Deep Learning for Price Prediction of Cryptocurrencies

E4040.2021Spring.FLUG.proposal.gl2713.zf2261
Eric Lan gl2713, Zhongyuan Peter Fu zf2261
Columbia University

Abstract

With the advancement of blockchain technology and the increasing demand of blockchain applications, the population of investors attracted by the cryptocurrency market grows exponentially. However, forecasting the price movement of cryptocurrencies is an extremely hard task because they are backed up by different mechanisms and protocols which are different from regular stocks. By analyzing the price movement of four popular cryptocurrencies, Bitcoin (BTC), Ethereum (ETH), Monero (XMR) and Ripple (XRP), for the period 2016 - 2021, this project compares the performance of three deep learning algorithms: Long Short Term Memory (LSTM) neural network, Gated Recurrent Units (GRU) neural network and Convolutional Neural Network (CNN). The result shows that, if we only consider technical indicators, GRU outperforms other algorithms in predicting the price movement.

1. Introduction

The rise of cryptocurrencies took place in the last decade with the growth of blockchain technology. More specifically, the total market value of cryptocurrencies surpassed $2 trillion last month and the market value of Bitcoin, the most popular and expensive cryptocurrency, reaches $1.1 trillion recently, which can be ranked after the market value of Google.

During the last decade, most of the participants were individual investors, who are along with the passion and nature. Individual investors are less risk sensitive and lack analysis skills when they are compared with institutional investors. Witnessing the exponential growth and evaluating the potential risk of the cryptocurrency market, an increasing number of institutional investors have a positive perception toward this market.

Based on the survey from Fidelity Digital Assets[2], despite the concerns around price volatility and market manipulation, 36% of institutional investors currently invest in digital assets and 91% of institutional investors will make an allocation to this market within five years. Bitcoin continues to be the most popular digital asset, with more than quarter of the investors holding it.

Because institutional investors, compared with individual investors, have better analysis skills and are more risk sensitive, choosing accurate models or algorithms to forecast the price movement with better predictability (or technically less cost in cost function) and to evaluate the potential risk becomes important and inevitable. However, this is not an easy task considering the fast oscillation of coin prices and lack of indicating indexes (like S&P 500 and Dow Jones Industrial Average).

In this paper, we will use technical indicators, like closing prices and volume traded, as the input. The time span of the raw dataset starts from 2016/03/07 to 2021/04/12, and the frequency is 5 minutes. (Because we did not learn Natural Language Processing (NLP) in this semester, any indicators related with NLP, like social media and emotional indicators, will not be considered in this paper.)

We will analyze and compare the predictability of three deep learning algorithms: Convolutional Neural Network (CNN), Gated Recurrent Units (GRU) and Long Short Term Memory neural network (LSTM). Our goal is to find out the model with higher accuracy.

2. Summary of the Original Paper

2.1 Methodology of the Original Paper

The original paper compares four different algorithms: Multilayers Perceptron (MLP), Long Short Term Memory (LSTM), Multivariate Attention LSTM with Fully Convolutional Network (MALSTM-FCN) and Convolutional Neural Network (CNN).

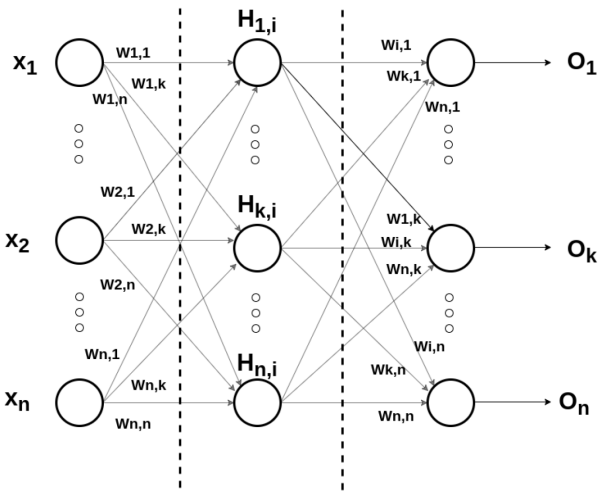

Figure 1: Multilayer Perceptron architecture

A Multilayer Perceptron (MLP) is a class of feed-forward artificial neural networks (ANNs), characterised by multiple layers of perceptrons and a typical activation function [3]. A MLP contains three layers: input layer, hidden layer(or layers) and output layer.

The Long Short-Term Memory Networks (LSTM) is a specialized type of Recurrent Neural Network (RNN) and it has been shown to learn long-term dependencies more easily than regular RNN architecture.

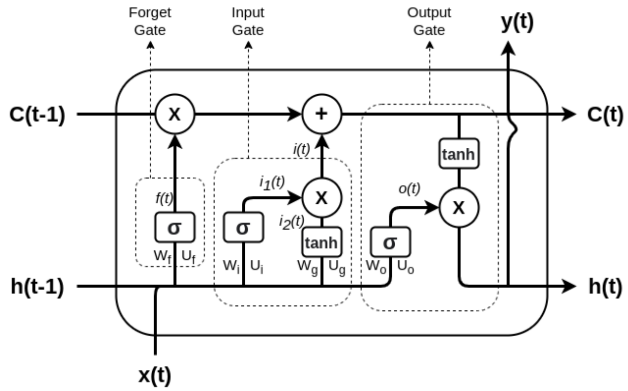

Figure 2: LSTM Cell Gate

Attention mechanism is an extension of encoder and decoder architecture. It gives the input series different priorities and pays more attention to more important inputs.

Multivariate Attention LSTM with Fully Convolutional Network (MALSTM-FCN), proposed by Karim et al. [4], shows that the model reduces the computation time of training and improves the accuracy in time series classification problems.

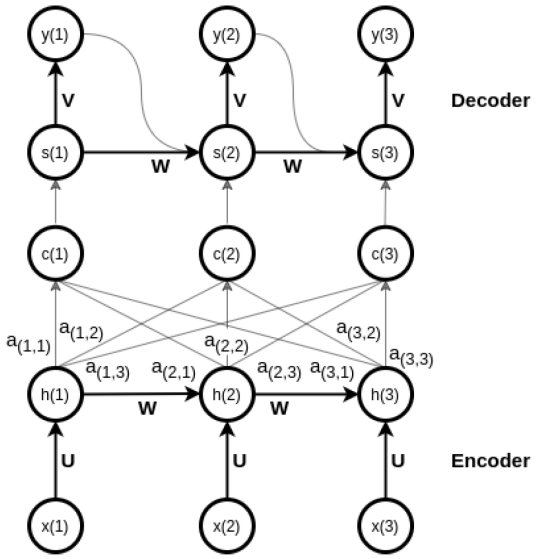

Figure 4: Attention Mechanism Neural Network

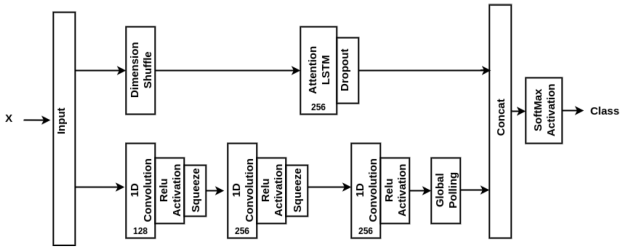

Figure 5: Attention LSTM cells to construct the MALSTM-FCN architecture [4]

Convolution Neural Networks (CNN) shares a similar architecture with traditional neural networks. The main difference is the convolutional operation, which is the matrix multiplication between input data and kernels (learnable parameters).

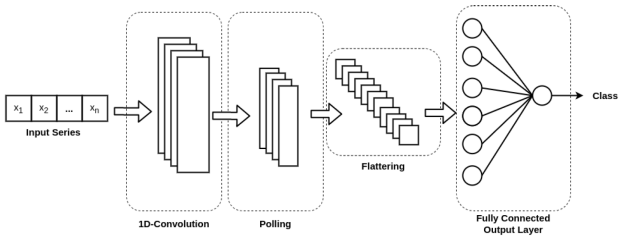

Figure 6: Convolutional Neural Network for time series forecasting

2.2 Key Results of the Original Paper

Algorithm	Parameter	Values	Accuracy ($\mu \pm \sigma$)
MLP	epochs hidden layers batch size optimizer activation neurons	250 2 256, Nadam relu 128	0.537 ± 0.029
LSTM	epochs hidden layers batch size optimizer activation neurons	250 2 256, Adamx tanh 256	0.535 ± 0.034
MALSTM-FCN	epochs hidden layers batch size optimizer activation neurons	250 - 256, Adamx - -	0.542 ± 0.034
CNN	epochs hidden layers batch size optimizer activation neurons	250 2 128, Nadam tanh 128	0.435 ± 0.024

Table 1: Accuracy of four algorithms [3]

The results are achieved from four deep learning algorithms for the hourly frequency. Table 1 shows the type of parameters and parameter values used in each model, and the final accuracy result. Among four algorithms, MALSTM-FCN achieves higher accuracy but the variance is also relatively high.

3. Methodology (of the Students' Project)

3.1. Objectives and Technical Challenges

The original paper uses four deep learning algorithms: The original paper compares four different algorithms: MLP, LSTM, MALSTM-FCN and CNN. Among them, MLP is the easiest algorithm and its performance is not the best; therefore, we will not use it in our report. MALSTM-FCN is an extension of LSTM and, moreover, we did not dive deep into the attention mechanism, so we will give up testing MALSTM-FCN. However, we will add a variation of RNN algorithm, Gated Recurrent Unit (GRU), and compare its performance with other algorithms.

The original paper trains the data from 2017/01/01 to 2021/01/01 and at hourly frequency. More specifically, each algorithm trains 35,638 hourly observations in total. However, we believe that the sample size is too small to evaluate. Instead we will analyse the price movement of cryptocurrencies at 5 minutes frequency and the time period span from 2016/03/07 to 2021/04/04. The sample size increases around fifteen-fold to more than a half million observations.

Although Bitcoin and Ethereum are two most representative cryptocurrencies and their prices are the market indicators, in order to evaluate algorithms more accurately, we are going to add two more popular cryptocurrencies, Monero (XMR) and Ripple (XRP).

3.2.Problem Formulation and Design Description

The problem formulation and architecture design start from data collection and data preparation.

Collecting reliable and clear data is important because it directly influences the final result. There are two useful databases: Kaggle and Poloniex. Poloniex is relatively preferable because it is a specialised crypto trading platform which has the most updated information and dataset.

As mentioned in the introduction, apart from Timestamp, each dataset has four other technical indicators:

- High: the highest price of traded cryptocurrency during a trading period
- Low: the lowest price of traded cryptocurrency during a trading period
- Open: the price of the first traded cryptocurrency given a certain trading period
- Close: the price of the last traded cryptocurrency given a certain trading period

	Timestamp	High	Low	Open	Close
0	1612860000	46889.919665	46706.660389	46789.372540	46803.464136
1	1612860300	47090.558200	46803.464136	46803.464136	47069.483494
2	1612860600	47126.684045	47068.219216	47071.837209	47068.219216
3	1612860900	47142.501459	46954.009700	47050.656111	47110.741744
4	1612861200	47140.165107	46968.528083	47100.000000	47007.266300

Table 2: Example of collected data (from Poloniex) including technical indicators and their values

In order to ensure the integrity of the historical dataset, the size of the training set should increase; therefore, our model should collect the data on a 5-minute basis and time span for data collection should also increase.

The validation set (or test set) should only include the cryptocurrency within a year because only the predictability of the most recent data shows the performance of algorithms. One algorithm might outperform others in 2017, but it does not indicate it is still the best model in 2020.

The next step is data preparation. We parse the source data into datas and labels and input them into the algorithms. The input size (N) is 256 and the output size (K) is 16.

Since the value of source data ranges from 0 to over 60,000, data scaling is needed to allow the neural network to understand the data easier.

The next step goes to input the parsed data into deep learning algorithms and check their performance. We will use three models: LSTM, GRU and CNN. The architecture of these algorithms will be detailedly explained in **Section 4.1 Deep Neural Network**.

The comparison among three algorithms is the final and the most critical step. We will evaluate their performance from three different aspects:

- Mean Squared Error (MSE) Loss: it evaluates the average squared error between the estimated value and actual value (Eq. 0). The best algorithm will always have relatively low MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} Y_i - \widehat{Y}_i \qquad (0)$$

- Normalized MSE (Eq. 1): considering that different cryptocurrencies have different prices and, sometimes, the difference is huge, we need a normalized variable to compare the performance across various coins.

$$Normailized\ MSE = \frac{MSE}{(Max\ Price)^2} \quad (1)$$

- Training Time: the length of training period

4. Implementation

Section 4.1. will give an overview of three deep learning algorithms including their architectures and mathematical formulations. **Section 4.2.** will introduce low level implementation in detail.

4.1. Deep Learning Network

4.1.1. Long Short Term Memory

Long Short Term Memory network (LSTM) is a specialized type of Recurrent Neural Networks (RNN). It has the same input and output as regular RNNs, but LSTM has more parameters and gate systems which control the memory and information flow.

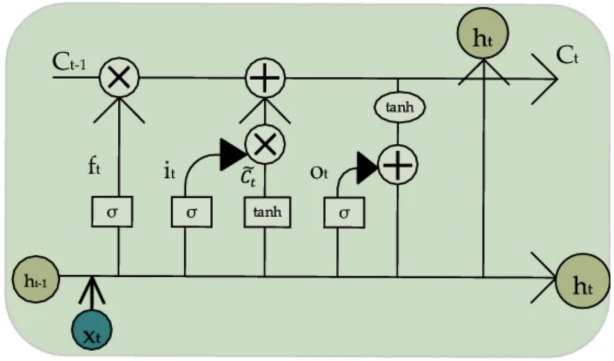

Figure 7: Example of a LSTM neural network

The main differences between LSTM and regular RNNS are cell state and three gates:

- Forget gate *f*: the sigmoid function helps the forget gate layer throw away/forget the irrelevant information (Eq. 2).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (2)$$

- Input gate *i*: it uses sigmoid functions to decide which portion of new value is relevant (Eq. 3) and uses Tanh to create new candidate values (Eq.4).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\widetilde{C_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

- Cell state *C*: it updates old cell state, C_{t-1}, into new cell state, C_t, by discarding the information decided by the forget gate layer and adding new candidate values generated by the input gate layer (Eq. 5).

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C_t} \quad (5)$$

- Output gate *o*: it uses sigmoid function to decide the relevant information (Eq. 6)and uses tanh function to output (Eq. 7).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

Because different gates have the ability of keeping relevant information/value and discarding the irrelevant ones, LSTM can achieve long-term dependencies more easily than regular RNNs.

Due to its special architecture, LSTM is capable of achieving better correlation between historical technical indicators and more recent ones.

4.1.2. Gated Recurrent Units

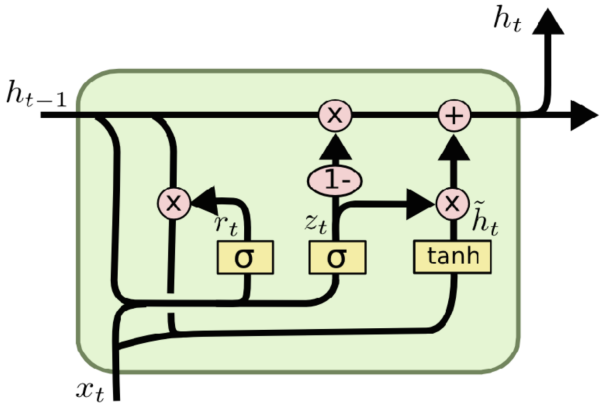

Figure 8: Example of a GRU neural network

Gated Recurrent Units (GRU) neural network , proposed by Chung et al. in 2014 [6], is a variation of LSTM. Instead of having cell state and three gates, GRU only has two gates:

- Reset gate *r*: it uses sigmoid function to combine new input with previous memory.

$$r_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (8)$$

- Update gate *z*: it uses the sigmoid function to decide which portion of new value is relevant

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \quad (9)$$

Because GRU does not have internal memory (or cell state *C*), the output *h* is generated by Eq. 10 and Eq. 11.

$$\widetilde{h_t} = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \quad (10)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \quad (11)$$

4.1.3. Convolution Neural Network

A Convolution Neural Networks (CNN), a class of deep neural networks, has plenty of applications which are widely used in multiple fields and industries. These applications include image and video recognition, natural language processing (NLP), medical image analysis and financial time-series data analysis[5].

A CNN has three types of layers: input layer, hidden layers and output layer. The most important operation in the hidden layers is convolution, which performs the dot product of the input layer matrix *I* and the convolution kernel *K*. The following equation (Eq. 1) represents the output *s*, which is also known as the feature map, if both the input and the kernel are two-dimensional.

$$s[i,j] = (I * K)[i,j] = \sum_m \sum_n I[m,n]K[i-m, j-n] \quad (12)$$

Instead of using regular two-dimensional CNN, we choose one-dimensional CNN for time series forecasting. Because the price movement has only two options: either going up or down. It means that the kernel only slides along one dimension.

Pooling layer and flattening layer are useful in image and video processing; however, these layers are redundant in our model.

Because we want the data as accurate as possible, pooling will lose valuable information. Although we will spend more time on computation, achieving better accuracy is the priority.

Flattening layer converts the output of convolution layers into a one-dimensional array of data. Considering that we only use one-dimensional CNN for time series forecasting, the output of convolution layers is one-dimensional; therefore, we will not use flattening.

4.2. Software Design

In this task, we use NVIDIA T4 GPU on Google Cloud Platform to complete the training test. At first, the accuracy is still high after many training epochs. According to our observation, training accuracy and validation accuracy are almost the same. So, it is impossible to be overfit. Thus, we decide to improve the model space from 256 neurons to 512 neurons to solve the underfit problem, and adopt the dropout method to improve the model robustness (also prevent overfit problem). We decided to use an adaptive learning rate algorithm Adam for all three models, and change epochs from 250 to 100, because it is not necessary to use so many epochs for a fast learning problem. The results in **Section 5** shows our approach is wise, which gives us higher accuracy in both the training set and validation set.

Table 3 shows the low level implementation of three deep learning algorithms. The number of neurons in LSTM and GRU are both 512. In CNN model, filters=16, kernel_size=32 for the first layer, and kernel_size=28 for the second layer. Then, we use the same models for four kinds of cryptocurrencies in different price ranges to show the robustness of our models.

	CNN	LSTM	GRU
Epochs	100	100	100
Number of Hidden Layers	2	1	1
Batch Size	128	256	256
Optimizer	Adam	Adam	Adam
Activation	ReLU	Tanh + Leaky ReLU	Tanh + ReLU
Type of Loss	MSE	MSE	MSE
Dropout	0.5	0.6	0.5

Table 3: Parameters value in each algorithm

5. Results

5.1. Project Results

We use 80% data for the training set, and 20% data for validation set randomly to get prediction results.

Table 4 gives an elaborate description of collected data. The frequency of data collection is 5 minutes which is the same for every cryptocurrencies. Data set period spans more than 5 years which starts from 2016/03/07 to 2021/04/12. The size of the training set varies slightly among different cryptocurrencies due to incompleteness of data collected from Poloniex.

Data in the validation set should be the most updated and representative one, so the validation set period starts from 2020/04/04 to 2021/04/12. For the same reason, the size of the validation set changes slightly.

However, small fluctuation in the size is not a serious problem because the overall size of both sets is big and, therefore, relatively stable.

	Bitcoin (BTC)	Ethereum (ETH)	Monero (XMR)	Ripple (XRP)
Data Collection Frequency	5 minutes	5 minutes	5 minutes	5 minutes
Data Set Period	2016/03/07 03:40:00 – 2020/04/04 16:45:00	2016/03/07 03:40:00 – 2020/04/04 16:50:00	2016/03/07 03:40:00 – 2020/04/04 18:05:00	2016/03/07 03:40:00 – 2020/04/04 23:20:00
Size of the Data Set	400,176	400,176	400,175	400,175
Validation Set Period	2020/04/04 16:50:00 – 2021/04/12 11:50:00	2020/04/04 16:55:00 – 2021/04/12 11:50:00	2020/04/04 18:10:00 – 2021/04/12 11:50:00	2020/04/04 23:25:00 – 2021/04/12 11:50:00
Size of the Validation Set	107,365	107,364	107,349	107,286
Maximum Price	61,050.0	2,189.5	342.6	XRP 1.5

Table 4: Data processing information

Figure 9-11 shows the prediction result of three algorithms on BTC. The blue line is actual value and red rot is predicted value. The predicted results in three models all fit good with the actual value. Figures of prediction for ETH, XMR, and XRP prices are in **Section 9. Appendix**.

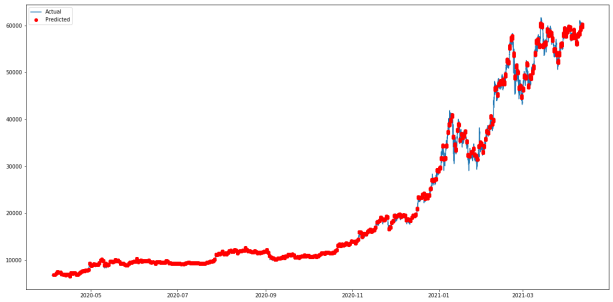

Figure 9: LSTM model on BTC

Figure 10: GRU model on BTC

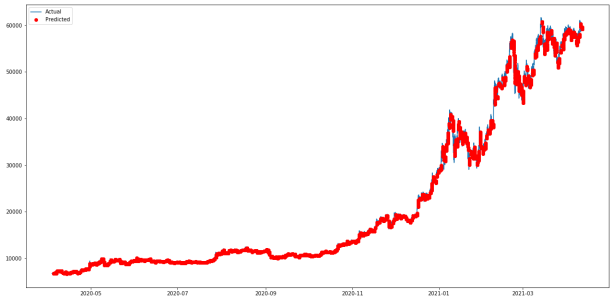

Figure 11: CNN model on BTC

More prediction results of different deep learning algorithms on three other cryptocurrencies can be found in **Section 9.2. Prediction Result on ETH, XMR and XRP**.

Table 5 gives the evaluation of MSE loss in three models for BTC, ETH, XMR, and XRP respectively. As we can see, the MSE losses vary a lot for different cryptos, because of the different price ranges. To solve this problem, we divide MSE losses by square of the max prices to normalize them. The normalized MSE losses have no huge difference in one kind of model. Therefore, we can say our models work well from low value (XRP) to high value (BTC) for different price curves.

Comparing between different models, it is noticeable that CNN model has the highest MSE loss, but, at the meantime, it is the most time-effective model, which only costs 8 seconds training time. GRU model and LSTM model have comparable MSE loss, while GRU is more time-effective (only about 80% training time of that in LSTM model).

To sum up, CNN is cost-effective, and GRU can approach higher accuracy. LSTM is slightly inferior to GRU in both accuracy and training time cost.

		Bitcoin (BTC)	Ethereum (ETH)	Monero (XMR)	Ripple (XRP)
LSTM	**MSE Loss**	84,932.7	138.8	3.0	0.0001
	Normalized MSE	2.3e-5	2.9e-5	2.5e-5	4.6e-5
	Training Time	281s	280s	280s	280s
GRU	**MSE Loss**	46,026.1	93.8	3.0	0.00005
	Normalized MSE	1.2e-5	1.9e-5	2.5e-5	2.3e-5
	Training Time	225s	225s	225s	225s
CNN	**MSE Loss**	595,087.9	1,103.5	22.8	0.0006
	Normalized MSE	1.6e-4	2.3e-4	1.9e-4	2.8e-4
	Training Time	8s	8s	8s	8s

Table 5: Evaluation of LSTM, GRU and CNN models on BTC, ETH, XMR and XRP

5.2. Comparison of the Results Between the Original Paper and Students' Project

In general, the results in both the original paper and our report predict the prices well, while the original paper still has lower accuracy and less comparison transformability for different kinds of cryptocurrencies.

Table 1 shows that, in the original paper, a variation of LSTM, MALSTM-FCN, achieves the highest accuracy (0.542) and CNN with 2 hidden layers has the worst performance (0.435). However, the disparity among the accuracy of MLP (0.537), LSTM (0.535) and MALSTM-FCN (0.542) is not substantial. Therefore, it is

hard to say which algorithm is the best or outperforms the rest.

In our model, we design a new GRU model. It has the best performance and, similarly, CNN is the worst one. GRU outperforms other algorithms because it has the lowest MSE loss in all four different cryptocurrencies.

Moreover, we use a more advanced GPU, NVIDIA T4 GPU, to accelerate the training tasks, which approaches 8s in CNN, 225s in GRU, and 280s in LSTM. However, device type and training time are omitted in the original paper. Hardware computing power indeed matters for deep learning training[7].

If we use one algorithm to predict the price, Bitcoin always has the lowest normalized MSE regardless which algorithm we choose. Because the Bitcoin market is more popular and regularized, more individual and institutional investors entered into this market. Therefore, the market is more stable and more predictable.

5.3. Discussion of Insights Gained

Starting from the data collection and preparation. We substantially increase the sample size. Though MSE losses, we believe that the size of the training set increases more than fourteen-fold.

In addition, we use less epochs (100) than that in the original paper (250), because Adam algorithms can approach the optimal solutions earlier. It saves much training time.

In general, we use much more data (about half a million compared to 35,638), deploy a higher model space (512 neurons compared to 256 neurons) to improve our training accuracy (1e-5 ~ 3e-4 MSE loss). We also use dropout methods to prevent overfit problems, and, thus, improve validation accuracy.

Furthermore, we design GRU models, which are more time-effective and more accurate than LSTM models in the original paper. The reason is that GRU's units are simpler and good enough for the price prediction tasks.

6. Conclusion

By analyzing the MSE loss, normalized MSE and training time, and comparing these parameters to the original paper, we find out that GRU is the best algorithm to predict the future cryptocurrency price if we only consider the technical indicators. CNN uses less training time, but has the worst predictability in both project and original paper. Bitcoin, regardless of its high price, becomes the most predictable cryptocurrency due to its popularity among investors and more regularized trading market. However, no matter which algorithm we use, the MSE loss is substantially high which makes the prediction more difficult. In the future, we are going to add social media indicators and sentiment indicators into our project. Social platforms, like Reddit and Twitter, always have the heated debate on cryptocurrency market and the latest blockchain technology. We strongly believe that the topic discussions and their sub-comments have strong correlation with the prices; therefore, sentiment index will be helpful in improving the predictability of the price movements.

7. Acknowledgement

We would like to thank Prof. Kostic and TAs for their teaching and helping in this project.

8. References

[1] Link to the project GitHub repository. https://github.com/ecbme4040/e4040-2021spring-project-flug-gl2713-zf2261

[2] R. Bhutoria, "The Institutional Investors Digital Asset Survey", Fidelity Digital Assets, June, 2020, pp. 5-6. [Online]. Available: https://www.fidelitydigitalassets.com/articles/institutional-digital-asset-survey-report.

[3] M. Ortu, N. Uras, C. Conversano, G. Destefanis, S. Bartolucci, "On Technical Trading And Social Media Indicators In Cryptocurrency Price Classification Through Deep Learning", in arXiv: 2102.08189, February, 2021.

[4] F. Karim, S. Majumdar, H. Darabi, S. Harford, "Multivariate LSTM-FCNs for Time Series Classification", in arXiv: 1801.04503, July, 2019.

[5] J. Chen, W. Chen, C. Huang, S. Huang and A. Chen, "Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks", 2016 7th International Conference on Cloud Computing and Big Data (CCBD), 2016, pp. 87-92, doi: 10.1109/CCBD.2016.027.

[6] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", presented in NIPS 2014 Deep Learning and Representation Learning Workshop, in arXiv:1412.3555, December 2014.

[7] I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning", The MIT Press, 2016, ISBN: 9780262035613.

9. Appendix

9.1 Individual Student Contributions in Fractions

	gl2713	zf2261
Last Name	Lan	Fu
Fraction of (useful) total contribution	1/2	1/2
What I did 1	Build modes	Methodology
What I did 2	Completed code	Implementation
What I did 3	Analyzed result	Literature Overview

9.2 Prediction Result on ETH, XMR and XRP

Figure 12-14 shows the prediction result of three algorithms on ETH. The blue line is actual value and red rot is predicted value. The predicted results in three models all fit good with the actual value.

Figure 12: Prediction based on LSTM model for ETH prices.

Figure 13: Prediction based on GRU model for ETH prices.

Figure 14: Prediction based on CNN model for ETH prices.

Figure 15-17 shows the prediction result of three algorithms on XMR.

Figure 15: Prediction based on LSTM model for XMR prices.

Figure 16: Prediction based on GRU model for XMR prices.

Figure 17: Prediction based on CNN model for XMR prices.

Figure 18-20 shows the prediction result of three algorithms on XRP.

Figure 18: Prediction based on LSTM model for XRP prices.

Figure 19: Prediction based on GRU model for XRP prices.

Figure 20: Prediction based on CNN model for XRP prices.