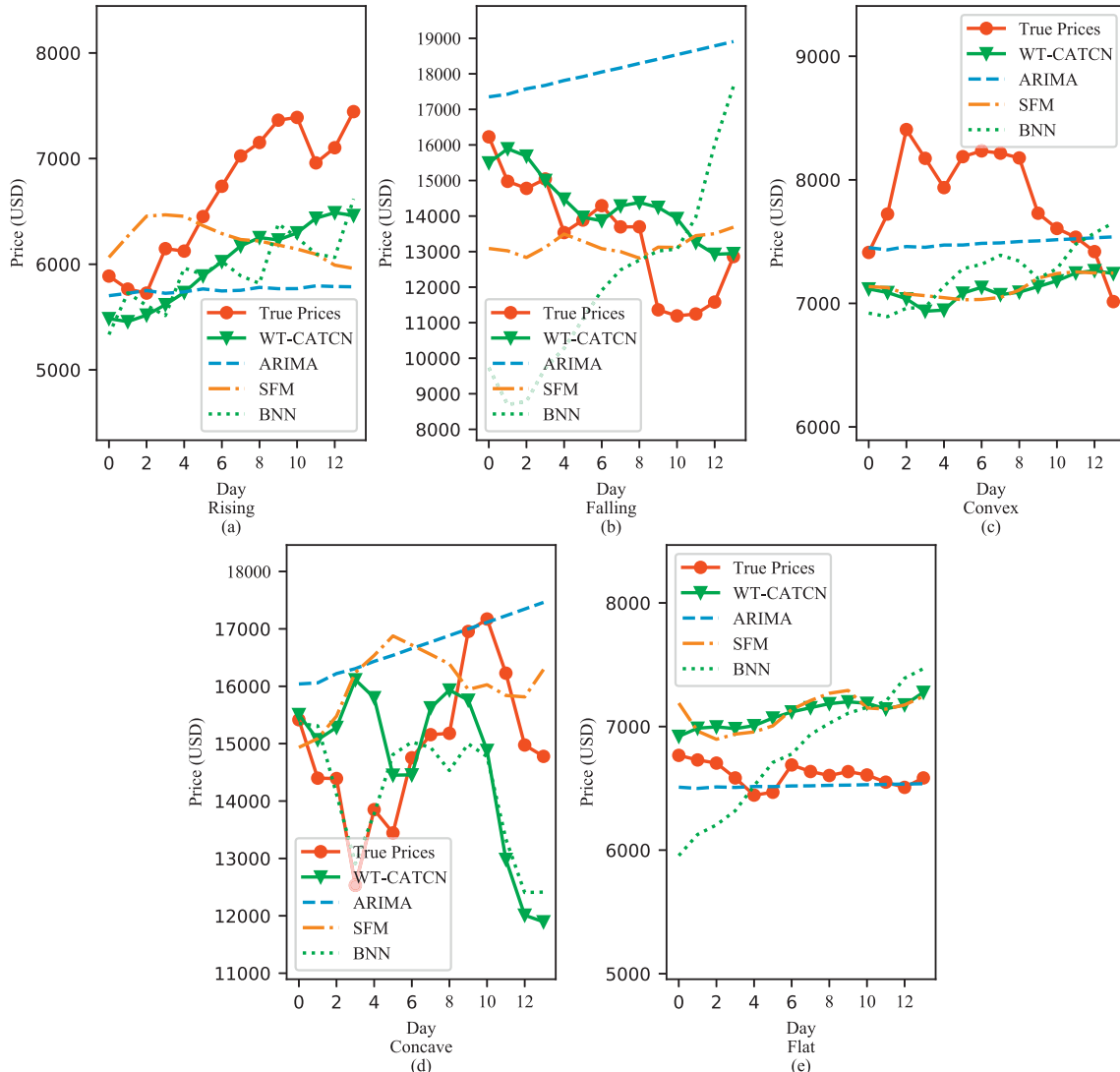
Fig. 6. Comparison of price trends: rising, falling, convex, concave, flat.

capture the falling pattern (subfigure (*b*)) and achieve the minimum RMSE of 1356. None of the models are able to capture the convex pattern (subfigure (*c*)). The forecasting curves are either flat or far away from the true value. WT-CATCN and BNN are better able to capture the concave pattern (subfigure (*d*)) compared to SFM and ARIMA, which forecast the wrong trend. For the flat trend (subfigure (*e*)), WT-CATCN and SFM forecast a closer price trend to the true value, and ARIMA achieves the small RMSE measure.

Furthermore, we conduct experiments to show our model's performance in short-term and long-term forecasting from 2017-10-27 to 2018-02-16 as a randomly selected time period.

Fig. 7 shows the daily forecasting (the short-term) of our model and compare it with the SFM model. The solid curve in the figure is the real price of Bitcoin, and the dashed lines are the forecasting price. We can observe that our forecasting is very close to the actual result. WT-CATCN is more sensitive to price rising and falling, while SFM has a delay effect for the trend during the forecasting. As for the SFM model, its forecasting price is similar to the previous day price and results in the poor forecasting results.

Fig. 8 shows the 14-step forecasting (the long-term). The gray vertical lines split the time range into 14-day regions. For example, the first line indicates that the continuous forecasting started from 2017-10-27 and ended at 2017-11-10. Thus the forecast prices will break at two adjacent regions. It is always very challenging to forecast in long term, but our method is able to forecast better trends than the state-of-the-art SFM model as shown in the figure.

From the above two figures, we conclude that our model has high forecasting ability for both short term and long term forecasting, is able to capture the price rising and falling trends, and has better performance on the concave trend compared to the SFM and BNN.

5.2.3. Model ablation

To investigate the role of different components of WT-CATCN on the forecasting performance, we compare our model with the original TCN, WT-TCN (TCN with wavelet transform), and WT-ATCN (disabling the causal mask of the CATCN block) using the testing dataset.

We can observe from Table 4 that, if we only leverage TCN, its RMSE is about 1700. After applying wavelet transform to TCN, WT-TCN achieves better performance (RMSE 1588) than TCN (RMSE 1700), with an approximately 6.5% improvement on RMSE. This improvement is possible because wavelet transform can effectively alleviate the non-stationary data, which greatly improves the forecasting accuracy. We can also see that the design of using different networks to predict the time-frequency features can help to improve performance. WT-ATCN incorporates multi-head attention based on WT-TCN, which can decrease the RMSE by 14%. The result shows that attention is a useful mechanism and plays a critical role in our model. To demonstrate the importance of the casual mask, we compare the two models with and
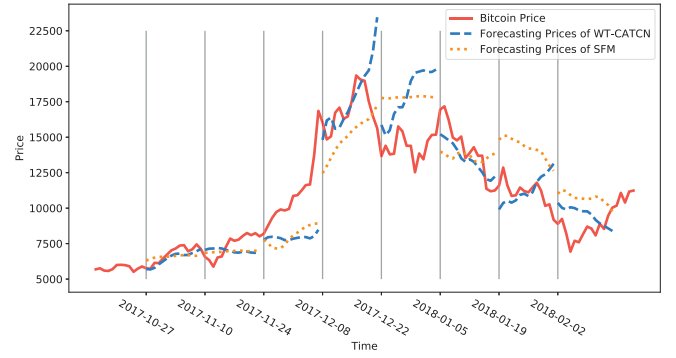

Fig. 8. The 14-days (long-term) forecasting curve of WT-CATCN and SFM from 2017-10-27 to 2018-02-16.

Table 4

Model comparison. WT-TCN is the TCN with wavelet transform. WT-ATCN is the model WT-CATCN disabling the causality of attention.

Model	Closeness metric			Direction metric	
	MSE	MAE	RMSE	Fréchet dist	DTW
TCN	6.092e+06	1552.139	1699.961	2582.408	6360.673
WT-TCN	4.288e+06	1419.110	1588.554	2496.086	5943.826
WT-ATCN	2.791e+06	1213.885	1371.706	2138.074	5132.457
WT-CATCN	**2.018e+06**	**1044.375**	**1204.091**	**1903.501**	**4505.296**

Bold values are our experimental results (WT-CATCN is our algorithm).

without this layer, and we find that WT-CATCN utilizing causal multi-head attention can improve the performance by 12% compared to WT-ATCN. This explains why causality is important in the networks when we design WT-CATCN.

To summarize, each component of our model contributes to the improvement of the forecasting performance. Specifically, the causality of WT-CATCN brings about 12% improvement over WT-ATCN. The causal multi-head attention layer can bring about 24% improvement compared to WT-TCN. Compared to the original TCN, by adding wavelet transform and causal multi-head attention, WT-CATCN improves the forecasting performance by 29%.

5.2.4. Impact of key parameters

In this subsection, we investigate the impact of two key parameters: the selection of different wavelets and the number of casual multi-head attention N_m.

With different decomposition and reconstruction processes, various types of wavelets have different impacts on the result of *n*-step forecasting. Therefore, we first investigate how wavelet type affects the
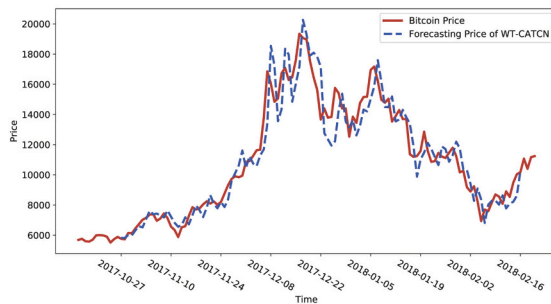
(a) Daily forecasting of WT-CATCN.
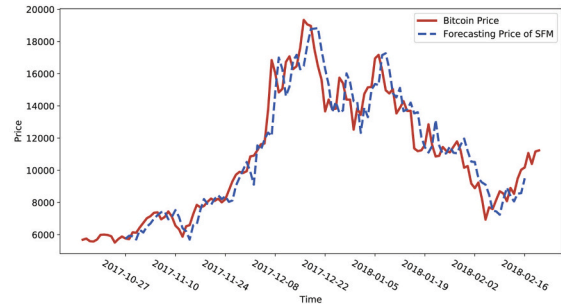
(b) Daily forecasting of SFM.

Fig. 7. The daily forecasting curve of WT-CATCN and SFM from 2017-10-27 to 2018-02-16. The daily forecasting (short-term) result of our method WT-CATCN has 4.5% better than SFM.

forecasting results. We test the popular wavelets *db*1, *db*4 and *db*8, and vary the *n* from 2 to 28. The results are presented in Table 5.

We observe that the forecasting performance decreases (with increasing RMSE) as *n* increases. In general, *db*4 is better than *db*8 and *db*1, especially when *n* is less than 14. Considering that short-term forecasting is usually used by common users and the performance difference is very small when *n* is larger than 14, we set *db*4 as the default in our experiments.

Second, we further investigate the impact of N_m (the number of the casual self-attention layers) on the forecasting results. We set N_m to 1, 4 and 8, and their impacts on forecasting performance (RMSE) are 1215.39, 1204.09, and 1232.85, respectively. It is clear that the 4-head model achieves better performance and brings more improvement than the 1-head model. One possible explanation for this is that more heads bring additional useful information to our model, which leads to more accurate forecasting. Furthermore, we find that the 4-head model is also better than the 8-head model. A possible explanation is that more heads make the model focus on more information sources than needed, resulting in distractions when the information is fused.

6. Scalability

To investigate the scalability of our model, such as the number of days we can perform accurate forecasting, we conduct the *n*-step experiment by applying different time lengths. First, we test the number of *n* from 1 to 30. The experiment results are demonstrated in Fig. 9. We observe that the error of WT-CATCN increases slowly as the *n* increases until *n* > 18. After 18 days, the error increases with a faster speed than before. This is possible because there are too many uncertainties associated with future prices, and forecasting based on previous data leads to a larger error.

Second, we test other different time granularity combinations of *n*-step and *T*-point. We summarize the experimental results in Table 6. Enlarging the training data time granularity (*T*-point) does not improve the forecasting performance. However, the forecasting performance decreases when the testing data time granularity (*n*-step) increases.

To summarize, the result of forecasting within 18 days is the most accurate. If the query time period is longer than this, the forecasting performance decreases. Therefore, it is preferable for users to choose a value of *n* smaller than 18 when they use our proposed model to forecast Bitcoin prices. Moreover, we find that enlarging the training data granularity does not improve the performance.
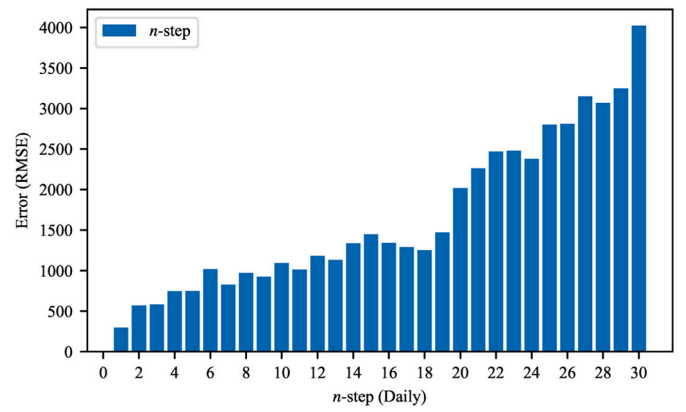
To find out the impact of the length of training data *T*, we test the number of *T* from 20 to 100; results of the experiment are shown in Table 7. The SFM model is excluded from this experiment, as it is trained on all the training sequences [40]. We find that 60 and 80 of *T*-points are good parameters for WT-CATCN. The possible reasons for this are as follows. A short *T* value cannot capture the important data movement patterns in the historical data, whereas a long *T* value may increase the difficulty associated with finding the important information hidden in massive input data. Therefore, in our later experiment, we set the *T* as 60 by default.

Fig. 9. Forecasting error (RMSE) for *n*-step ($n = \{1, 2, …, 30\}$) by WT-CATCN.

Table 6
The impact of different time granularity (*T*-point and *n*-step) on forecasting performance. *T* represents the length of input data of our model, and *n* represents the length of prices the model will predict.

(*T*-point) – (*n*-step)	*n*			
	1	2	3	4
60 days – *n* days	295.630	571.784	583.040	747.191
60 days – *n* weeks	1060.331	1487.104	2337.846	3717.012
60 days – *n* months	3495.360	3875.256	4097.356	5171.829
8 weeks – *n* weeks	1189.644	1723.072	2990.677	4223.840

Table 7
The impact of different *T*-point (daily) on the forecasting performance (RMSE) of the 14-step prediction.

	T				
	20	40	60	80	100
ARIMA	2164.77	2164.77	1392.61	2164.77	2164.77
ARIMAX	2911.10	2911.1	2147.59	2911.10	2911.1
CNN	1885.96	2645.37	3044.91	3334.99	3429.67
MLP	2705.88	2668.73	2675.22	2599.30	2662.09
LSTM	2176.54	2820.98	2207.37	2235.02	1785.23
Seq2Seq	1835.37	2265.31	2799.8	2600.29	2952.03
BNN	2679.78	3013.07	1702.97	4163.76	5631.86
TCN	**1244.52**	**1320.71**	1699.96	1564.68	1448.34
WT-TCN	2443.00	2309.91	1588.55	3054.65	3967.12
WT-CATCN	1651.83	1417.28	**1204.09**	**1289.77**	**1395.33**

Bold values are the best performance results when we compare different parameters.

6.1. Feature importance

Using the testing dataset, we investigate the role of the three features (market prices features, inter-exchange transaction features, and social interest features) on the forecasting performance. We summarize the experimental results in Table 8. Our model achieves the best performance when we consider all three types of features and outperforms the

Table 5
The impact of different wavelets on forecasting *n*-step price performance (RMSE).

	n				
	2	7	14	21	28
*db*1	**572.093**	1103.132	1382.069	1859.933	2465.548
*db*4	605.388	**883.728**	**1204.091**	2087.809	3361.800
*db*8	659.412	948.925	1226.658	**1765.025**	**2440.552**

Bold values are the best performance results when we compare different parameters.

Table 8
The impact of different types of features on forecasting performance. We use *M*, *X*, *B* to represent market prices data, inter-exchange transactions data, social interest data respectively.

Features	*M*	*M + X*	*M + B*	*M + X + B*
Metric (RMSE)	1670.395	1350.864	1294.480	**1204.091**

Bold values are the best performance results when we compare different parameters.

model with only the market price features by about 28%.

To be more specific, the RMSE is 1670.395 when the input data contain only market price features. However, the performance is significantly improved, by 19%, when the input data contain both market price features and inter-exchange transaction features (the RMSE is 1350.864). Similarly, when the input features contain both market price features and social interest features, the performance can be improved by 22%.

7. Conclusion and future work

Despite the fact that the importance of cryptocurrency price forecasting has been recognized by both financial market practitioners and researchers [4], how to better predict cryptocurrency prices given their highly volatile nature remains a vexing problem. In this paper, we focus on Bitcoin, the most popular cryptocurrency, and aim to solve this problem by (1) identifying the potential features that have impacts on the Bitcoin price from the perspective of underlying blockchain transactions; and (2) examining the proposed model's performance with regard to Bitcoin price forecasting.

First, although factors such as user comments and blockchain features have been examined in previous literature to test their impacts on Bitcoin price forecasting [8–10], there is still a lack of careful consideration of the potentially involved factors. To address this concern, we integrate factors including the transaction volumes of inter-exchange transactions, the market prices of inner-exchange, and the Google Trend search data that capture social interest.

Second, we propose a deep learning method called WT-CATCN. By leveraging Wavelet Transform (WT) and Casual multi-head Attention (CA) in the Temporal Convolutional Network (TCN), we are able to accurately forecast time-series data with high fluctuation (e.g., Bitcoin price), because our model is able to focus on the key parts of the input data while also modeling the correlations among the data.

Third, we conduct a series of experiments based on real-world data to show that our model can effectively forecast the Bitcoin price and capture its movement patterns (e.g., rising and falling), even in the highly fluctuating situation. To find appropriately complex models and meet the requirement of accuracy, we compare our model with various existing models, and the experiment results show that our WT-CATCN model outperforms the state-of-the-art models by at least 25% in terms of the RMSE metric.

Our research fills an important gap in the information systems literature on blockchain technology and cryptocurrency by proposing and validating a model that facilitates better price forecasting for cryptocurrencies in general and Bitcoin in particular. First, we analyze the features which may affect the Bitcoin price, and we find that the volume difference between big and small exchanges contributes significantly to the price forecasting. This finding is also of great value for financial market practitioners because it helps to broaden the consideration set for variables used when these practitioners make investment decisions on the cryptocurrency market. Second, the model we proposed in this paper, WT-CATCN, has been validated using real-world Bitcoin data and provides a rigorous alternative model when the traditional models fail to predict prices with high volatility. The managerial implications of our work are obvious, as firms can apply our proposed methodology to enhance their cryptocurrency price forecasting performance. As a result, the problem of predicting cryptocurrency prices with a high volatilely nature is alleviated, customer satisfaction is increased, and firms achieve sustainable competitive advantages.

For future work, first, we have validated our model using Bitcoin's historical transaction data in the current research. In addition to Bitcoin, there are other cryptocurrencies available in the market, such as Ethereum and Litecoin. To ensure the generalizability of our model, we need to evaluate it using other cryptocurrencies' price data. Following the same logic, instead of daily data, we should also investigate the price forecasting performance of our model using a more granular dataset,

such as 15-min or hourly data. Third, it would be interesting to investigate more algorithms that can improve the price forecasting performance of our model. For instance, it may help our model to focus on the useful positions if we increase the number of network layers to build a deeper neural network or design a more concise and efficient attention function. Finally, cryptocurrency market practitioners will receive the most benefit from this study if we deploy our model in an online forecasting tool in the near future.

Acknowledgements

References

[1] S. Ransbotham, R.G. Fichman, R. Gopal, A. Gupta, Special section introduction-ubiquitous it and digital vulnerabilities, Inf. Syst. Res. 27 (4) (2016) 834–847.

[2] P. Constantinides, O. Henfridsson, G.G. Parker, Introduction-Platforms and Infrastructures in the Digital Age, 2018.

[3] S.F. Cheng, G. De Franco, H. Jiang, P. Lin, Riding the blockchain mania: public firms' speculative 8-k disclosures, Manage. Sci. 65 (12) (2019) 5901–5913.

[4] Z. Chen, C. Li, W. Sun, Bitcoin price prediction using machine learning: an approach to sample dimension engineering, J. Comput. Appl. Math. 365 (2020) 112395.

[5] X. Li, C.A. Wang, The technology and economic determinants of cryptocurrency exchange rates: the case of bitcoin, Decis. Support Syst. 95 (2017) 49–60.

[6] J. Li, Y. Yuan, F.-Y. Wang, A novel GSP auction mechanism for ranking bitcoin transactions in blockchain mining, Decis. Support Syst. 124 (2019) 113094.

[7] F. Mai, Z. Shan, Q. Bai, X. Wang, R.H. Chiang, How does social media impact bitcoin value? A test of the silent majority hypothesis, J. Manage. Inf. Syst. 35 (1) (2018) 19–52.

[8] Y.B. Kim, J.G. Kim, W. Kim, J.H. Im, T.H. Kim, S.J. Kang, C.H. Kim, Predicting fluctuations in cryptocurrency transactions based on user comments and replies, PLoS One 11 (8) (2016).

[9] H. Jang, J. Lee, An empirical study on modeling and prediction of bitcoin prices with Bayesian neural networks based on blockchain information, IEEE Access 6 (2017) 5427–5437.

[10] T. Guo, A. Bifet, N. Antulov-Fantulin, Bitcoin volatility forecasting with a glimpse into buy and sell orders, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 989–994.

[11] V. Dhar, T. Geva, G. Oestreicher-Singer, A. Sundararajan, Prediction in economic networks, Inf. Syst. Res. 25 (2) (2014) 264–284.

[12] S. McNally, J. Roche, S. Caton, Predicting the price of bitcoin using machine learning, in: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), IEEE, 2018, pp. 339–343.

[13] W. Kristjanpoller, M.C. Minutolo, A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis, Expert Syst. Appl. 109 (2018) 1–11.

[14] P. Giudici, I. Abu-Hashish, What determines bitcoin exchange prices? A network VAR approach, Finance Res. Lett. 28 (2019) 309–318.

[15] I. Makarov, A. Schoar, Trading and arbitrage in cryptocurrency markets, J. Financ. Econ. 135 (2) (2020) 293–319.

[16] J.M. Griffin, A. Shams, Is bitcoin really untethered? J. Finance 75 (4) (2020) 1913–1964.

[17] A. Grossmann, J. Morlet, Decomposition of hardy functions into square integrable wavelets of constant shape, SIAM J. Math. Anal. 15 (4) (1984) 723–736.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[19] S. Bai, J.Z. Kolter, V. Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, 2018 arXiv preprint arXiv: 1803.01271.

[20] H.H. Sun Yin, K. Langenheldt, M. Harlev, R.R. Mukkamala, R. Vatrapu, Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain, J. Manage. Inf. Syst. 36 (1) (2019) 37–73.

[21] P. Katsiampa, Volatility estimation for bitcoin: a comparison of garch models, Econ. Lett. 158 (2017) 3–6.

[22] D.C. Mallqui, R.A. Fernandes, Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques, Appl. Soft Comput. 75 (2019) 596–606.

[23] S. Alonso-Monsalve, A.L. Suárez-Cetrulo, A. Cervantes, D. Quintana, Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators, Expert Syst. Appl. 149 (2020) 113250.

[24] E. Stenqvist, J. Lönnö, Predicting Bitcoin Price Fluctuation With Twitter Sentiment Analysis, 2017.

[25] T.R. Li, A. Chamrajnagar, X. Fong, N. Rizik, F. Fu, Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model, Front. Phys. 7 (2019) 98.

[26] S. Mohapatra, N. Ahmed, P. Alencar, KryptoOracle: a real-time cryptocurrency price prediction platform using twitter sentiments, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 5544–5551.

[27] G. Cheuque Cerda, J.L. Reutter, Bitcoin price prediction through opinion mining, in: Companion Proceedings of the 2019 World Wide Web Conference, 2019, pp. 755–762.

[28] A. Jain, S. Tripathi, H. DharDwivedi, P. Saxena, Forecasting price of cryptocurrencies using tweets sentiment analysis, in: 2018 Eleventh International Conference on Contemporary Computing (IC3), IEEE, 2018, pp. 1–7.

[29] J. Rebane, I. Karlsson, S. Denic, P. Papapetrou, Seq2Seq RNNs and ARIMA models for cryptocurrency prediction: a comparative study, SIGKDD Fintech 18 (2018).

[30] M. Amjad, D. Shah, Trading bitcoin and online time series prediction, in: NIPS 2016 Time Series Workshop, 2017, pp. 1–15.

[31] P.R. Hansen, A. Lunde, A forecast comparison of volatility models: does anything beat a GARCH (1, 1)? J. Appl. Econom. 20 (7) (2005) 873–889.

[32] M. Hagenau, M. Liebmann, D. Neumann, Automated news reading: Stock price prediction based on financial news using context-capturing features, Decis. Support Syst. 55 (3) (2013) 685–697.

[33] K. Nam, N. Seong, Financial news-based stock movement prediction using causality analysis of influence in the Korean stock market, Decis. Support Syst. 117 (2019) 100–112.

[34] M.R. Hassan, K. Ramamohanarao, J. Kamruzzaman, M. Rahman, M.M. Hossain, A HMM-based adaptive fuzzy inference system for stock market forecasting, Neurocomputing 104 (2013) 10–25.

[35] R.K. Nayak, D. Mishra, A.K. Rath, A naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices, Appl. Soft Comput. 35 (2015) 670–680.

[36] M. Lam, Neural network techniques for financial performance prediction: integrating fundamental and technical analysis, Decis. Support Syst. 37 (4) (2004) 567–581.

[37] Y. Jiao, J. Jakubowicz, Predicting stock movement direction with machine learning: an extensive study on s&p 500 stocks, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE, 2017, pp. 4705–4713.

[38] R. Faccini, E. Konstantinidi, G. Skiadopoulos, S. Sarantopoulou-Chiourea, A new predictor of us real economic activity: the s&p 500 option implied risk aversion, Manage. Sci.nce 65 (10) (2019) 4927–4949.

[39] C.J. Neely, D.E. Rapach, J. Tu, G. Zhou, Forecasting the equity risk premium: the role of technical indicators, Manage. Sci. 60 (7) (2014) 1772–1791.

[40] L. Zhang, C. Aggarwal, G.-J. Qi, Stock price prediction via discovering multi-frequency trading patterns, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 2141–2149.

[41] H.K. Sul, A.R. Dennis, L. Yuan, Trading on twitter: using social media sentiment to predict stock returns, Decis. Sci. 48 (3) (2017) 454–488.

[42] Q. Li, L. Jiang, P. Li, H. Chen, Tensor-based learning for predicting stock movements, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[43] L. Kristoufek, Bitcoin meets google trends and wikipedia: Quantifying the relationship between phenomena of the internet era, Sci. Rep. 3 (2013) 3415.

[44] A. Aghajani, R. Kazemzadeh, A. Ebrahimi, A novel hybrid approach for predicting wind farm power production based on wavelet transform, hybrid neural networks and imperialist competitive algorithm, Energy Convers. Manage. 121 (2016) 232–240.

[45] O. Rioul, P. Duhamel, Fast algorithms for discrete and continuous wavelet transforms, IEEE Trans. Inf. Theory 38 (2) (1992) 569–586.

[46] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 674–693.

[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

Haizhou Guo is currently a postgraduate student in the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include data analytics and blockchain.

Dian Zhang received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010. She is currently an associate professor at Shenzhen University. Her research interests include big data analytics and mobile computing. She has published many papers in top tier journals including IEEE Transaction on Parallel Computing, Transaction on Mobile Computing. She is the PI of many projects including National Natural Science Foundation of China (NSFC). She received 2nd Class Academic Research Outstanding Award in Nature Science from the Ministry of Education of China in 2019.

Siyuan Liu is currently an Assistant Professor of Information Systems at the Smeal College of Business, Pennsylvania State University. His current research interests include big data for business intelligence, social networks analytics, and mobile marketing. He has published research papers in top tier journals and conferences, including Management Science, Transportation Research: Part B, IEEE Transactions on Knowledge and Data Engineering, ACM Transactions on Knowledge Discovery from Data, IEEE Transactions on Big Data, AAAI, ACM SIGKDD, and ACM SIGMOD. His work has been recognized by Google Internet of Things Technology Research Award, Google Faculty Research Award, Marketing Science Institute Award, CPIC Research Achievement Award, etc.

Lei Wang is an Assistant Professor of Information Systems at the Smeal College of Business, Pennsylvania State University. She received her Ph.D. degree in Operations and Information Management from the University of Connecticut along with a master's degree in Economics and a bachelor's degree in Electronic Engineering. Her research interests include location-based services, FinTech, gamification and user engagement on digital platforms. She was the recipient of the Nunamaker-Chen Dissertation Award in 2015. Her research has published in premium outlets including MIS Quarterly, Decision Support Systems, and conference proceedings such as ICIS and HICSS. She serves as Associate Editor for ICIS and PACIS, and has served as reviewer for leading Information Systems journals including ISR, MISQ, POMS, JMIS and DSS.

Ye Ding received his Ph.D. degree in 2014 supervised by Prof. Lionel M. Ni from the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. He is currently an Associate Professor in School of Cyberspace Security at Dongguan University of Technology. His research interests are spatial-temporal data analytics, big data, and machine learning. He is a member of IEEE since 2013.