Cryptocurrencies Price Prediction Using Weighted Memory Multi-Channels

Zhuorui Zhang, Junhao Zhou, Yanan Song, and Hong-Ning Dai

Macau University of Science and Technology
zhuorui.zhang@qq.com; junhao_zhou@qq.com; ynsong@must.edu.mo; hndai@ieee.org

Abstract. After the invention of Bitcoin and a peer to peer electronic cash system based on the blockchain, the market of cryptocurrencies increases rapidly and attracts substantial interest from investors and researchers. Cryptocurrencies price volatility prediction is a challenging task owing to the high stochasticity of the markets. Econometric, machine learning and deep learning models are investigated to tackle the stochastic financial prices fluctuation and to improve the prediction accuracy. Although the introduction of exogenous factors such as macro-financial indicators and blockchain information helps the model prediction more accurately, the noise and effects from markets and political conditions are difficult to interpret and modelling. Inspired by the evidence of strong correlations among cryptocurrencies examined in previous studies, we originally propose a Weighted Memory Channels Regression (WMCR) model to predict the daily close price of cryptocurrencies. The proposed model receives time series of several heavyweight cryptocurrencies price and learns the interdependencies of them by recalibrating the weights of each sequence wisely. Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) components are exploited to establish memory and extract spatial and temporal features. Moreover, regularization methods including kernel regularizers and bias regularizers and Dropout method are exploited to improve the generalization ability of the proposed model. A battery of experiments are conducted in this paper. The results present that the WMCR model achieves the state-of-art performance and outperforms other baseline models.

Keywords: Blockchain; Cryptocurencies price prediction; Weighted memory channels; Convolutional neural network; Long short-term memory; Deep learning

1 Introduction

The market capitalization of cryptocurrencies has been growing rapidly in recent years. As one of the best known cryptocurrencies, Bitcoin was invented in 2008 and defined as a peer to peer electronic cash system without central bank and administrators [23]. At the beginning of 2017, the Bitcoin market exceeded 10 billion dollars and rapidly hit 300 billion dollars by December 2017.

Remarkably, with the unmatched advantages of transaction security, decentralization and transparency [29], cryptocurrencies as a new kind of digital assets, have received extensive attention than traditional currencies. Due to the significant value, the volatility prediction of cryptocurrencies has attracted lots of investors and researchers [1,21]. As a kind of virtual asset, the price of cryptocurrencies is influenced by many factors such as fake news, market manipulation and government regulation. Developing a model with high prediction accuracy helps investors make profits and reduce loss. Meanwhile, a sharp rise or fall in prediction reminds the investors to be aware of large price fluctuations in the short-term.

Cryptocurrencies such as Bitcoin are unique assets. Meanwhile, the price fluctuated characteristics of them are similar to both a typical commercial resource and a speculative asset [19]. The impact of facors such as trading volume on cryptocurrencies is complex and subjects to time [27]. Considering the high degree of uncertainty and stochasticity of financial forecasting, previous studies proposed various approaches. These approaches mainly include time series modelling based on historical price and modelling with exogenous drivers such as macro-economic indicators [28]. Conventional interpretable time series models including Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) are leveraged to predict Bitcoin volatility by using the historical prices as the primary information [5,11]. Moreover, to tackle the intractable random changes of financial price, machine learning [21] and deep learning models are proposed in various sectors such as coal price, precious metal price and stock price prediction [2,20,30]. Neural networks with elements from Twitter and market data as input features by using sentiment analysis [11] and Bayesian neural networks based on blockchain information [16] are construct to study the latent driven factors.

Albeit numerous approaches and regression models for cryptocurrency prediction have been developed, the prediction accuracy of models with the only use of price time series has encountered bottleneck. In the context of having few systematic analysis on complex exogenous factors such as political and economic conditions, a natural problem arises: what kernel features are worth taking into account and how to extract those features when forecasting the cryptocurrencies price volatility? In [16], researchers find the evidence that the price of Bitcoin is mainly affected by the information of blockchain directly involved in fund and trade of Bitcoin rather than other macro-financial markets. Studies in the correlations among cryptocurrencies have been conducted in which the correlations are examined to be positive [4]. In particular, some authors investigated the price leadership dynamics of Ethereum (ETH) and Bitcoin (BTC). The result indicates that it has a lead-lag relationship between ETH and BTC [26] while other researchers examined the interdependencies between BTC and altcoin markets in short period and long period [9]. Although the interdependencies between cryptocurrencies have been investigated in different articles, few solutions are available to leverage and to modelling the correlations among cryptocurrencies.

Aiming to fully exploit the latent interdependencies between cryptocurrencies and recalibrate the importance of each cryptocurrency, we propose and develop a Weighted Memory Channels Regression model (WMCR model). Our proposed model is adapt to learn the correlations among several heavyweight cryptocurrencies price and extract both temporal and spatial features of time series after constructing the weighted memory channels. Inspired by the power of Long short-term memory (LSTM) in sequence learning [14] and the approach of dynamic channel-wise feature recalibration in Squeeze-and-Excitation Networks adapt for image processing task [15], we adopt an LSTM layer for each time series of cryptocurrency (which are also regarded as channels) to construct memory. Next we recalibrate the weights of channels by construct a Multi-Channel Weighting block. In addition, convolutional neural networks with kernel regularizers and bias regularizers are employed to extract temporal and spatial features. Non-linear activation functions such as ReLU [22], sigmoid and tanh have been exploited to add non-linear factors. These functions help to solve more complex problems and enhance the representation and the learning ability of neural networks. The implementation of our proposed model is based on Keras [7]. The main research contributions of this paper are summarized as follows:

- We originally propose a regression model based on deep learning framework to predict the daily close price of cryptocurrencies. It is worth mentioning that the WMCR model is efficient at modelling the non-linear correlations between cryptocurrencies dynamically and wisely, this process is novel and has not been used in other studies. It is also capable of extracting temporal and spacial features and establishing memories in short and long term.
- We compare the prediction accuracy and performance of our proposed model with a battery of econometric, deep learning and machine learning models. The WMCR model outperforms all baseline mdoels in interpretable and commonly used evaluation metrics.
- We investigate the effect of three significant parameters on the WMCR model. Among these parameters, the number of convolutional neural network (CNN) layers impacts both the training loss and the validation loss noticeably. In addition, the factors of window length and number of neurons in each hidden layer mainly influence the convergence speed of validation loss.

2 Methodology

2.1 Model Overview

Based on the deep learning framework, we design a WMCR model to predict closing price of cryptocurrencies. The architecture of WMCR model is illustrated in Fig. 1 and the components of the WMCR model are listed in the following paragraphs.

LSTM layers for each channel: To address the exploding and vanishing gradient problems in training the recurrent neural networks, LSTM is proposed

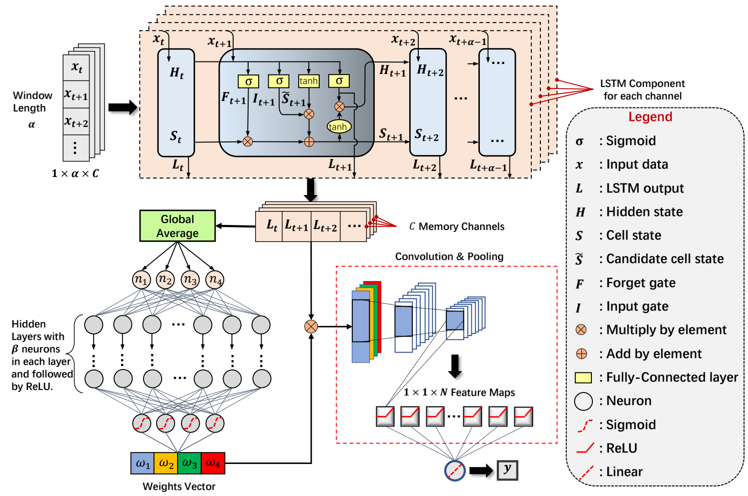


Fig. 1. The architecture of WMCR model.

by [14] and developed by [12]. Since the inputs of WMCR model consist of time series of several cryptocurrencies price after preprocessing, we employ an independent LSTM layer for each channel to filter the noises and memorize the important information of different sequences. In essence, this component can extract temporal features for each channel. In the LSTM component depicted in Fig. 1, the input gate I_t, forget gate F_t, LSTM output L_t, cell state S_t and candidate cell state $\widetilde{S}_t$ are computed according to the following equations:

$$I_t = \sigma(\mathbf{W}_i x_t + \mathbf{W}_{hi} x_t + b_i), \tag{1}$$

$$F_t = \sigma(\mathbf{W}_f x_t + \mathbf{W}_{hf} x_t + b_f), \tag{2}$$

$$L_t = \sigma(\mathbf{W}_l x_t + \mathbf{W}_{hl} x_t + b_l), \tag{3}$$

$$S_t = F_t \odot S_{t-1} + I_t \odot \tilde{S}_t, \tag{4}$$

$$\widetilde{S}_t = \tanh(\mathbf{W}_s x_t + \mathbf{W}_{hs} H_{t-1} + b_s), \tag{5}$$

where the weights matrices $\mathbf{W}$ are in $\mathbb{R}^{h\times\alpha}$. In addition, b_i, b_f, b_l, b_s are vectors of bias terms in $\mathbb{R}^{h\times 1}$. Moreover, σ is the sigmoid activation function.

Multi-Channel Weighting block: Inspired by the SENet architecture establishing weights for feature maps presented by authors in [15], we design a Multi-Channel Weighting block. This block is applicable for discriminating the importance of different time series. Given C channels, the Multi-Channel block starts with a global average pooling layer for each channel of L_i and generates C neurons. We denote the k-th channel of L_i by L_i^k. The value of the k-th neuron is n_k which is given by:

$$n_k = \frac{1}{\alpha}\sum_{i=1}^{\alpha} L_i^{(k)}. \tag{6}$$

However, C neurons obtained via the preliminary average pooling are not enough to construct flexible weights for each channel. We next increase the dimension in hidden layers. In each hidden layer, there are β neurons followed by the ReLU [22] activation. The output $h^{(i)}$ of the i-th hidden layer is given by:

$$\begin{aligned} h^{(i)} &:= \left[h_1^{(i)}\ h_2^{(i)}\ \cdots\ h_\beta^{(i)}\right] \\ &:= \mathrm{ReLU}(h^{(i-1)}\mathbf{W}), \end{aligned} \tag{7}$$

where W can be expressed as follows:

$$\mathbf{W} := \begin{bmatrix} \mathbf{W}_{1,1} & \mathbf{W}_{1,2} & \cdots & \mathbf{W}_{1,\beta} \\ \mathbf{W}_{2,1} & \mathbf{W}_{2,2} & \cdots & \mathbf{W}_{2,\beta} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{\beta,1} & \mathbf{W}_{\beta,2} & \cdots & \mathbf{W}_{\beta,\beta} \end{bmatrix}. \tag{8}$$

In our model, the block contains 3 hidden layers while the weights vector contains 4 elements to perform the weighting for 4 channels correspondingly. In Fig. 1, the weights vector ω is given by the following equation:

$$\omega := \left[\omega_1\ \omega_2\ \omega_3\ \omega_4\right] := \sigma(h^{(3)}\mathbf{W}^*), \tag{9}$$

where W^* is a weights matrix in $\mathbb{R}^{\beta \times C}$.

With the obtained weights vector ω, we multiply the original channels by elements of ω_k in the weight vector to rescale the memory channels generated by LSTM component. The rescaling channel $L^{*(k)}$ can be computed by $L^{*(k)} = \omega_i L^{(k)}$ and different k represents different channel. This block in WMCR model is responsible for modelling the nonlinear correlations between channels and identifying a reasonable weight vector for rescaling. While the interdependencies between different cryptocurrencies are challenging to identify, this block provides a succinct multi-channel weighting solution that is instrumental in improving prediction accuracy.

CNN Layers: The CNN is a representative feed-forward neural network in the field of deep learning. The CNN architecture brings the ability of representation learning. Motivated by the power of extracting spatial features and reducing parameters in CNN, we exploit a component mainly containing two CNN layers to extract the price characteristics of cryptocurrencies. In the second CNN layer, We double the number of convolution kernels to extract the features more sufficiently.

Fully-Connected Layer: We leverage a fully-connected layer to reduce dimensions by concatenating numerous feature maps into a single output neuron. This layer finally receives all the feature maps generated by CNN layers and connects with a single output neuron. After a linear activation function, the WMCR model ultimately outputs the price of prediction of the target cryptocurrency.

2.2 Weights and Memory Establishing for Multi Price Series

In finance, time series regression modeling is a challenging problem caused by the high stochasticity of market and the complex dependencies between latent driven factors. Inspired by the previous studies of relationships between cryptocurrencies and the SENet architecture, we design a WMCR model to adapt for modelling interdependencies between cryptocurrencies prices. As depicted in Fig. 1, the number of channels with α time steps in each sample is denoted by C, representing as channels. First, to remember the critical features for each channel, we employ C LSTM layers. The task in next phase is to modelling the interdependencies between these channels. Then, global average pooling layers are utilized to concentrate feature maps of each channel, generated by memory establishing process.

Next, we construct several full-connected hidden layers to generate different weights for every channel by exploiting the information aggregated in previous steps. In general, the number of channels or the species of interdependent cryptocurrencies is rare. In order to increase the dimension, the number of neurons in the first full-connected layer is much larger than C. To enable full-connected layers to make non-linear transformation, we utilize the ReLU as the activation function in each hidden layer. We also fix neurons as C in the last dense layer, to return to the dimension of the original channels.

Specially, we employ a sigmoid activation function after the last full-connected layer to generate the weights vector. In particular, the sum of C elements in the weights vector is 1. The generated weights vector is next multiplied with each channel by corresponding element.

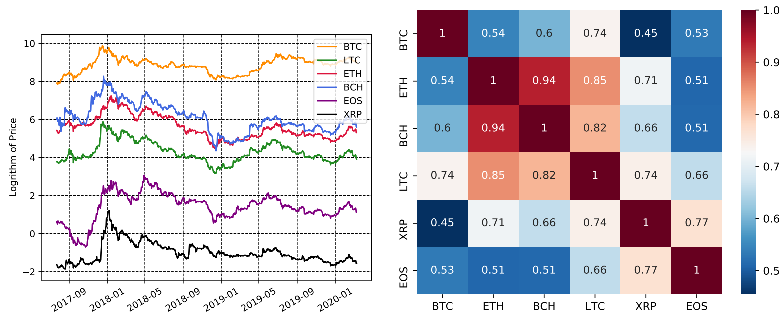
2.3 Regularization Auxiliary

After the weights and memory establishing process for multi sequences, we build two CNN layers to extract spatial features of multi weighted memory channels. In this module, several methods of regularization are employed to tackle the problem of poor generalization and overfitting.

- **Dropout method:** Dropout is a technique for preventing overfitting. It reduces training time by randomly dropping neurons along with their connections from the networks with a specified probability [13]. We set a Dropout rate as 0.55, representing that 55% neurons are dropped.
- **Kernel regularizer:** In the model optimization process, the kernel regularizer allows to apply penalties on weight matrices W. Meanwhile, W also represents the convolutional kernels in convolutional layers. In the proposed model, we use the ℓ_2 regularization in the kernel regularizer.
- **Bias regularizer:** Similarly, we use ℓ_1 regularization and employ a bias regularizer to apply penalties on bias terms b. We also conduct the combinations of ℓ_1 regularization and ℓ_2 regularization in both kernel regularizer and bias regularizer, showing the best performance.

The choice of different norm for kernel and bias regularizer is inspired by the Regularization Self-Attention Regression model in [30]. The loss function with regularization is computed as:

$$\mathbf{L} = \mathbf{L}_0 + \lambda(\sum_{i=1}^{N}\sum_{j=1}^{M} W_{ij}^2 + \sum_{i=1}^{N} |b_i|), \tag{10}$$



(a) Logarithm of prices. (b) Correlation matrix.

Fig. 2. The similarity and correlation between price of popular cryptocurrencies.

where $\mathbf{L}$ and $\mathbf{L}_0$ represent the loss before and after using the regularizers, respectively. And the elements in convolutional kernels are denoted by W_{ij} while b_i denotes the bias term and λ refers to the degree of punishment. In the proposed model, λ is taken as 0.01 and the loss function is computed according to the following equation:

$$\mathbf{L}_0 = \frac{1}{N}\sum_{i-1}^{N}(\hat{y}_i - y_i)^2, \tag{11}$$

where $\hat{y}_i$ represents the forecast value and y_i denotes the actual value.

3 Data Specification

According to the cryptocurrencies price released by CoinMarketCap at the website https://coinmarketcap.com/, we investigate and compare six popular and valuable cryptocurrencies including Bitcoin (BTC), Bitcoin Cash (BCH), Litecoin (LTC), Ethereum (ETH), XRP and EOS. Considering different prices at different time, we pick out the daily closing prices from July 23, 2017 to March 9, 2020. In Fig. 2(a), we take logarithm of prices to unify the scale. It can be noticed that the price curves show high similarity of fluctuation. To analyze the degree of similarity, we employ Pearson Correlation Coefficient(PCC) on paired price time series. The PCC value between two variables is computed as following,

$$\mathrm{PCC} = \frac{\mathrm{E}[(\mathrm{X}-\mu_\mathrm{X})(\mathrm{Y}-\mu_\mathrm{Y})]}{\sigma_\mathrm{X}\sigma_\mathrm{Y}}, \tag{12}$$

where X and Y are different variables and μ is the mean. In addition, σ is the standard deviation and E denotes the expectation. The numerator in Eq. (12) is essentially the covariance between X and Y. It is shown in Fig. 2(b) that PCC values between each pair in the prices of ETH, BCH and LTC are greater than 0.8. This result indicates the high correlations among them. In addition, the lowest PCC values of 0.45 and 0.51 are from the rows (or columns) of XRP and EOS separately. In Fig. 2(a), it shows the prices of XRP and EOS are

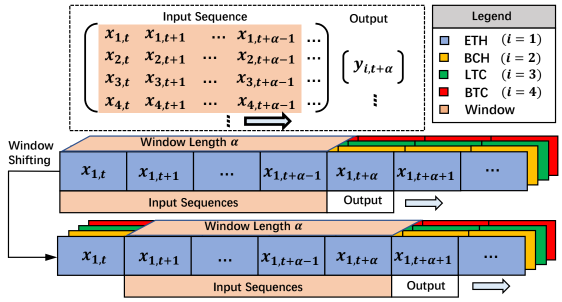


Fig. 3. Data preprocessing.

orders of magnitude lower than other four cryptocurrencies. To the input of WMCR model, we choose the cyptocurrencies in similar price levels and have high PCC values with each other. This choice is conducive to construct more related channels in Fig. 1. We conduct the experiments mainly based on the prices of BTC, BCH, ETH and LTC. The preprocessing can be divided into two steps. First, we have employed the `StandardScaler` method from scikit-learn [24]. It standardizes features by subtracting the mean and scaling to unit variance.

Next, a scrollable window is applied to move on the dataset. The number of time steps of each sample is regarded as the window length α. We use the previous data of the prices in α days to predict the price in day $\alpha+1$. Fig. 2(a) shows the scrollable window concretely. As discussed in previous analysis of correlations, price of four kinds of typical cryptocurrencies are employed as the input of the proposed model. Specifically, these time series are combined and reshaped to a tensor with the dimensions arranged as batch size, time steps, features and channels. It is worth mentioning that the multi-channel price data is only applicable to our model while the dataset of baseline models merely includes single type of cryptocurrency. In this paper, we make price prediction primarily on ETH and LTC.

4 Experiment

4.1 Training Setup

We use a 7-day window length of the the data in a batch and our goal is to predict the closing price in the 8-th trading day. To unify the standard of measurement, we use the first 80% of the data as the training set while the last 20% as the test set. Moreover, we set a random seed before the training process to make the experiment results reproducible. According to the experiments, both training loss and validations loss converge after approximately 100 epochs of training. Meanwhile, a small-batch training is slow and hard to converge while a large-batch training may converge to sharp minimizers and result in poor generalization [17]. Considering the efficiency and feasibility, the training epochs and batch-size are fixed to 100 and 80 empirically, respectively. In addition, we use the Adam optimizer [18] in the training process and the initial learning rate is 0.01.

4.2 Evaluation Metrics

To compare the performance of distinct models, we employ four commonly-used metrics to evaluate the prediction accuracy. There are the root mean square error (RMSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the R-squared. RMSE represents the square root of the average of squared residuals and the effect of each error is proportional to the squared error. Consequently, RMSE is susceptible to outliers. MAE measures the mean absolute errors between predicted prices and actual prices. MAPE usually expresses the accuracy as a ratio. It considers the ratio of the error to the actual value. R-squared (R^2) is a statistical measure of the regression model prediction. It indicates the extent to which the regression explains the change of dependent variable [3]. The R-squared measure is close to 1, meaning the better for the regression fittings.

4.3 Baseline Models

We employ six baselines to compare the performance with our proposed model. These baselines are the representative and prevailing methods from the fields of machine learning, deep learning and time series analysis.

Autoregressive Integrated Moving Average (ARIMA) is an advanced method fit to time series data analysis and prediction [6]. It is widely applied to non-stationary financial data . Support Vector Regression (SVR) is an important application of SVM (Support Vector Machine). For non-linear separable datasets, SVR uses kernel functions to map data to high-dimensional space, and finds a hyperplane closest to the data [10,25]. Multilayer Perceptron (MLP) have achieved the state-of-art performance in various computer vision tasks. We construct an MLP with 2 dense layers and employ ReLU and sigmoid as the activation. The CNN structure is a type of feed-forward network with deep structure and convolution computation. It mainly includes convolutional layers, pooling layers and fully-connected layers. In this baseline, we apply two CNN layers. LSTM neural network is specially developed to tackle the long-term dependence of general RNN. It is well-adapted for predicting time series data and can control the transmission state through three gates. Then it can remember the important features and forget the unimportant information. Two layers of LSTM are employed in this experiment. To capture the pivotal temporal information and to extract features simultaneously, we implement a mixed-structure baseline of LSTM+CNN. Gated Recurrent Unit (GRU) is a frequently used type of Gated Recurrent Neural Network [8]. It achieves a close performance to LSTM but is computationally much simpler. We also construct a baseline of GRU+CNN.

Table 1. Influence of different parameters

Loss	Window length α			Dense Neurons β			CNN Layers γ		
	7	**11**	**15**	**16**	**32**	**64**	**1**	**2**	**3**
Training loss	0.1154	0.0882	0.0905	0.1162	0.1070	0.1197	0.1886	0.1031	0.1595
Validation loss	0.0419	0.0706	0.1182	0.0427	0.0160	0.0316	0.1792	0.0279	0.0594

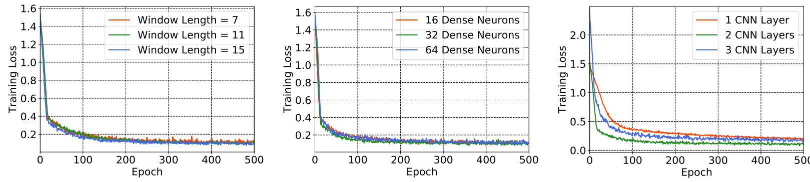
4.4 Parameter Study

To investigate the impact of multifarious parameters of WMCR model, we divide the previous training set in section 4.1 into a new training set (the first 70% data) and a validation set (the remaining part of 30%). Moreover, both the training loss and validation loss are computed by Eq. (11). In addition, since predicting different cryptocurrencies influences on the performance slightly, to eliminate this effect, we all employ the dataset of ETH. Table 1 shows the training loss and validations loss after 500 epochs of training. The effect of three critical parameters are discussed as follow:

Effect of window length α. We perform the effect comparison between different window length α. Specially, the value of β in the multi-channel weighting block is fixed to 32 and γ is fixed to 2. Then, we vary α from 7, 11 and 15. Fig.4(a) shows the fluctuation of the training loss is not obvious with the change of α. In Fig.5(a), while α is 7, the validation loss converges more rapidly. The reason is that our model shows the better generalization ability since a shorter sequence requires less memory cells in LSTM layers. Moreover, while we increase α to 11 or 15, the performance on validation loss drops sharply. This result indicates that establishing memory for a long sequence causes over-fitting. Therefore, we fix α to be 7 in the proposed model.

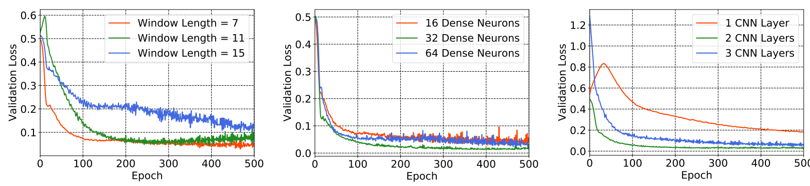
Effect of number of dense neurons β. We next investigate the effect of number of neurons β in each hidden layer in multi-channel weighting block. We vary β from 16, 32 to 64 while the parameters of α and γ are fixed to 7 and 2 separately. Fig. 4(b), the training loss with different β show similarity. It indicates that 16 neurons in each dense layer are capable of fitting the training set. However, Fig.5(b) shows that the model with 32 dense neurons in multi-channel weighting block achieves noticeable lower validation loss and faster speed of convergence in the first 100 iterations. Combined with the results in Table 1, fixing β to 32 shows the superior performance.

Effect of number of CNN layers γ. To investigate the effect of number of CNN layers γ, we vary γ from 1, 2 and 3. Meanwhile α is fixed to 7 and β is fixed to 4. As shown in Fig. 4(c) and Fig. 5(c), the model with a single CNN layer generates obvious higher loss than other two groups with more CNN layers. In addition, the validation loss with 1 CNN layer increases notably at the beginning. This increment implies that a single CNN layer is insufficient to extract features and demands more iterations to acquire generalization ability. Meanwhile, as γ increases form 2 to 3, the curves of both training loss and validation loss exhibit



(a) Effect of α. (b) Effect of β. (c) Effect of γ.

Fig. 4. The impact of the significant parameters on training loss.



(a) Effect of α. (b) Effect of β. (c) Effect of γ.

Fig. 5. The impact of the significant parameters on validation loss.

the increased tendency. It implies that a deep convolutional neural network is not suit to our WMCR model.

4.5 Performance Comparison

We also perform the performance comparison between the WMCR model and baseline models. In particular, we set the parameter of α, β and γ to be 7, 32 and 2, respectively. Other parameters for all models in Table 2 retain consistency. For instance, we set the training ratio to be 80% and the number of training epochs is 100. Moreover, we make the prediction and regression on ETH and BTC datasets.

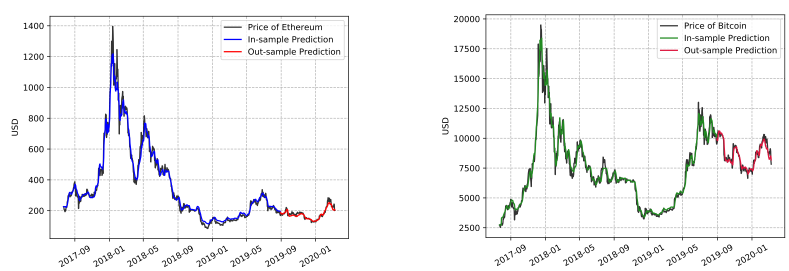
Table 2. Performance comparison

Models	Ethereum				Litecoin			
	RMSE	**MAE**	**MAPE**	**R-squared**	**RMSE**	**MAE**	**MAPE**	**R-squared**
SVR	1.24E+02	1.15E+02	7.09E−01	−1.11E+01	1.92E+01	1.60E+01	3.21E−01	−2.14E+01
ARIMA	1.56E+02	1.48E+02	8.45E−01	−1.80E+01	7.50E+01	7.33E+01	1.30E+00	−4.69E+01
MLP	2.45E+01	1.89E+01	1.15E−01	5.30E−01	5.71E+00	4.76E+00	8.21E−02	7.22E−01
LSTM	1.77E+01	1.40E+01	8.26E−02	7.54E−01	4.70E+00	3.63E+00	6.32E−02	8.21E−01
CNN	1.98E+01	1.72E+01	1.06E−01	6.93E−01	4.69E+00	4.00E+00	7.60E−02	8.13E−01
LSTM+CNN	2.09E+01	1.68E+01	9.81E−02	6.57E−01	6.37E+00	4.84E+00	8.87E−02	6.54E−01
GRU+LSTM	2.13E+01	1.75E+01	1.01E−01	6.42E−01	6.83E+00	5.05E+00	9.13E−02	6.03E−01
WMCR	**1.23E+01**	**9.69E+00**	**5.62E−02**	**9.03E−01**	**3.80E+00**	**2.90E+00**	**5.06E−02**	**8.77E−01**

First, we evaluate three typical machine learning models including SVR, ARIMA, MLP. As shown in Table 2, the evaluation metrics including RMSE, MAE, MAPE and R-squared have been computed for each model on ETH and LTC price datasets. For instance, the SVR achieves 1.24E+02, 1.15E+02, 7.09E+01, −1.11E+01 in RMSE, MAE, MAPE and R-squared measure, respectively on test set. Among these three machine learning models, MLP achieves better performance in both two datasets.

Second, we compare the performance of four deep learning models including LSTM, CNN, LSTM+CNN, GRU+CNN. The model of LSTM which consists of two LSTM layers shows superior performance. In particular, most of deep learning models have lower values of RMSE, MAE, MAPE and higher R-squared compared with conventional machine learning models. The reason of this improvement may lie in the distinguished ability of deep learning model in extracting features from highly stochastic data.

We then perform the evaluation of WMCR model. Table 2 shows the results of performance. Compared with other baseline models, the WMCR model achieves the lowest level of RMSE, MAE, MAPE and the highest value of R-squared. This result indicates our proposed model achieves better fitting and predicts more accurately on the price of ETH and BTC than the baselines. In Fig. 6, we show the real price curves and the predicted curves generated by our model on ETH and BTC datasets. The blue and green curves (in-sample prediction) represent the closing price fitted by our model on training set while the red curve (out-of-sample prediction) represents the predicted price on test set. Both the in-sample predictions and the out-sample predictions are close to the real prices.



(a) Prediction of WMCR on ETH. (b) Prediction of WMCR on BTC.

Fig. 6. Prediction on prices of different cryptocurrencies

5 Conclusion

In this paper, we put forth a Weighted Memory Channels Regression (WMCR) model to predict the daily close price of cryptocurrencies such as ETH and BTC by exploiting the price of four closely related cryptocurrencies. We use the WMCR model to establish the structure memory and to recalibrate the importance for different channels before extracting the temporal and spatial features of different price sequences. We also test the WMCR model on the dataset of historical prices of four heavyweight cryptocurrences (from July 23, 2017 to March 9, 2020). Based on the experiments, we show that our proposed model performs better than other baselines including prevailing econometric, machine learning and deep learning approaches. To improve this study, we are going to examine the use of more market information of cryptocurrencies in WMCR model such as market capitalization, volume and open price. We will also investigate how the weights vector generated by Multi-Channel weighting block affects the performance of WMCR model. Moreover, we intend to explore more reasonable numbers and types of channels in WMCR model to improve the prediction accuracy.

Acknowledgments

The work described in this paper was partially supported by Macao Science and Technology Development Fund under Macao Funding Scheme for Key R & D Projects (0025/2019/AKP).

References

1. Abraham, J., Higdon, D., Nelson, J., Ibarra, J.: Cryptocurrency price prediction using tweet volumes and sentiment analysis. SMU Data Science Review **1**(3), 1 (2018)
2. Alameer, Z., Fathalla, A., Li, K., Ye, H., Jianhua, Z.: Multistep-ahead forecasting of coal prices using a hybrid deep learning model. Resources Policy **65**, 101588 (2020)
3. Anderson-Sprecher, R.: Model comparisons and r 2. The American Statistician **48**(2), 113–117 (1994)
4. Aslanidis, N., Bariviera, A.F., Martínez-Ibañez, O.: An analysis of cryptocurrencies conditional cross correlations. Finance Research Letters **31**, 130–137 (2019)
5. Bakar, N.A., Rosbi, S.: Autoregressive integrated moving average (arima) model for forecasting cryptocurrency exchange rate in high volatility environment: A new insight of bitcoin transaction. International Journal of Advanced Engineering Research and Science **4**(11) (2017)
6. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)
7. Chollet, F., et al.: Keras. https://keras.io (2015)
8. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
9. Ciaian, P., Rajcaniova, M., et al.: Virtual relationships: Short-and long-run evidence from bitcoin and altcoin markets. Journal of International Financial Markets, Institutions and Money **52**, 173–195 (2018)
10. Drucker, H., Burges, C.J., Kaufman, L., Smola, A.J., Vapnik, V.: Support vector regression machines. In: Advances in neural information processing systems. pp. 155–161 (1997)
11. Dyhrberg, A.H.: Bitcoin, gold and the dollar–a garch volatility analysis. Finance Research Letters **16**, 85–92 (2016)
12. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. pp. 6645–6649. IEEE (2013)

13. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
15. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
16. Jang, H., Lee, J.: An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. Ieee Access **6**, 5427–5437 (2017)
17. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836 (2016)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Kristoufek, L.: What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. PloS one **10**(4) (2015)
20. Long, W., Lu, Z., Cui, L.: Deep learning-based feature engineering for stock price movement prediction. Knowledge-Based Systems **164**, 163–173 (2019)
21. McNally, S., Roche, J., Caton, S.: Predicting the price of bitcoin using machine learning. In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP). pp. 339–343. IEEE (2018)
22. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
23. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Tech. rep., Manubot (2019)
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
25. Peng, Y., Albuquerque, P.H.M., de Sá, J.M.C., Padula, A.J.A., Montenegro, M.R.: The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression. Expert Systems with Applications **97**, 177–192 (2018)
26. Sifat, I.M., Mohamad, A., Shariff, M.S.B.M.: Lead-lag relationship between bitcoin and ethereum: Evidence from hourly and daily data. Research in International Business and Finance **50**, 306–321 (2019)
27. Sovbetov, Y.: Factors influencing cryptocurrency prices: Evidence from bitcoin, ethereum, dash, litcoin, and monero. Journal of Economics and Financial Analysis **2**(2), 1–27 (2018)
28. Walther, T., Klein, T., Bouri, E.: Exogenous drivers of bitcoin and cryptocurrency volatility–a mixed data sampling approach to forecasting. Journal of International Financial Markets, Institutions and Money **63**, 101133 (2019)
29. Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H.: Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services **14**(4), 352 – 375 (2018). https://doi.org/10.1504/IJWGS.2018.095647
30. Zhou, J., He, Z., Song, Y.N., Wang, H., Yang, X., Lian, W., Dai, H.N.: Precious metal price prediction based on deep regularization self-attention regression. IEEE Access (2019)