CS 334 Machine Learning

Cryptocurrency Price Forecasting at Minute Resolution Based on Improved LSTM Models

Group: Yu-jan Ting, Kaifu Xiao

December 15, 2021

Abstract

The long short-term memory (LSTM) network is a state-of-the-art model for time series forecasting like stock or cryptocurrency markets. However, less study has been made on the minute-level prediction of cryptocurrency markets. We proposed three different models, LSTM, stacked LSTM, and CNN-LSTM, to forecast the closing price of cryptocurrency based on a small amount of previous sequential data. We trained our models on minute bitcoin price from 2019/4/1 to 2019/5/2. The proposed CNN-LSTM outperformed other models under different evaluation indicators such as mean squared error (MSE), root means squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). We also explained why CNN-LSTM was better than others in minute-level prediction. The study reveals the possibility of automatic transaction robots in the Bitcoin market.

1 Introduction

Bitcoin (BTC), a decentralized digital currency, is one of the most famous examples of cryptocurrencies. Based on the history of the bitcoin diagram (Figure 1), we can find that there has been a steep increase in the price of bitcoin starting from 2018. The fluctuation attracted people to participate in this digital currency market. On the other hand, the cryptocurrency market's instability and high uncertainty interest us in predicting the price at a minute level. If the minute-level prediction is possible, this proposes introducing automatic transaction robots in the Bitcoin market.

LSTM model proposed by Sepp Hochreiter and Jürgen Schmidhuber is the state-of-the-art model for time series prediction in neural networks. However, even though the LSTM model performs on daily level price prediction, it seems not promising as it acts on the minute level price prediction. Moreover, we found that the original LSTM model requires many trial-and-error processes and parameters tuning during the experiments. The bad performance was due to the high fluctuation of minute-level data. Therefore, we proposed to use the stacked LSTM model, which stacks multiple LSTM layers and dropout layers to prevent oscillating and overfitting. Inspiring by the idea of using model stacking of CNN-LSTM [6], we also propose the CNN-LSTM model to hope it will capture the critical characteristics from the highly fluctuating data. Our approach was to convert the raw dataset to a ready-to-train dataset. After using gird search to tune the best model parameters and hyperparameters, we compared the performance of different models by comprehensive evaluation indicators.

2 Literature Review

Over the past decade, Bitcoin has been studied on multiple levels, including price formation, volatility, system dynamics, and economic value. A study by D. G. Baur et.al. helped clarify the fundamental and

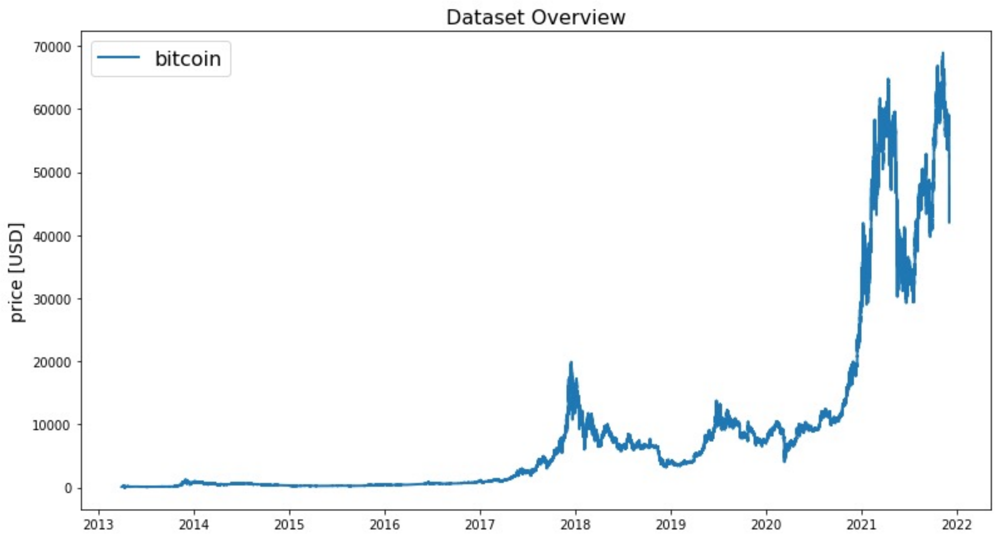

Figure 1: The price of Bitcoin to USD from 2013 to 2021

speculative value of Bitcoin from an economic perspective. It gave ample support for the claim that Bitcoin is an financial asset that is readily susceptible to dynamic market activities. [1] Due to the resemblance of cryptocurrency to other investment properties such as stocks [1], the price of cryptocurrencies is likely to be predicted using similar forecasting methods for the stock market. Traditional analysis methods and machine learning methods are two types of forecasting methods for the noisy stock market.[8] However, traditional analysis involves econometric methods or equations, both of which are not suitable for analyzing series data in a highly dynamic and complicated market system.[6] Therefore, neural networks have come into the play when a powerful and dynamic forecasting method is required. Neural networks are capable of extracting patterns from a large volume of data without requiring any knowledge beforehand.

In 2018, [3] compares the ARIMA Time Series Model and LSTM Deep Learning Algorithm to determine the future price of Bitcoin. The daily prices were estimated with the obtained models, with a result of MAPE 11.86% with ARIMA and MAPE 1.40% with LSTM. Combining other accuracy test results, the research supported that LSTM is a better model for time series price prediction. However, the price forecasting method for daily price may perform differently from price data in minute-level, since the price fluctuates rapidly and may lead to serious over-fitting. In response to this issue, [9] has proposed a solution that reduces over-fitting by randomly omitting half of the feature detectors on each training.

Different from the monotonous neural network above, the studies by [2] and [4] have proposed mixture models of LSTM. The mixture models proposed in [4] have better-defined data flows and architecture to capture time-varying effect in data compared to recurrent neural network based methods. In this paper, we aim to compare the performance of mixture models comparing to other variants of LSTM (CNN-LSTM, LSTM, stacked-LSTM) and find a possible data preprocessing method to enhance the performance of mixture or vanilla models when dealing with minute-level time series data.

3 Models

3.1 LSTM model

LSTM was first introduced by Hochreiter & Schmidhuber (1997), which is explicitly designed to avoid the long-term dependency problem based on RNN. RNNs have feedback from the loops in the recurrent layer, which will save the parameters in the memory over time. However, it does not perform well in learning long-term temporal dependencies because of a limited range of contextual information. Also, RNN faces the problem of vanishing gradient. The Long Short-Term Memory (LSTM) network is a type of recurrent neural network capable of learning order dependence in sequence prediction problems, which is designed to address the vanishing gradient problem. LSTM units have a memory cell that can save the parameters for a long period of time by manipulating 4 gates. The forget gate, input gate, cell gate, output gate. The details of each LSTM cell is illustrated in Figure 2.

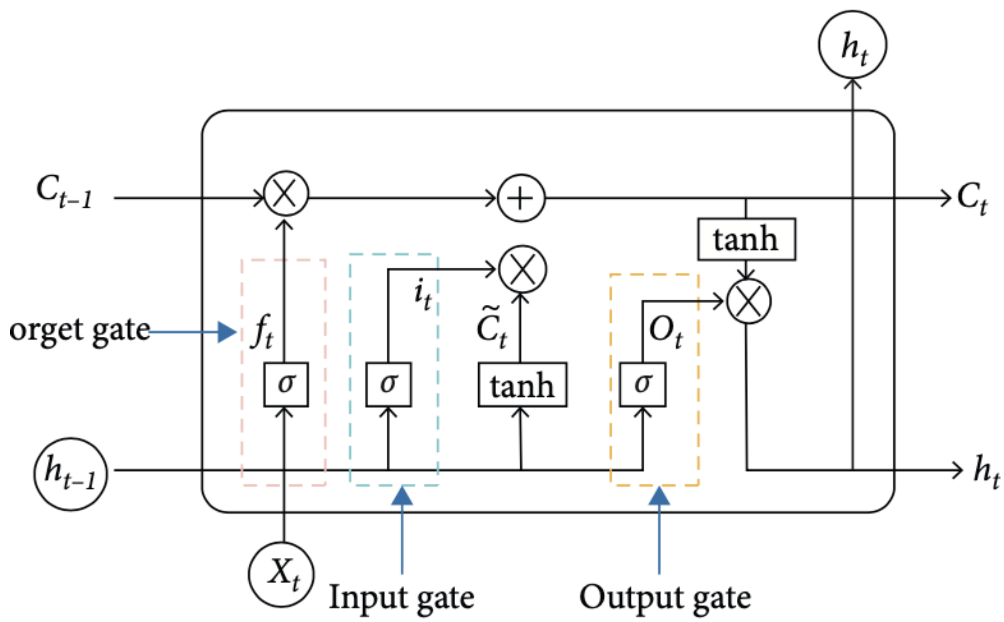

Figure 2: The illustration of a LSTM cell, cited from Lu, W., Li, J., Li, Y., Sun [6]

3.2 Stacked LSTM model

Since training the bitcoin prediction model is to learn a hierarchical representation of the time-series data, we want our model's hidden layer deeper to improve the performance. By vertically stacked the LSTM Architecture, each LSTM layer outputs a sequence of vectors which will be used as an input to a subsequent LSTM layer. This approach potentially allows the hidden state at each level to operate at different timescale. Between LSTM layers, we also include dropout layers to prevent overfitting.

3.3 CNN-LSTM model

The CNN-LSTM [6] was designed for the sequence prediction problem, like the image description, video description, and textual description. The CNN-LSTM includes applying Convolutional Neural layers for feature extraction on input data and then passing the results to the LSTM to perform sequence perdition. Using the CNN network to get the most important features and apply these to improve the performance in the LSTM model. The CNN-LSTM contains the input layer, conv1d, maxpool1d, LSTM hidden layer, and dense layer.

4 Experiment

4.1 Data

The data utilized in this research concern historical data from 1 April 2019 to 2 March 2019 of BTC in USD at the minute level, which constitute the cryptocurrencies with the highest market capitalization. The data for this research were collected from www.kaggle.com. For evaluation purposes, the dataset was divided into the training set, validation set, and testing set. The training data set contains the minute level data of BTC from 1 April 2019 to 25 April 2019 (26394 data points), the validation data set contains the minute level data from 26 April 2019 to 29 April 2019 (6440 data points), and the test data set contains the minute level data from 30 April 2019 to 2 March 2019 (6440 data points).

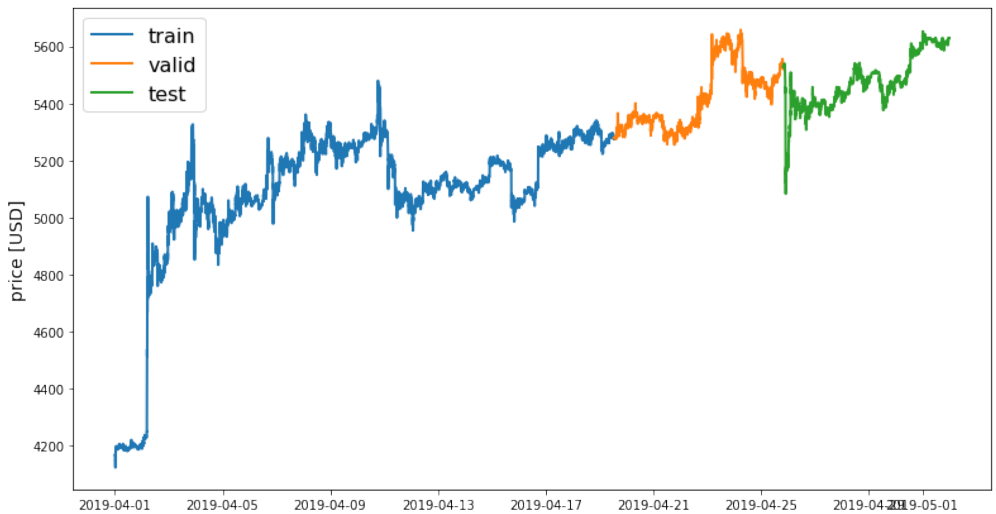

Figure 3: The overview of the train, valid, test data

The results show that the BTC value at the minute level fluctuated significantly during the time period of data collection (Figure 3). These fluctuations were most prominent from 26 April to 28 April 2019. An observation is that the Bitcoin market is in considerable volatility during this period and deviations from the regular behavior from the normal stock market (Figure 3). This period of data could represent the regularity of uncertainty of the Bitcoin market, which made traditional time series prediction models less likely to catch the essence of the data.

To fully understand the data, we include the results of the t-statistics and the associated p-values of the augmented Dickey-Fuller (ADF) test performed on our data set (Table 1). We selected the statistical significance at a 5% critical level (see Table 1). The interpretation of the Table 1 shows that BTC time-series implies that these series are non-stationary.

Table 1: Augmented Dickey-Fuller (ADF) unit root test of the Bitcoin time-series

Time-Series	t-Statistic	p-Value
Train	-4.00218	0.011820
Valid	-2.05183	0.213406
Test	-2.05202	0.228939

4.2 Data Processing

To prepare our data ready for training, we mainly did two following things to our original data set. One is the normalization of our data set, which is to change the values of numeric columns in the dataset to a common scale without distorting differences in the ranges of values. The other one is the use of the sliding window algorithm, which makes may be suitable for the prediction of highly nonlinear stock data.

For the normalization, we use zero-based normalizing processing. The zero-based normalization is to reflect changes with respect to the first entry. The method is described as given a vector V with the size of n, and we constructed the new vector V' by the following equation

$$V' = V/V[0] - 1 \tag{1}$$

After constructing the V', we recorded the minimum value V_{min} and maximum value V_{max} among all transformed V' vectors. Then we construct the final result V'' through

$$V'' = (V' - V_{min})/(V_{max} - V_{min}) \tag{2}$$

Following this processing, we preserve the ordering of data in their original data set and make the data set easier for the neural network to train.

Besides normalization, we also introduce the sliding window algorithm, which is a common technique for manipulating time series data. The sliding window algorithm for time series data works by selecting a specific point t in time and the fixed width of window length win_len of the subsequent data. Then the corresponding y label was created from data at the position $t + win\_len + 1$ of the original data set (see figure x). The sliding window maximizes the usage of time series data while maintaining the inherent sequence of the data set. The window length we choose for our project is $win\_len = 5$. The batch size we choose is $batch = 5$. Thus the resulting input which is ready for training has the dimension of $(x, 5, 5)$, where the first omitted value x is the size of the data set, first 5 is the batch size $batch$ and the second 5 is our win_len

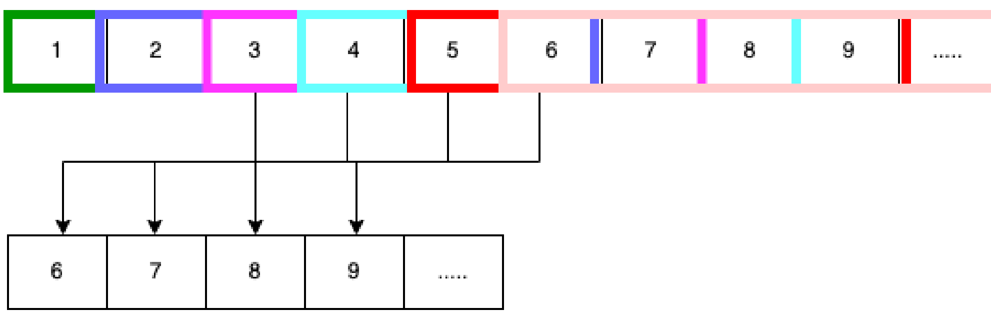

Figure 4: The illustration of sliding window for time series data

4.3 Parameters

We tune our hyperparameters through the grid search technique. The parameters we choose for each model are summarized in Table 2 for reproducibility.

Table 2: Hyperparameters and parameters choices for each models

Models	window_len	hidden_dim	learning_rate	num_epoch	drop_out_prob [a]
LSTM(Baseline)	5	5	1e-5	5	-
Stacked LSTM	5	5	1e-5	5	0.2
CNN-LSTM	5	5	5e-6	4	-

[a] Note that the dropout probability was set to same acorss all dropout layers

4.4 Method

To evaluate the performance, we uses the mean squared error (MSR), root mean square error (RMSE), mean absolute error (MAE), and mean absolute error (MAE), and mean absolute percent error (MAPE) as our perfomance indicators. The equations are:

$$MSE = \frac{1}{T}\sum_{t=1}^{T}(P_t - P_t')^2 \tag{3}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(P_t - P_t')^2} \tag{4}$$

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|(P_t - P_t')| \tag{5}$$

$$MAPE = \frac{100}{T}\sum_{t=1}^{T}|\frac{P_t - P_t'}{P_t}| \tag{6}$$

where P_t is the acutal price of the Bitcoin, P_t' is the predicted price of the Bitcoin.

4.5 Results

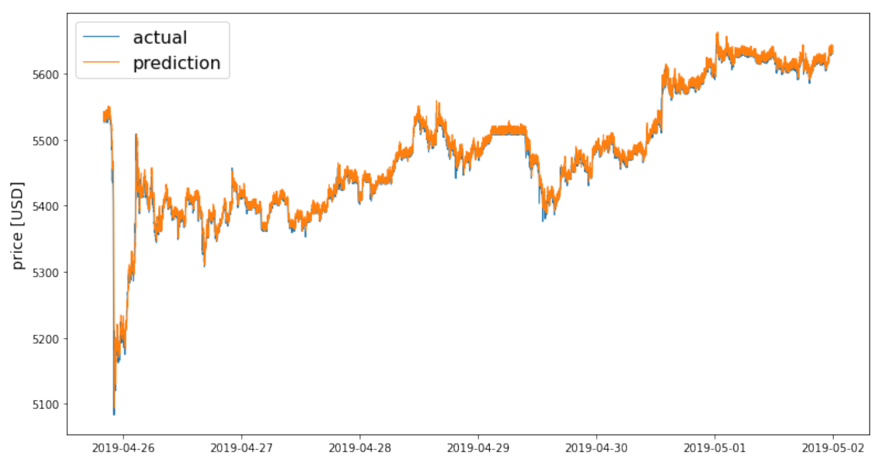

Figure 5: Comparison of the predicted value and the real value for LSTM(baseline)

Table 3: Precision results of the models under different estimation indicators.

	Precision			
Models	MSE[a]	RMSE[b]	MAE[c]	MAPE[d]
LSTM(Baseline)	27.89	52.82	50.76	1.56
Stacked LSTM	25.86	50.85	48.79	1.60
CNN-LSTM	7.041	26.53	22.41	8.07

[a] MSE Unit: 10^{-6} [b] RMSE Unit: 10^{-4} [c] MAE Unit: 10^{-4} [d] MAPE Unit: Percentage %
Note: these values were obtained by average the outcome running ten times.

We used the processed training data set to train the LSTM, Stacked LSTM, and CNN-LSTM models, respectively. After we finish training, we use the trained models to predict the test data set, and the real

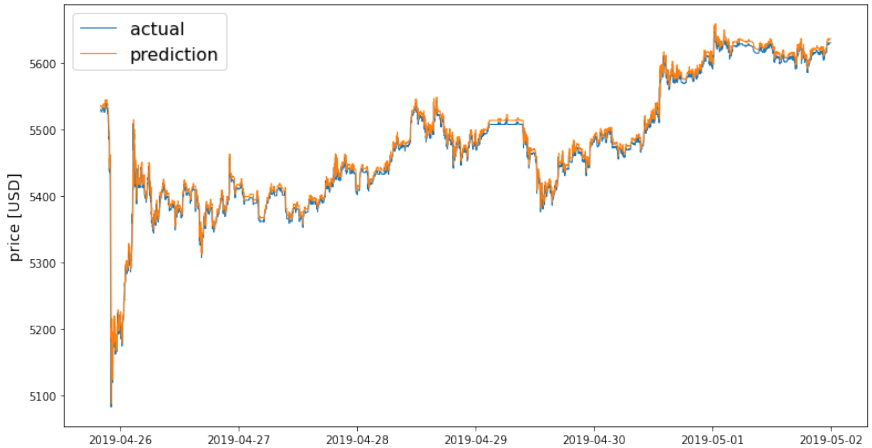

Figure 6: Comparison of the predicted value and the real value for Stacked LSTM

Figure 7: Comparison of the predicted value and the real value (all test data set) for CNN-LSTM

value of the test data set is compared. The comparison of the predicted value and the real value was shown in Figures 5-8, which are LSTM, stacked LSTM, and CNN-LSTM, respectively. We also included figures 8-10 which scale up by including only 500 data points to help us visually estimate the performance of the models. After several experiments, we found the appropriate hyperparameters and model parameters for each model on the validation data set. Then we run each model several times and get the average of the different estimation indicators. The results are included in Table 3.

4.6 Analysis

From table 2, the MSE, RMSE, and MAE of CNN-LSTM are the smallest among the three models. By comparing stacked LSTM with LSTM, we see that the MAE of stacked LSTM decreased from the baseline LSTM from 27.89 to 25.86, and the MAE decreased from 52.83 to 50.85. The decrease is matched with our expectation since we have added dropout layers for each LSTM layer, and the dropout layers will prevent the model from overfitting, which will make the data less likely to oscillate. The result shows that the performance of CNN-LSTM is the best among the three models. It has the lowest MSE, RMSE, and MAE, which are 7.041, 26.53, 22.41, respectively. The accuracy was significantly increased was due to the newly added CNN layers. The CNN layer works as we expected that it would select categorize the important feature from data. This will make the model more reluctant to a small change in the original data set. In figure 10, we can see that the predicted curve and the actual curve have almost exactly shape. The model can well predict the closing price of the next minute, which provides the possibilities for the automatic transaction robot.

5 Conclusion

To predict the closing price of cryptocurrency of next minute based on the previous closing price a few minutes ago, we proposed three different models, LSTM, stacked-LSTM, and CNN-LSTM, to forecast. Among the three models, we found the CNN-LSTM was best suitable for forecasting the closing price of the Bitcoin market on the minute level. We then tested the performance of our model on the Bitcoin market. The test results show that the forecast MSE and MAE score of the CNN-LSTM model was statistically significantly better than the original LSTM, which is believed to be the state-of-art model for prediction on time series data.

Our study mainly contributed to the following aspects. First, we offer more perspective views on the minute-level cryptocurrency price prediction. The minute-level cryptocurrency data set was different from daily-level data since the data fluctuated rapidly in minutes. Additionally, due to the nature of cryptocurrency, such an extremely sudden price change happened fast and common, which makes the models hard to distinguish from the normal fluctuation of price or the sudden change. Such characteristics will make many statistical testing required by the traditional time series models like ARMA fail, and even the transformation won't have satisfactory results. It also proposed the challenge to the traditional ML models like LSTM. We explained how to overcome the specificity of minute-level price prediction with different architectures of LSTM and CNN on a minute-level cryptocurrency data set. Second, we compared the performance of three different neural network models, LSTM, stacked-LSTM, and CNN-LSTM, in different estimations. We found that CNN-LSTM was more suitable for predicting cryptocurrency closing prices than LSTM in this case. The results would lead us to make further interpretations about why one model works better than another model. Third, we also add possible explanatory for the better performance of CNN-LSTM models. We believe that CNN is able to capture the key characteristics from the highly fluctuating data. We think CNN transformed the highly stable data into relatively stable data. This is the reason why it is easy for CNN-LSTM to correctly predict the closing prices in an extremely small amount of previous data. Further researches could be carried out to verify this hypothesis. They could also verify if the CNN-LSTM model also has better results in other cryptocurrency markets besides Bitcoin.

References

[1] D. G. Baur, K. J. Hong, and A. D. Lee, "Bitcoin-Currency or Asset?," presented at the 2016 Financial Institutions, Regulation Corporate Governance (FIRCG) Conference, Melbourne Business School, 2015.

[2] T. Guo and N. Antulov-Fantulin, "An experimental study of Bitcoin fluctuation using machine learning methods," in arXiv preprintarXiv:1802.04065,https://arxiv.org/abs/1802.04065, 2018.

[3] E. Ş. Karakoyun and A. O. Çibikdiken, "Comparison of ARIMA TimeSeries Model and LSTM Deep Learning Algorithm for Bitcoin Price Forecasting," in The 13th Multidisciplinary Academic Conference in Prague 2018 (The 13th MAC 2018), ed. Czech Republic, 2018, pp. 171-180.

[4] Wu, Chih-HungMa,Yu-FengLu, Chih-ChiangLu, Ruei-Shan"A New Forecasting Framework for Bitcoin Price with LSTM" in 2018 IEEE International Conference on Data Mining Workshops (ICDMW) International Journal of Computer Applications (0975 – 8887) Volume 31– No.8, October 2011

[5] Haşim Sak, Andrew Senior, Françoise Beaufays,"Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition, arxiv.org, 05 Feb, 2014. "

[6] Lu, W., Li, J., Li, Y., Sun, A., amp; Wang, J. (2020, November 24). A CNN-LSTM-based model to forecast stock prices. Complexity. Retrieved December 10, 2021, from https://www.hindawi.com/journals/complexity/2020/6622927/.

[7] Akyildirim, E., Goncu, A., Sensoy, A. (2020, April 7). Prediction of cryptocurrency returns using machine learning. Annals of Operations Research. https://link.springer.com/article/10.1007/s10479-020-03575-y.

[8] J. Li, S. Pan, L. Huang, and X. Zhu, "A machine learning based method for customer behavior prediction," Tehnicki VjesnikTechnical Gazette, vol. 26, no. 6, pp. 1670–1676, 2019.

[9] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2012). "Improving neural networks by preventing co-adaptation of feature detectors, " ArXiv, abs/1207.0580.

Appendix: Source Code

To access the source code and see our experiment results, please follow this link:

https://github.com/YujanTing/CS344-MachineLearning-FinalProject/blob/d7e960012922aab5f9007a421630c5efd68183ba/CS_334_Final_Project.ipynb

Appendix: Contribution

Yujan Ting and Kaifu Xiao shares the equal contribution of the final project. Yujan Ting was mainly in charge of performing process on data set and implementing the details of the model. Kaifu was mainly in charge of dataset collection and reviewing current related literature.