2020 International Conference on Smart Technology and Applications (ICoSTA)

Prediction of Bitcoin Price Change using Neural Networks

1st Rahmat Albariqi
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
rahmat.albariqi@mail.ugm.ac.id

2nd Edi Winarko*
Department of Computer Science and Electronics
Universitas Gadjah Mada
Yogyakarta, Indonesia
ewinarko@ugm.ac.id

2020 International Conference on Smart Technology and Applications (ICoSTA) 978-1-7281-3083-5/20/$31.00 ©2020 IEEE 10.1109/ICoSTA48221.2020.1570610936

***Abstract*—In recent years, Bitcoin is rising and become an attractive investment for traders. Unlike stocks or foreign exchange, Bitcoin price is fluctuated, mainly because of its 24-hours a day trading time without close time. To minimize the risk involved and maximize capital gain, traders and investors need a way to predict the Bitcoin price trend accurately. However, many previous works on cryptocurrency price prediction forecast short-term Bitcoin price, have low accuracy and have not been cross-validated**

This paper describes the baseline neural network models to predict the short-term and the long-term Bitcoin price change. Our baseline models are the Multilayer Perceptron (MLP) and the Recurrent Neural Networks (RNN) models. Data used are Bitcoin's blockchain from August 2010 until October 2017 with 2-days period and the total amount of 1300 data. The models generated are predicting both for short-term and long-term price change, from 2-days until 60-days.

The result shows that long-term prediction has a better result than short-term prediction, with the best accuracy in Multilayer Perceptron when predicting the next 60-days price change and Recurrent Neural Networks when predicting the next 56-days price change. Multilayer Perceptron outperforms Recurrent Neural Networks with accuracy of 81.3 percent, precision 81 percent, and recall 94.7 percent.

Index Terms*—*cryptocurrency, neural networks, multilayer perceptron and recurrent neural networks

I. Introduction

Market prices forecasting is very interesting and challenging both for investors and researchers due to the many uncertainties involved and lots of variables that influence the market, such as economic conditions and political events [1]. In recent years, the market is not only about the stock and foreign exchange (forex) but also about cryptocurrency. ODO defines cryptocurrency as a digital currency in which encryption techniques are used to regulate the generation of units of currency and verify the transfer of funds, operating independently of a central bank.

The first cryptocurrency was Bitcoin, which began trading in January 2009 [2]. Bitcoin is the largest cryptocurrency in the world. It is a peer-to-peer electronic cash system that allows online payments to be sent directly from one party to another without going through a financial institution [3].

Reference [4] analyzed the Bitcoin blockchain data to predict the price of Bitcoin using SVM and ANN (Multilayer Perceptron), which score 55% accuracy. Random Forest, SVM, and Binomial Logistic algorithms are used in to predict short-term Bitcoin price and achieve high accuracy result of 97% in [5], but it is stated in [6] that this research has one limitation, in which the result was not cross-validated so that it may have overfit the data and one cannot be sure if the model will generalize. Therefore, [6] uses LSTM (Long Short-Term Memory) network and achieves an accuracy of 52%. Most of the previous works on cryptocurrency price prediction forecast short-term Bitcoin price, have low accuracy, and have not been cross-validated, which gives a high risk for traders to trust the model.

Unlike cryptocurrency prediction, stock market prediction research had many satisfying results, mainly because they focused not only on a short-term prediction but also on long-term prediction. Reference [7] uses the Random Forest algorithm with the result of 75% in Precision, Recall, and F-score to predict the 'good' and 'bad' stocks in a one-year period. Similarly, [1] achieves really high accuracy using Random Forest and achieve accuracy in the range of 85-95% for long term price prediction. Also, [8] uses SVM, QDA, Logistic Regression, and GDA to predict the future price change and compare the result to predict both short-term and long-term price change with their short-term result only achieves 58% accuracy but their long-term result with time-ahead of 44-days achieve accuracy of 79.3%.

Inspired by works done on stock market price prediction field that achieve better accuracy than the cryptocurrency price prediction field by predicting the long-term price, this research tries to generate neural networks model that can be used to predict the price movement of Bitcoin not only in the short-term but also in the long-term. This research is part of the first author's undergraduate thesis [9].

II. Methodology

The purpose of this study is to generate baseline neural network models that can predict the future price change of Bitcoin both in short-term and long-term. In predicting the future price in the market, traders and investors used technical analysis or fundamental analysis. In this research, fundamental data used are the Bitcoin blockchain data obtained from *blockchain.info*. There are 35 possible blockchain features that can be used for the input of the neural network, but this research only used 14 of them based on research by [5], as shown in Table I.

Each feature downloaded from *blockchain.info* is a single CSV file. Each file contains two columns, the first column is a timestamp, and the second column is the feature value. The number of rows in each file 1324, in which the time stamp

*Corresponding author

starts from 02 August 2010 and ends on 30 October 2017 with a 2-days period. In order to create a training data file, these 14 CSV files are merged, the resulting file contains 1324 rows and 15 columns (timestamp and 14 features).

TABLE I. FOURTEEN INPUT FEATURES SELECTED

	Feature	Definition
1	Block Size	Average block size in MB
2	Cost per Transaction	Miners revenue divided by the number of transactions in USD.
3	Difficulty	A relative measure of how difficult it is to find a new block.
4	Hash Rate	The estimated number of tera hashes per second the Bitcoin network is performing.
5	Market Capitalization	The total USD value of bitcoin supply in circulation.
6	Median Confirmation Time	The median time for a transaction to be accepted into a mined block.
7	Miners Revenue	The total value of coin base block rewards and transaction fees paid to miners in USD.
8	Number of Orphaned Blocks	The total number of blocks mined but ultimately not attached to the main Bitcoin blockchain.
9	Number of Transaction	Total number of confirmed transactions per day.
10	Number of Transaction per Block	The average number of transactions per block.
11	Unique Addresses	The total number of unique addresses used on the Bitcoin blockchain.
12	Bitcoins in Circulation	The total number of bitcoins that have already been mined.
13	Trade Volume	The trading volume on major bitcoin exchanges.
14	Transaction Fees	The total value of all transaction fees paid to miners.

Both input and output datasets are grouped into various time-window, that is, 3, 5, and 7. There are 1324 timesteps in the data, but only 1300 of them are actually predicted because of the time-window grouping. Z-score normalization is applied to the input dataset to standardize the data. Before training the model, the data were split using a *k*-fold cross-validation method with $k = 5$.

Algorithm 1 Output Label Calculation

if $MarketPrice(t + x) \geq 1.01 * MarketPrice(t)$ **then**
    $label \leftarrow [1, 0]$
else
    $label \leftarrow [0, 1]$
end if

Fig. 1. Output label pseudocode

The output label is binary and represented using one-hot encoding value. If the future market price increases, the label is [0, 1]; if it decreases, the label is [1, 0]. The algorithm used to assign the label shown on Fig. 1, where *t* is the current time, and *x* is the time ahead.

The neural network models were trained using various epochs values, with different time-windows, and are used to predict the price change for both short-term and long-term. There is 30 predicted value (time-ahead) from 2-days until 60-days with 2-days period. Learning rate, hidden state, and learning algorithm were experimented to find the best value by looking at the lowest validation loss they produce.

III. NEURAL NETWORK MODELS

In this research, Multilayer Perceptron (MLP) and Recurrent Neural Network (RNN) were used as baseline models because both models already used in previous research with MLP [4] and RNN [6] show that the networks were capable of predicting Bitcoin price change, but both research only achieve accuracy less than 60% which still need improvement.

A. Multilayer Perceptron

The model used for MLP is based on research by [10], which uses 2-hidden layer feedforward networks with a formulated number of hidden nodes.

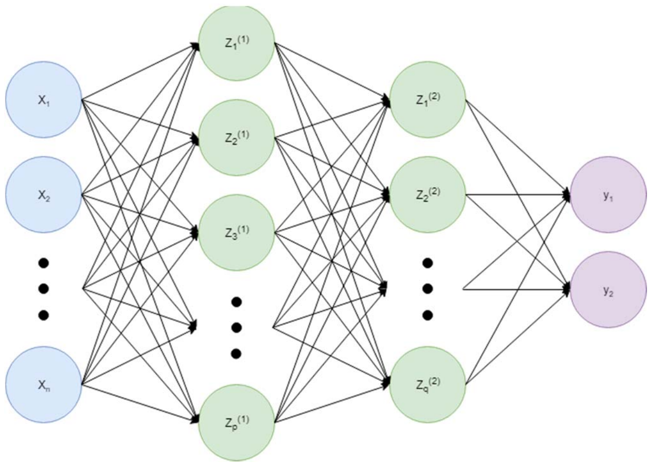

Fig. 2. Multilayer Layer Perceptron architecture

Fig. 2 visualizes the architecture of MLP with *X* as input layer, Z_1, and Z_2 as the hidden layer and *y* as output layer.

- The input layer has *n* nodes with *n = the number of features x time-window.* Since there are *14* features (Table I), if the size of *time-window = 3*, then the input layer has $n = 42$ nodes.
- The number of nodes in the hidden layer is calculated using formulas based on the number of input node *n* and output node *m*. For example, if the first hidden layer has *p* nodes, and the second hidden layer has *q* nodes, then *p* and *q* can be calculated using (1) and (2), respectively. An activation function used in the hidden layer is ReLu.

$$p = \sqrt{(m+2)\,n} + 2\sqrt{n/(m+2)} \tag{1}$$

$$q = m\sqrt{n/(m+2)} \tag{2}$$

- Output layer has 2 node representing one-hot encoding value of decrease [1, 0] or increase [0, 1]. Softmax is used as an activation function in this layer.

B. *Recurrent Neural Network*

Recurrent Neural Network (RNN) allows the network to learn from its previous state, which makes this network really useful for time-series data. The architecture used was many-to-one with various experimented time-window.

Fig. 3 visualizes the architecture of RNN with X as input layer, Z_1, and Z_2 as the hidden layer and y as output layer.

- The input layer has 14 nodes (feature dimension).
- Hidden layer nodes were varied between 10, 20, 30, and 40 to find the hidden nodes with the lowest validation loss using ReLu as an activation function. The identity matrix was used for RNN initial weight in hidden nodes, which works well with long-term dependencies [11].
- Output layer has 2 nodes, representing one-hot encoding value of decrease [1, 0] or increase [0, 1]. Softmax is used as an activation function in this layer.

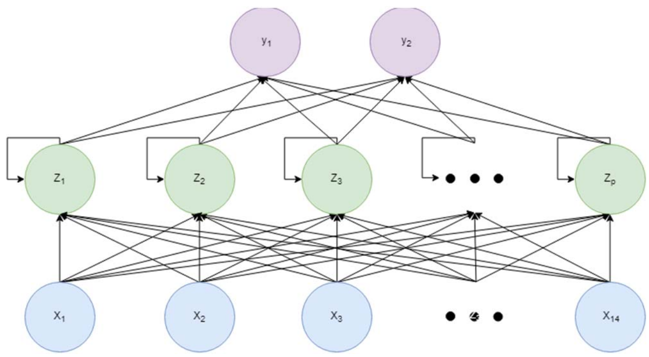

Fig. 3. Recurrent Neural Network architecture

IV. Experiments and Discussion

The experiment is performed in two steps. The first step is to find the best hyperparameters. Then, based on the best hyperparameters, the second step is used to compare the models with various time-windows and time-ahead.

A. *Hyperparameter Experiment*

Hyperparameters that were experimented are learning rate, hidden node, and learning algorithm.

1) Learning rate: in the experiment, the value learning is varied to 0.01, 0.001, and 0.0001 for both MLP and RNN algorithms. Other hyperparameter values are constant, as follows: the value of epoch is 50, the number of hidden nodes for RNN is 10, the learning algorithm used is Adam. The results of this experiment show that the best learning rate for both MLP and RNN is 0.01, with validation loss 0.58 and 0.62, respectively.

2) Hidden node: the experiment on the number of hidden nodes is only for RNN. The number of nodes used in MLP is determined using the formula. The constant hyperparameters are learning rate, epoch, and learning algorithm, with a value of 0.01, 50, and Adam, respectively. Lowest validation loss is found with 20 hidden nodes with a value of 0.611.

3) Learning algorithm: There are four learning algorithms that are compared, that is, Adam, AdaDelta, RMSProp, and Stochastic Gradient Descent (SGD). The learning rate and the hidden unit are set to 0.01 and 20, respectively. Adam shows the lowest validation loss for both MLP and RNN, with a value of 0.58 and 0.611, respectively.

Hyperparameter used is summarized in Table II for MLP and Table III for RNN. There are constants that are already defined in this research, such as dropout rate and batch size. These hyperparameters are used to find the accuracy, precision, and recall for epoch 50, 100, 150, and 200.

TABLE II. MLP HYPERPARAMETER

Hyperparameter	**Value**
Learning Rate	0.01
Learning Algorithm	Adam
Batch Size	128
Hidden Layer 1 Node Size	$\sqrt{(m+2)N} + 2\sqrt{N/(m+2)}$
Hidden Layer 2 Node Size	$m\sqrt{N/(m+2)}$
Dropout	0.5

TABLE III. RNN HYPERPARAMETER

Hyperparameter	**Value**
Learning Rate	0.01
Learning Algorithm	Adam
Batch Size	128
Hidden State Size	20
Dropout	0.5

B. *Result and Discussion*

In the second step of our experiments, the performance of the model is measured by the model's accuracy, precision, and recall, for each time-window of 3, 5, and 7, and time-ahead of 2-days, 4-days, 6-days until 60-days. In this research, short-term price prediction is defined as time-ahead between 2-days and 30-days and long-term price prediction from 32-days until 60-days. The highest result for each epoch is compared as in Table IV for short-term and Table V for the long-term; the highlighted row is used to show the epoch with the highest accuracy for each model.

Table IV shows the highest accuracy is 70.04% on MLP and 67.56% on RNN. Both MLP and RNN achieved their highest short-term accuracy using a time-window of 3. It means that increasing the time-window does not always the case to increase the accuracy of short-term prediction.

Table V shows the highest accuracy for long-term price prediction, with 81.3% on MLP and 77.2% on RNN. The results also show a similar pattern with short-term prediction on Table IV, with both highest accuracy achieved using a time-window of 3.

Table VI put together the highest accuracy for short-term and long-term prediction both for MLP and RNN. Long-term prediction achieved better accuracy than the short-term one, both in MLP and RNN. This result shows that long-term prediction is easier to predict and can achieve better accuracy than the short-term one.

TABLE IV. SHORT-TERM HIGHEST ACCURACY

Neural Network	Epoch	Highest Accuracy		
		Time-window	*Time-ahead*	*Value*
Multilayer Perceptron	50	5	2-days	67.68%
	100	3	26-days	70.04%
	150	7	28-days	68.77%
	200	7	26-days	69.61%
Recurrent Neural Networks	50	5	2-days	67.31%
	100	3	28-days	66.70%
	150	5	2-days	66.74%
	200	3	2-days	67.56%

TABLE V. LONG-TERM HIGHEST ACCURACY

Neural Network	Epoch	Highest Accuracy		
		Time-window	*Time-ahead*	*Value*
Multilayer Perceptron	50	7	58-days	78.0%
	100	5	60-days	80.0%
	150	5	58-days	80.0%
	200	3	60-days	81.3%
Recurrent Neural Networks	50	5	60-days	75.2%
	100	3	60-days	74.8%
	150	5	58-days	76.2%
	200	3	56-days	77.3%

TABLE VI. MLP AND RNN HIGHEST ACCURACY

Neural Network	Group	Highest Accuracy
Multilayer Perceptron	Short-term	70.04%
	Long-term	81.3%
Recurrent Neural Networks	Short-term	67.56%
	Long-term	77.3%

Accuracy alone is not really good evaluation benchmark for imbalanced dataset. The model probably suffers the accuracy paradox in which it can only predict the majority of the output label that fed into its networks in training session. The solution to finding out whether our model suffer from accuracy paradox is to calculate the precision and recall score for both networks.

Table VII shows the highest accuracy for each model with its precision and recall. Both precision and recall have value matching its accuracy. Thus, it can be certain that the models do not suffer from accuracy paradox. In this research, the highest accuracy is achieved by MLP with accuracy of 81.3%, precision of 81.0% and recall of 94.7%.

TABLE VII. FINAL RESULT

Neural Network	Accuracy	Precision	Recall
Multilayer Perceptron	81.3%	81.0%	94.7%
Recurrent Neural Networks	77.3%	76.6%	95.8%

V. CONCLUSION

In this research, we have studied the performance of MLP and RNN models for prediction of bitcoin price change. The long-term price prediction achieves higher result than the shorter one both in MLP and RNN. Just like in stock market price prediction, long-term price prediction also shows high accuracy in cryptocurrency price prediction with result of accuracy in range of 60-80%. The performing model in this research is Multilayer Perceptron with time-window of 3 and 200 epochs with accuracy of 81.3%, precision of 81% and recall of 94.7%.

Further research can be done by feature engineering to make neural networks learn faster and better, for example, by combining multiple features into a single feature, and removing unwanted features. There are various hyperparameters that can be tweaked, such as dropout rate, weight initialization, and batch size. Other direction in our future research is to explore more advance deep learning models, especially sequences to sequence models and temporal convolutional neural networks along with their architectural variants.

REFERENCES

[1] L. Khaidem, S. Saha, and S.R. Dey, “Predicting the direction of stock market prices using Random Forest,” arXiv preprint arXiv:1605.00003, 2016.

[2] E. Dourado and J. Brito, Cryptocurrency, The New Palgrave Dictionary of Economics, 2014.

[3] Cryptocurrency, In Oxford Dictionaries, 2019. [online] Available at: https://www.oxfordlearnersdictionaries.com/definition/english/cryptocurrency

[4] A. Greaves and B, Au, Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin, Stanford University, 2015.

[5] I. Madan, S. Saluja, and A. Zhao, Automated Bitcoin Trading via Machine Learning Algorithms, Stanford University, 2015.

[6] S. McNally, Predicting the price of Bitcoin using Machine Learning, Doctoral Dissertation, Dublin, National College of Ireland, 2016.

[7] N. Milosevic, “Equity forecast: predicting long term stock price movement using machine learning,” arXiv preprint arXiv:1603.00751, 2016.

[8] Y. Dai and Y. Zhang, Machine Learning in Stock Price Trend Forecasting, Stanford University, 2013.

[9] R. Albariqi, Predicting Bitcoin Price Change Using Neural Network, Undergraduate Thesis, Universitas Gadjah Mada, 2017.

[10] G.B. Huang, “Learning capability and storage capacity of two-hidden-layer feedforward networks,” IEEE Trans. Neural Netw., 14(2), pp.274-281, 2003.

[11] Q.V. Le, N. Jaitly, and G.E. Hinton, “A simple way to initialize recurrent networks of rectified linear units,” arXiv preprint arXiv:1504.00941, 2015.