ICIC Express Letters
Part B: Applications
Volume **13**, Number **8**, August **2022**
 ISSN 2185-2766
pp. 853–860

COMPARATIVE STUDY FOR DATA NORMALIZATION METHODS ON PREDICTING CRYPTOCURRENCY PRICE

Jantima Polpinij[1,*], Bancha Luaphol[2] and Khanista Namee[3]

[1]Intellect Lab
Department of Computer Science
Faculty of Informatics
Mahasarakham University
Kamrieng, Kantharawichai District, Mahasarakham 44150, Thailand
[*]Corresponding author: jantima.p@msu.ac.th

[2]Department of Digital Technology
Faculty of Administrative Science
Kalasin University
13, Songpuay, Namon District, Kalasin 46230, Thailand
bancha.lu@ksu.ac.th

[3]Faculty of Industrial Technology and Management
King Mongkut's University of Technology North Bangkok
1518, Pracharat 1 Road, Bang Sue District, Bangkok 10800, Thailand
khanista.n@fitm.kmutnb.ac.th

Received December 2021; accepted February 2022

Abstract. *Feature values on highly different scales can decrease model performance prediction of cryptocurrency prices. Therefore, this work aimed to present a comparative study for data normalization in order to recognize the most appropriate method of data normalization for cryptocurrency price prediction. Three common data normalization methods often used in regression analysis as z-score, min-max and log scaling were compared. These data normalization methods were performed in the pre-processing data step, with scaled feature values used to develop the predictive models based on Support Vector Regression (SVR) and Long Short-Term Memory (LSTM). After evaluating the results by Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), the evaluation showed that the z-score method returned slightly better results than the min-max and log scaling methods. If considering the computational time, the z-score method required a slightly longer time because it calculates the mean and standard derivation values before the scaling feature values.*
Keywords: Data normalization, Cryptocurrency, Price predication, Z-score, Min-max, Log scaling, SVR, LSTM

1. **Introduction.** Cryptocurrencies, as types of crypto-assets, are digital or virtual exchange media that use cryptography to protect and verify transactions through a secure network system known as blockchain technology [1]. The middle trading price varies according to market forces. A cryptocurrency acts as a medium for exchanging value through the Internet. Most investors in this field focus on speculating future cryptocurrency value [2].

Currently, there are over 2,000 different cryptocurrencies, and each has a different level of popularity and trust [3]. The five most valuables by market cap, calculated as the number of coins multiplied by the trading rate are Bitcoin, Ethereum, Ripple, Bitcoin Cash and Litecoin. Bitcoin, with the highest market value, is the most popular [4], while

DOI: 10.24507/icicelb.13.08.853

Ethereum comes a close second and can be used for transactions of more than 116 of the world's most famous companies including Intel, Microsoft, JP Morgan, and Toyota [5].

A cryptocurrency does not provide value such as gold or any other guaranteed asset like regular currency. Its value is set according to the market supply and demand. This can vary in popularity; therefore, unit prices of cryptocurrencies are volatile, sensitive and highly changeable over time. As a result, if the price of cryptocurrencies can be predicted in advance, this information can be used for beneficial investment decision-making [6-10]. Many studies have attempted to predict the price of cryptocurrencies over the past decade, using data mining as the main process, to discover a potentially useful predictive model from cryptocurrency datasets through the process of data mining [11].

The main tasks in the data mining process are data preparation, modeling, and evaluation [12]. Data preparation is an important processing step in data mining [13,14]. It eliminates noise and transforms the data to a suitable format for data analysis and modeling using data mining algorithms such as K-nearest neighbor, support vector regression, random forest and artificial neural networks to develop a model to predict future cryptocurrency prices. Finally, after modeling, the performance of the predictive model is evaluated.

The data preparation stage is very significant in predictive data mining, and this process involves over 80% of the data analytics tasks [13,14]. Data preparation includes many important steps and data normalization is fundamental in this part of the processing [15]. Data normalization is used to transform feature/attribute values in a dataset to a similar or the same scale [15-17]. This phase removes outliers and ensures that all features/attributes equally impact the result during the prediction process. Data normalization techniques include min-max scaling [17], z-score (or standard scaling) [17], and log scaling [18]. Many studies for cryptocurrency price prediction applied min-max scaling for normalizing data, while a few utilized z-score scaling. Also, some studies such as [19], improved an existing normalization method to design a new one for improving classification accuracy. However, these studies did not provide sufficient reasons for using specific data normalization techniques.

Here, we compared data normalization techniques to identify the most appropriate for use in a framework to predict the price of cryptocurrencies. Four data normalization techniques were compared based on predictive models developed by Support Vector Regression (SVR) and Long Short-Term Memory (LSTM). Experimental results were evaluated by Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

2. **Data Normalization Techniques.** To the best of our knowledge, common data normalization techniques used for predicting the price of cryptocurrencies are presented as follows.

Min-Max Normalization: This technique, also known as min-max scaling, converts floating-point feature values as 0 and 1 (or -1 to $+1$ in some cases) from their natural range to a standard range [17]. Minimum and maximum feature values are 0 and 1, respectively. The technique scales floating-point feature values to a range using the formula:

$$x' = (x - x_{\min})/(x_{\max} - x_{\min}) \tag{1}$$

where x is considered as a feature value, while the approximate lower and upper bounds on the dataset with few or no outliers are $x_{\min}$ and $x_{\max}$, respectively. The dataset is commonly normalized as the scikit-learn object MinMaxScaler through Python.

Log Scaling: This technique takes the logarithm (log) of floating-point feature values to convert a wide data range to a narrow data range [18]. It can use the following simple formula.

$$x' = \log(x) \tag{2}$$

where x is considered as the feature value.

Z-score: This technique, also known as *data standardization* or *standard scaling*, is a scaling approach used to represent the number of standard deviations away from the mean [17]. The z-score approach helps to ensure that feature distributions have a mean of 0 and a standard deviation of 1. This technique is useful when there are few outliers, and can be represented as

$$x' = (x - \mu)/SD \tag{3}$$

where x is considered as a feature value, while μ is a mean and SD is a standard deviation.

3. **Main Results.** This section explains how to identify the most appropriate data normalization technique for use in a framework to predict the price of cryptocurrencies utilizing Python and scikit-learn.

3.1. **Datasets.** This section describes datasets used for this study. Three cryptocurrency datasets download from https://www.cryptocompare.com, which are shown in Table 1. Those datasets are represented in the csv format and example of bitcoin dataset can be shown as Figure 1.

TABLE 1. Datasets

Cryptocurrency	Total number of instances in each dataset
Bitcoin (BTC)	2,000
Ethereum (ETH)	2,000
Litecoin (LTC)	2,000

	high	low	open	volumefrom	volumeto	close	conversionType	conversionSymbol
time								
2019-04-05	5326.07	5113.60	5021.84	4067491.58	2.141033e+10	5263.77	multiply	BTC
2019-04-06	5380.86	5209.99	5263.77	3675655.57	1.944638e+10	5290.59	multiply	BTC
2019-04-07	5667.96	5416.24	5290.59	4130987.52	2.320080e+10	5616.28	multiply	BTC
2019-04-08	6040.28	5774.30	5616.28	5434639.01	3.183822e+10	5858.39	multiply	BTC
2019-04-09	5823.86	5707.15	5858.39	4225973.63	2.422547e+10	5732.52	multiply	BTC
...	...	...	...	...	...	...	...	...

FIGURE 1. Examples of Bitcoin dataset

These datasets consist of nine common features as cryptocurrency name, time stamp, highest price during the day (high), lowest price during the day (low), the number of assets traded at the start of the day (volume from), the number of assets traded at the end of the day (volume to), types of exchange rate (conversion type), open price (open), and close price (close). The time stamp of these datasets is between August 13th, 2016 and January 3rd, 2022.

However, high, low, volume from, volume to, and open are used as the predictive variables to learn a mapping from input variables to a target variable (close). The holdout method is conducted on all the models. 80% is used as the training set and the rest is used as the testing set.

3.2. **Framework for predicting cryptocurrency price.** To identify the most appropriate data normalization technique, three common normalization techniques are compared through a framework for predicting cryptocurrency price detailed as follows.

Stage 1: Missing Value Handling

This study used the simple technique of discarding any instance detected containing a feature with a missing value. Data values were missing for less than 1% of the total instances. Therefore, discarding instances containing features with missing values had little effect on the overall data analysis.

Stage 2: Data Normalization

Three techniques often used to normalize cryptocurrency data were compared. In general, the best normalization method depends on the data to be normalized. However, we aimed to recognize the most appropriate common technique used for predicting the price of many cryptocurrencies.

Stage 3: Predictive Model Development

Two algorithms as Support Vector Regression (SVR) and Long Short-Term Memory (LSTM) were compared to develop the predictive models.

Support Vector Regression (*SVR*): It is a modification of Support Vector Machines (SVM) [20]. SVR was introduced to solve regression problems and identify a function that approximates mapping from an input to an output domain as real numbers based on a training sample [20]. The basic idea behind SVR is to find the best fit line within a threshold as a hyperplane having the maximum number of data points. The threshold is the distance between the hyperplane and boundary line. The basic principle of SVM uses w as weight and b as bias. When any vector x is an input vector, the value of y can be estimated using the following equation:

$$y = w^T \cdot x + b \tag{4}$$

SVR also has an additional adjustable parameter ε, called the ε-insensitive errors. Its value is used to determine the width of the tube around the predictive function (or hyperplane or regression line). Data points falling inside this tube are regarded as correct predictions. The regularization constant C is a penalty parameter of the error as a tradeoff between the risk and the regularized term, while the slack variables (ξ_i and ξ_i') are applied to calculating the difference between training data outside the sensitive zone, where $i = 1, 2, \ldots, n$. The SVR can be formulated as

$$\textit{Minimize}: \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i') \tag{5}$$

$$\textit{Subject to}: \quad \begin{cases} y_i - w^T K(x_i) - b \leq \varepsilon + \xi_i' \\ w^T K(x_i) + b - y_i \leq \varepsilon + \xi_i' \\ \xi_i, \xi_i' \geq 0,\ i = 1, \ldots, n \end{cases} \tag{6}$$

where K is the kernel function used to map the data point x_i into high-dimensional feature space. This study applied the Radial Basis Function (RBF) kernel for the SVR algorithm, represented as follows:

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right) \tag{7}$$

Therefore, instead of transforming the points x_i and x_j to very high dimensions and calculating the dot product there, K can be calculated directly from two samples x_i and x_j, represented as feature vectors in some input space, assuming σ as a free parameter.

Long Short-Term Memory (*LSTM*): It is a modification of a Recurrent Neural Network (RNN) [21]. LSTM is useful for modeling sequence data because it retains an internal state to keep track of the data it has already seen [21]. However, there are connection differences between the hidden layers and memory cells of hidden layer structures. LSTM consists of three main layers as forget gate layer, input gate layer and output gate layer. Let f_h be the activation function of the hidden layer (e.g., tanh or sigmoid function), f_y be the activation function of the output layer (i.e., softmax function), W_h be the weight matrix of the hidden layer, while h_t be the hidden state and U_h be the transition matrix (or hidden-state-to-hidden-state matrix). An architecture of LSTM can be presented as Figure 2.

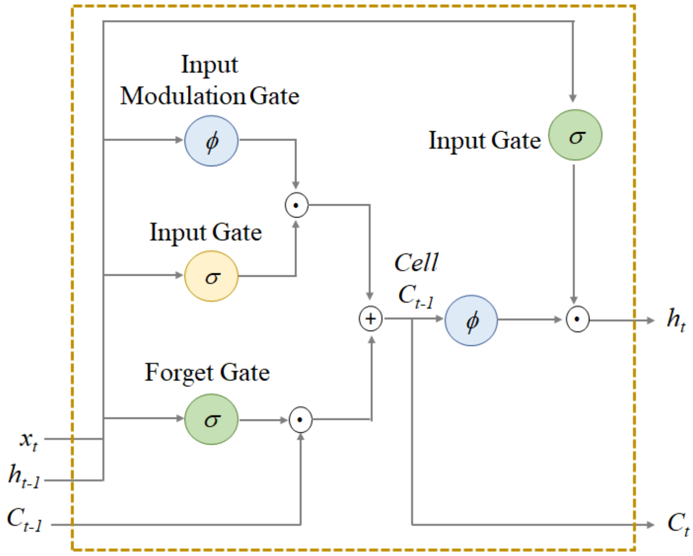

FIGURE 2. An architecture of LSTM

The fomulae of LSTM can be represented as

$$h_t = f_h(U_h h_{t-1} + W_h x_t + b_h) \tag{8}$$

$$y_t = f_y(W_y h_t + b_y) \tag{9}$$

The forget gate layer computes a value between 0 and 1 using a sigmoid function from input x_t, and the current hidden state (h_t). If the forget gate returns 0, then delete the original cell state but if the value obtained from the forget gate is 1, then continue to keep this cell state.

$$f_t = \sigma\left(W_{x^f} x_t + W_{h^f} h_{t-1} + b_f\right) \tag{10}$$

Next, the tanh function is utilized as a nonlinear function to the sigmoid (0-1) output function, and applied to calculating a threshold output in the range $[-1, 1]$.

The input gate layer determines which entries in the cell state to update by computing 0-1 sigmoid output, and then determines the number to add/delete from these entries by computing the tanh output function (value -1 to 1) of the input and hidden states. Then, the forget gate and input gate cell states that maintain a vector C_t in the same dimension as the hidden state (h_t) are updated and scaled to -1 to 1 using the tanh function, formulated by Equation (11).

$$g_t = \tanh\left(W_{x^c} x_t + W_{h^c} h_{t-1} + b_c\right) \tag{11}$$

Then,

$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \tag{12}$$

If f_t is 0, the original cell state is removed. Then, C_{t-1} is not used to update the cell state. However, if f_t is 1, C_{t-1} is maintained for updating the cell state. Consider i_t as an output from the input gate. If the value of i_t is 1, the value of i_t is used to update the cell state. However, if the value of i_t is 0, we cannot use the value of i_t to update the cell state. Finally, a new value of C_t is obtained.

To determine which elements of the cell state to "*output*", the output gate layer calculates a sigmoid written as Equation (13).

$$o_t = \sigma\left(W_{x^o} x_t + W_{h^o} h_{t-1} + b_o\right) \tag{13}$$

Finally, we obtain the output as the h_t value for next sequence, formulated by Equation (14).

$$h_t = o_t \odot \tanh(C_t) \tag{14}$$

If the output gate gives the o_t value as 0, then the value of h_t will also be 0 and will not send any values. At the same time, if the o_t value is 1, we will calculate h_t and be sent outside.

3.3. **Setting algorithms for training predictor models.** For modeling the cryptocurrency price predictors, the RBF was used as the kernel function for the SVR algorithm. Meanwhile, in LSTM algorithm, the batch size of the LSTM setting was kept constant at 34, the number of epochs was 100, the value of dropout was 0.2 and the window length was 5. The number of neurons was 100 and the number of densities was 1. We default '*Adam*' as the optimization function, and use the Rectified Linear Unit (ReLU) as the activation function.

4. **Results.** Two common metrics as Mean Absolute Error (MAE) [22] and Root Mean Squared Error (RMSE) [21] were applied to evaluating their prediction performance of the closing price of cryptocurrencies. Many data normalization techniques have been applied to transforming features on a similar scale. MAE is a widely used linear method to evaluate the difference between predicted and actual values, while RMSE is a quadratic scoring rule that determines the average magnitude of the error, and calculates the prediction error by measuring the distance of data points from the regression line. Experimental results are shown in Table 2.

Table 2. The experimental results

Datasets	**Normalization techniques**	**SVR with RBF kernel**		**LSTM**		**Computational time for data normalization (ms.)**
		MAE	**RMSE**	**MAE**	**RMSE**	
BTC	Z-score	0.0425	0.0531	0.0442	0.0588	0.000095
	Min-max scaling	0.0426	0.0531	0.0460	0.0601	0.000045
	Log scaling	0.0702	0.0818	0.0732	0.0933	0.000010
	Without normalization	0.0736	0.0930	89134.36	169312.78	0
ETH	Z-score	0.0589	0.0800	0.0618	0.0846	0.000102
	Min-max scaling	0.0602	0.0807	0.0732	0.1130	0.000068
	Log scaling	0.0658	0.0805	0.1235	0.1553	0.000011
	Without normalization	0.0920	0.1224	113496.32	192686.25	0
LTC	Z-score	0.0619	0.0824	0.0638	0.0854	0.000098
	Min-max scaling	0.0620	0.0824	0.0646	0.0877	0.000050
	Log scaling	0.0495	0.0667	0.0917	0.1177	0.000011
	Without normalization	0.1080	0.1396	123181.06	170729.19	0

Modeling future cryptocurrency prices without normalizing the predictive variables returned poor results for all models (Table 1). Performing data normalization before training the predictive models gave better results than modeling predictors without data normalization. Data normalization improved regression predictive modeling performance, especially for feature values with diverse ranges. Data normalization improved model training stability and also assisted machine learning using gradient descent algorithms (e.g., SVR) to converge to the global minima faster and better, thereby avoiding algorithm bias to one feature based on its representation.

Table 3 presents average scores of MAE and RMSE for each model. Results show that the z-score method was the most effective, while the log scaling method was least effective.

Table 3. The average scores of MAE and RMSE for each model

Algorithms	Average scores of MAE			Average scores of RMSE		
	Z-score	Min-max	Log	Z-score	Min-max	Log
SVR with RBF kernel	0.0544	0.0549	0.0618	0.0656	0.0721	0.0763
LSTM	0.0566	0.0613	0.0961	0.0763	0.0869	0.1221

The z-score scaling method returned better results than the other two methods because it considered the mean value as well as the variability in a set of raw scores. Consequently, this method effectively handled outliers and gave a more accurate representation of data distribution using standard deviation to consider the variability in a set of raw scores. The min-max scaling method does not adequately handle outliers and operates with all features having the same scale, while the z-score scaling method does not return the exact same scale. In this study, the z-score scaling method also requires more computation time than other normalization techniques for scaling feature values because it calculates the mean and standard derivation values before the scaling feature values.

5. **Conclusions.** Machine learning algorithms were applied to developing a model for predicting cryptocurrency prices and to determining trends by comparing data point features. Feature values on highly different scales often reduce model performance prediction. To handle this issue, data normalization is required during the data pre-processing step, also known as feature scaling, to transform the range of data features. Data normalization is a necessary pre-processing step for predictive modeling of wide-ranging values when using machine learning algorithms such as SVR and LSTM. This study compared data normalization methods to predict the price of cryptocurrencies. Three data normalization methods as z-score, min-max and log scaling were examined using SVR and LSTM. Results were also compared with models developed without normalizing the data. Predictive models that did not apply data normalization method returned the poorest results if comparing to the predictive models that apply data normalization methods. Meanwhile, z-score and min-max scaling methods returned better results than the log scaling method. However, the z-score method gave slightly better results than the min-max method. When considering real-world applications, both the z-score and min-max scaling methods returned efficient results. Choosing between z-score and min-max scaling for data normalization often depends on the type of model receiving the input data. It is also noted that data normalization may not be required for the linear regression problem because the coefficients are calculated to determine the proper scaling. For the future work, we may present a comparative study for recognizing the most appropriate algorithms for modeling of cryptocurrency price predictor.

Acknowledgment. This research project was supported by Mahasarakham University and King Mongkut's University of Technology North Bangkok.

REFERENCES

[1] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu and R. Brooks, A brief survey of cryptocurrency systems, *The 14th Annual Conference on Privacy, Security and Trust (PST)*, Auckland, New Zealand, pp.745-752, 2016.

[2] R. Auer and D. Tercero-Lucas, *Distrust or Speculation? The Socioeconomic Drivers of U.S. Cryptocurrency Investments*, https://ssrn.com/abstract=3928829, Accessed on 30 September 2021.

[3] B. Aiden and O. Mason, Cryptocurrency and the future currency in the United States of America, *Journal of Finance and Accounting*, vol.5, no.2, pp.10-17, 2021.

[4] A. Urquhart, Price clustering in Bitcoin, *Economics Letters*, vol.159, pp.145-148, 2017.

[5] S. Corbet, B. Lucey and L. Yarovaya, Datestamping the Bitcoin and Ethereum bubbles, *Finance Research Letters*, vol.26, pp.81-88, 2018.

[6] S. McNally, J. Roche and S. Caton, Predicting the price of Bitcoin using machine learning, *The 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Cambridge, United Kingdom, pp.339-343, 2018.

[7] E. Akyildirim, A. Goncu and A. Sensoy, Prediction of cryptocurrency returns using machine learning, *Annals of Operations Research*, vol.297, pp.3-36, 2021.

[8] M. Poonggodi, A. Sharma, V. Vijayakumar, V. Bhrdwaj, A. P. Sharma, R. Iqbal and R. Kumar, Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system, *Computers & Electrical Engineering*, vol.81, pp.1-12, 2020.

[9] M. M. Patel, S. Tanwar, R. Gupta and N. Kumar, A deep learning-based cryptocurrency price prediction scheme for financial institutions, *Journal of Information Security and Applications*, vol.55, pp.1-12, 2020.

[10] T. Zoumpekas, E. Houstis and M. Vavalis, ETH analysis and predictions utilizing deep learning, *Expert Systems with Applications*, vol.162, pp.1-14, 2020.

[11] H. Fatah, R. A. Anggraini, D. Supriadi, M. W. Pertiwi, A. I. Warnilah and N. Ichsan, Data mining for cryptocurrencies price prediction, *Journal of Physics: Conference Series*, vol.1641, pp.1-6, 2020.

[12] L. A. Kurgan and P. Musilek, A survey of knowledge discovery and data mining process models, *The Knowledge Engineering Review*, vol.21, no.1, pp.1-24, 2006.

[13] S. Zhang, C. Zhang and Q. Yang, Data preparation for data mining, *Applied Artificial Intelligence*, vol.17, nos.5-6, pp.375-381, 2003.

[14] S. García, J. Luengo and F. Herrera, *Data Preprocessing in Data Mining*, Springer, Cham, 2015.

[15] D. Singh and B. Singh, Investigating the impact of data normalization on classification performance, *Applied Soft Computing*, vol.97(Part B), pp.1-23, 2020.

[16] N. M. Nawi, W. H. Atomi and M. Z. Rehman, The effect of data pre-processing on optimized training of artificial neural networks, *Procedia Technology*, vol.11, pp.32-39, 2013.

[17] S. G. K. Patro and K. K. Sahu, Normalization: A preprocessing stage, *International Advanced Research Journal in Science, Engineering and Technology*, vol.2, no.3, pp.20-22, 2015.

[18] V. S. Durga and T. Jeyaprakash, An effective data normalization strategy for academic datasets using log values, *International Conference on Communication and Electronics Systems (ICCES)*, NJ, USA, pp.610-612, 2019.

[19] L. Zhao, S. Zhou and Y. Jia, Adaptation feature norm method based on L2-normalization and scaling parameter, *International Journal of Innovative Computing, Information and Control*, vol.17, no.5, pp.1501-1511, 2021.

[20] M. Awad and R. Khanna, Support vector regression, *Efficient Learning Machines*, pp.67-80, 2015.

[21] C. H. Wu, C. C. Lu, Y. F. Ma and R. S. Lu, A new forecasting framework for Bitcoin price with LSTM, *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, Singapore, pp.168-175, 2018.

[22] C. J. Willmott and K. Matsuura, Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, *Climate Research*, vol.30, no.1, pp.79-82, 2005.