Data & Knowledge Engineering 139 (2022) 102009

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Forecasting cryptocurrency prices using Recurrent Neural Network and Long Short-term Memory

I. Nasirtafreshi[1]

Department of Artificial Intelligence, Faculty of Engineering, Islamic Azad University, Ghods Branch, Tehran, Iran

ARTICLE INFO

Keywords:
Cryptocurrency
Recurrent Neural Network
Long Short-term Memory
Deep learning
Forecasting prices
Time series data

ABSTRACT

The rapid development of cryptocurrencies over the past decade is one of the most controversial and ambiguous innovations in the modern global economy. Numerous and unpredictable fluctuations in cryptocurrencies rates, as well as the lack of intelligent and proper management of transactions of this type of currency in most developing countries and users of this type of currency, has led to increased risk and distrust of these roses in investors. Capitalists and investors prefer to invest in programs which have the least risk, the most profit and the least time to achieve the main profit. Therefore, the issue of developing appropriate methods and models for predicting the price of cryptographic products is essential both for the scientific community and for financial analysts, investors and traders. In this research, a new deep learning model is used to predict the price of cryptocurrencies. The proposed model uses a Recurrent Neural Networks (RNN) algorithm based on Long Short-Term Memory (LSTM) method to predict the price. In the presented results of the simulation of the proposed method, factors such as the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), R-Squared (R2) were compared with other similar methods. Finally, the superiority of the proposed method over other methods was proven.

1. Introduction

In recent years, the rapid growth of cryptocurrencies for the world's economic markets has been understandable. Their market has evolved irregularly and at an unprecedented rate in its short lifespan. Cryptocurrency is a digital payment held by a network of computers that uses encryption to authenticate transactions. Depending on how investors expect and how they are structured, some cryptocurrencies may also be considered securities.

Since the introduction of the most popular cryptocurrency, Bitcoin, in January 2009, more than 550 cryptocurrencies have been developed, most of which have had little success [1,2]. When Bitcoin was created, it was decided that the global electronic currency would pass through the world in a matter of minutes. This feature made Bitcoin not only a coin but also a valuable saving and a network of payments. By September 2019, the market value of cryptocurrencies had reached $ 300 billion, and bitcoin alone accounted for nearly $ 200 billion [3]. In addition, more than 2000 types of cryptocurrencies have been launched and are available for public trading [3].

One of the most important and perhaps the main factors in investing in this type of business markets is the accurate forecast of digital currency prices, which can be achieved by analyzing the profits and losses of digital currencies in the world economic markets. Proper forecasting leads to the provision of helpful information to activists in this large economic field for accurate and timely decisions which can prevent the dissemination of incorrect information by profiteers and fraudsters.

E-mail address: nasirtafreshi@gmail.com.

[1] Senior Researcher of Computer Science

https://doi.org/10.1016/j.datak.2022.102009
Received 29 October 2021; Accepted 16 March 2022
Available online 23 March 2022

The success of using machine learning techniques to predict stock markets shows that these methods can be very effective and efficient in predicting the price of cryptocurrencies [4–10]. However, the use of machine learning algorithms in the cryptocurrency market so far to analyze the price of bitcoin, using RF[2] [11], BNN[3] [12], LSTM[4] [13] And other algorithms are limited [14,15]. These studies predicted varying degrees of bitcoin price fluctuations and showed that the best results were obtained with neural network-based algorithms. In addition, the fantastic results obtained from the application of deep learning techniques in predicting the purchase price and maintenance of 12 cryptocurrencies in one year indicate the success of these methods [16,17].

In this article, the RNN[5] deep learning algorithm based on the LSTM method is used to predict the price of cryptocurrencies. In the continuation of this article, in Section two, previous studies on the price forecast of cryptocurrencies are presented, and then in section third, the proposed model is presented. Experimental results are presented in Section fourth, and finally, conclusions and future work in Section fifth.

2. Review of related studies

This Section reviews several recent studies related to cryptocurrency price predictions conducted by machine learning techniques.

Z. Jiang and J. Liang (2017) state in their paper that cryptocurrencies can be a suitable and decentralized electronic alternative to paper money issued by any country. This paper presents a model-less CNN[6] which presents the price of financial assets over a specified period of time as input, and the portfolio's weight as output. The network is trained with 0.7 percent of price data per year from cryptocurrency exchanges in a reinforcing way, which maximizes cumulative returns and is considered as a network reward function. The experiment had a yield ten times longer with a trading time of 30 min at 1.8-month intervals. This network is not limited to cryptocurrencies and can be used in other financial markets [17].

Nor Azizah Hitam and Amelia Ritahani Ismail (2018) state in their paper that machine learning is one of the most important parts of artificial intelligence, which can predict future events with the help of the past experiences of an event. ANN,[7] SVM[8] and deep learning is the most well-known and widely used algorithms in the field of machine learning. In this paper, the time-series data of six cryptocurrencies are predicted using NN, SVM and deep learning, and then the outcome are compared with each other. As a result, the SVM algorithm ranked first compared to different algorithms due to its proximity to the actual result and improved accuracy of the final result [18].

Harsh Jot Singh and Abdelhakim Senhaji Hafid (2019) state in their paper that nowadays, with the exponential increase in the number of transactions and investments in cryptocurrencies around the world, users are more involved than ever with cryptocurrencies such as Bitcoin and Atrium. The first and most crucial factor in user investment is predicting the profit and loss of a transaction in one or more specific periods. In this paper, three machine learning models supervised by Naive Bayes, RF and MLP[9] were used to predict the execution time of an Atrium transaction classified into five classes based on historical data. It was observed that MLP with SoftMax output performance compared to other proposed models, due to the instability and unbalanced nature of the data set, provided more favorable results [19].

Vasily Derbentsev, Andriy Matviychuk, and Vladimir N. Soloviev (2020) state in their paper that their research addresses the challenges of short-term forecasting of time series of cryptocurrencies using machine learning techniques. Their study over 90-days period to predict the price of the three cryptocurrencies Bitcoin, Atrium, Ripple by the BART,[10] ANNs (MP,[11] MLP) and a set of classification trees and model-RF regression was performed. Analyzing the results obtained from the cryptocurrency price forecast of ML[12] models, it was found that BART and MLP models with an average error of 3.5% and RF models with an average error of 5% have more suitable dynamics for forecasting. Finally, BART and MLP models were in the first place with about 63% accuracy compared to other models [20].

Erdinc Akyildirim, Ahmet Goncu, and Ahmet Sensoy (2021) in their paper, analyzed and forecasted 12 cryptocurrencies using previous prices the minute and daily periods. They tested machine learning classification algorithms, including SVM, logistic regression, ANN, and RF. As a result, it was found that machine learning classification algorithms have a prediction accuracy of about 55%–65% over a period of time. Among these algorithms, the SVM algorithm showed the best results in terms of prediction accuracy compared to other algorithms such as logistic regression, artificial neural networks, and random forest. [21].

3. The proposed method

The proposed method in this research to forecast the price of cryptocurrencies is a combination of two algorithms RNN and LSTM (RNN-LSTM), as follows:

---

2 Random Forests (RF).
3 Bayesian Neural Network (BNN).
4 Long Short-Term Memory (LSTM).
5 Recurrent Neural Network (RNN).
6 Convolutional Neural Network (CNN).
7 Artificial Neural Network (ANN).
8 Support Vector Machines (SVM).
9 Multi-layer Perceptron (MLP).
10 Autoregressive Binary Tree Model (BART).
11 Multilayer Perceptron (MP).
12 Machine Learning (ML).

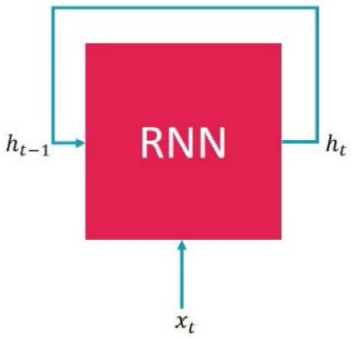

Fig. 1. Recurrent Neural Network (RNN).

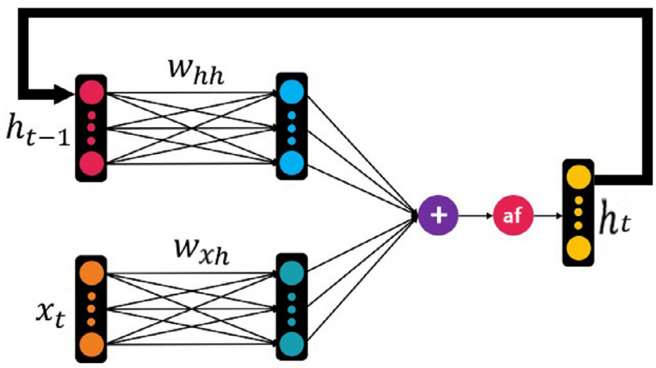

Fig. 2. Fully-connected layer (MLP layer).

3.1. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is another category of neural network, suitable for processing time-series data and sequential data. The RNN is the developed MLP neural network. RNN, LSTM, and GRU networks are three well-known and widely used networks of the return network family. The RNN consists of two inputs called memory and main. The memory input is called hidden state and is denoted by the symbol h, and the main input is represented by the symbol x. Fig. 1 shows a view of the RNN.

Each of the two inputs is connected to a Fully-Connected layer (MLP layer) (Fig. 2). The two layers w_{hh} and w_{xh} are intended for input h and x, respectively. Then h_{t-1} and x_{t-1} enter RNN as input and are multiplied by two weight matrices w_{hh} and w_{xh}. According to Eq. (2), multiplication is performed first and then addition. Finally, the output is passed through a nonlinear excitation function such as $\tan h$. The result of the excitation function is the same as h_t. Fig. 3 shows the details of the internal structure of the RNN.

Fully-Connected layer (MLP layer)

$$h_t = \tanh(w_{hh}h_{t-1} + w_{xh}x_t + b_h) \tag{1}$$

One of the most prominent problems and weaknesses of the RNN is the weakness of long-term dependency. In other words, this network cannot function appropriately in sentences, paragraphs, and all long data sequences. Therefore, LSTM neural network was proposed to solve RNN weakness. The LSTM neural network uses long-term memory, which is precisely the opposite of the RNN.

3.2. Long Short-Term Memory (LSTM)

The LSTM neural network was first introduced by Hochreiter and Schmidhuber in 1997. An LSTM neural network consists of a chain of several LSTM cells arranged in a row (such as an RNN network) (Fig. 4). The most common use of LSTM neural networks is in time-series data, such as binary classification of cryptocurrency price trends. The LSTM model can accept unlimited lengths of inputs and manage them flexibly. An LSTM unit (cell) consists of a forget gate, an input gate, and an output gate [22]. Fig. 5 shows the LSTM cell.

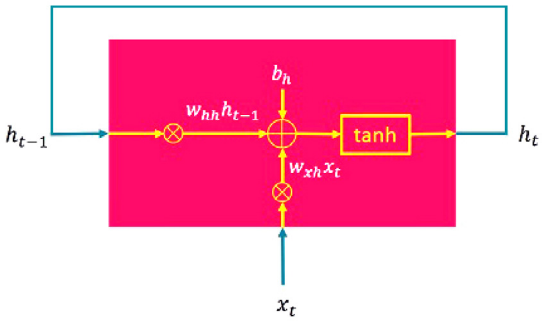

Fig. 3. Details of the internal structure of the RNN.

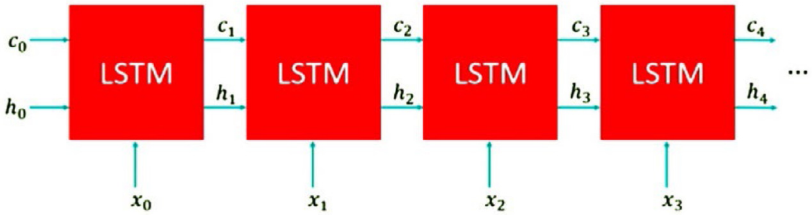

Fig. 4. STM neural network.

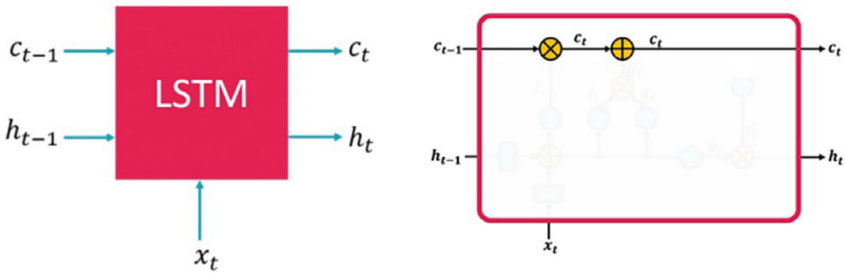

Fig. 5. LSTM cell.

3.2.1. The forget gate

The forget gate is a gate that allows the LSTM neural network to ignore or keep some of the components in C_{t-1}. The forget gate has two inputs h_{t-1} and x_t. The two inputs are combined (Fig. 6) and then pass through a sigmoid function. The Sigmoid function generates a number between 0 and 1 which multiplies element by element in the C_{t-1} vector. f_t is a vector with the same length as C_{t-1} and determines each element of C_{t-1} should be multiply by which number between 0 to 1. If any value of f_t is close to 1, it means to keep the same element in C_{t-1}, or else if it get closer to 0, the same element in C_{t-1} must be forgotten. Eq. (2) The formula describes how the forget gate inputs are combined.

Combine the forget gate inputs to each other

$$f_t = \sigma(w_{hf}h_{t-1} + w_{if}x_t + b_{hf} + b_{if}) \tag{2}$$

In Fig. 6, we have two layers in Fully-Connected weighing w_{hf} and w_{if}. These two layers are for input h_{t-1} and x_t, respectively. The hf index stands for hidden and forgotten. The if index stands for input and forget.

The two inputs f_t and C_{t-1} are combined by Eq. (3), and finally, a forget gate starts in an LSTM neural network (Fig. 7).

Combine the forget gate with LSTM neural network

$$c_t = c_{t-1} \circledast f_t \tag{3}$$

The $\circledast$ sign between the two variables in Eq. (3) refers to the multiplication of two input vectors.

The two weights w_h and w_i are the weights of the h_{t-1} and x_t paths.

In the next phase, after passing the operator $\otimes$, we enter to the operator $\oplus$ on the path $C_{t-1}C_t$ (Fig. 7). In this part, data is added to input C. It means that new information will be added to it, which means remembering or preserving the information. There are

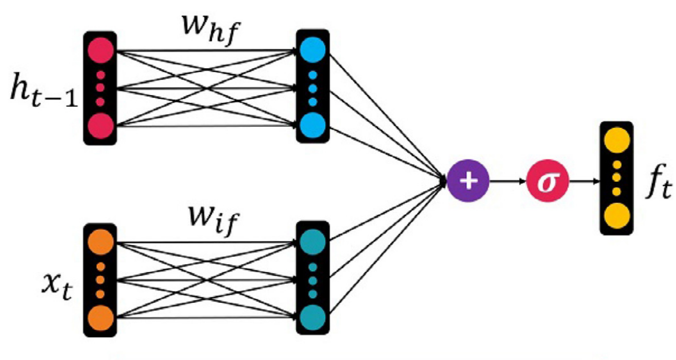

Fig. 6. Combine the forget gate inputs to each other.

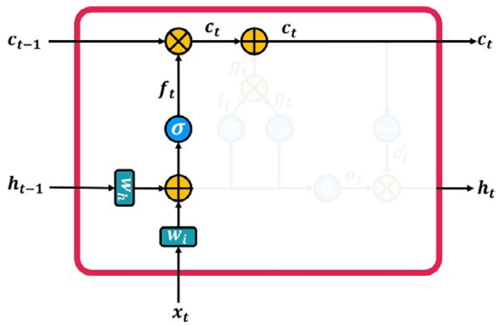

Fig. 7. A forget gate in the LSTM neural network.

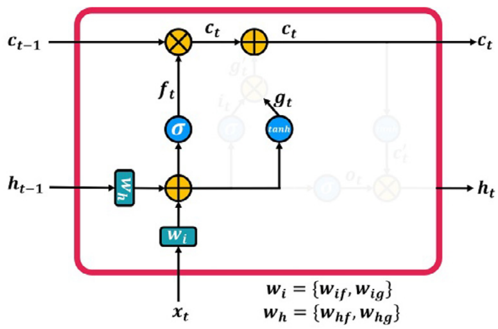

Fig. 8. A gate to store information in the LSTM neural network.

two inputs in this place, one is C_t and the other is unknown, which is the same size as C input. In fact, in the current time step (t), processes have been performed, and now the results must be left to the memory cell for storage.

A neural network is used to store information. This neural network, like the forget gate, consists of two inputs x_t and h_{t-1}. Again, these inputs must pass through two layers of Fully-Connected (w_{hg} and w_{ig}) and then be aggregated to each other. Now, this input must be passed through a hyperbolic tangent function. The output of g_t will be between −1 and 1, and this is because it may be necessary to reduce the effect of several elements or components in C. With values between −1 to 1, the impact of some features can be increased or decreased. Eq. (4) to calculate the output of this part and Fig. 8 is an example of a gate to store information.

Calculate the output to increased or decreased elements or components

$$g_t = \sigma(w_{hg}h_{t-1} + w_{ig}x_t + b_{hg} + b_{ig}) \tag{4}$$

The output of g_t, which contains current time step information, is calculated using a neural network.

A very important point is that in the g_t output, some information may not be of much value for the C_{t-1} update, so it is better to ignore it at the same time and not transfer it to long-term memory. To do this, a gate similar to the structure of the forget gate

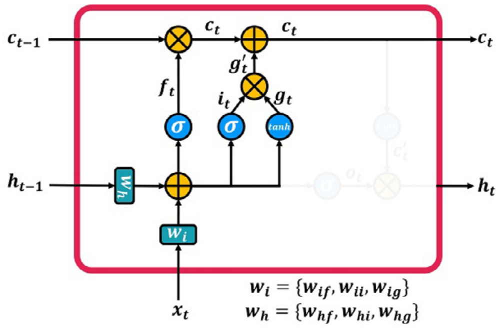

Fig. 9. Input gate in the internal structure of the LSTM neural network.

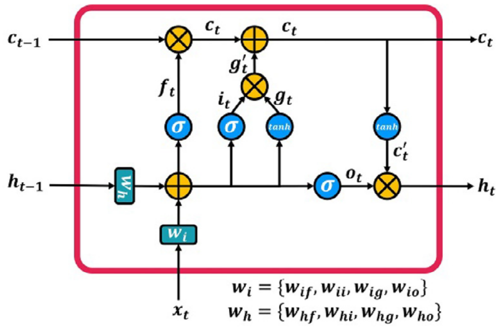

Fig. 10. Output gate in the internal structure of the LSTM neural network.

is used, which is placed on the g_t output path, it is easy to determine how much this output is worth. This gate is similar to Fig. 6, which is also called the input gate.

3.2.2. The input gate

The input gate evaluates the value of the information contained in g_t. In other words, the input gate examines the entry of new information into long-term memory. Similar to the forget gate, the values in the i_t vector may be close to zero, thus reducing the effect of g_t. Conversely, the values of the i_t vector may be close to 1, in which case g_t goes to be stored in long-term memory. The structure of this gate is exactly the same as the forget gate (Fig. 9). The input gate inserts the two inputs x_t and h_{t-1} into the two layers of the Fully-Connected, then they are added together and finally passes the sigmoid function. Eq. (5) shows how the final output (g_t) is formed, and Eq. (6) shows how the c_t update in the input gate.

Generate final output (g_t) at the input gate

$$\begin{aligned} i_t &= \sigma(w_{hi}h_{t-1} + w_{ii}x_t + b_{hi} + b_{ii}) \\ g'_t &= g_t \odot i_t \end{aligned} \tag{5}$$

Update c_t at the input gate

$$c_t = c_t + g'_t \tag{6}$$

3.2.3. The output gate

The output gate determines how much long-term memory should be transferred to the output (Fig. 10). The output gate, like the input gate, inserts the two inputs x_t and h_{t-1} into the two layers of the Fully-Connected, then they are added together and finally passes the sigmoid function. Finally, the output generated by the output gate (o_t) must be multiplied by the result of the sigmoid function to be transferred to the required size h_t output. Eq. (7) shows how the gate output (o_t) is formed, and Eq. (8) shows how the output of h_t is calculated.

How the gate output (o_t) is formed

$$o_t = \sigma(w_{ho}h_{t-1} + w_{io}x_t + b_{ho} + b_{io}) \tag{7}$$

How the output of h_t is calculated

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{8}$$

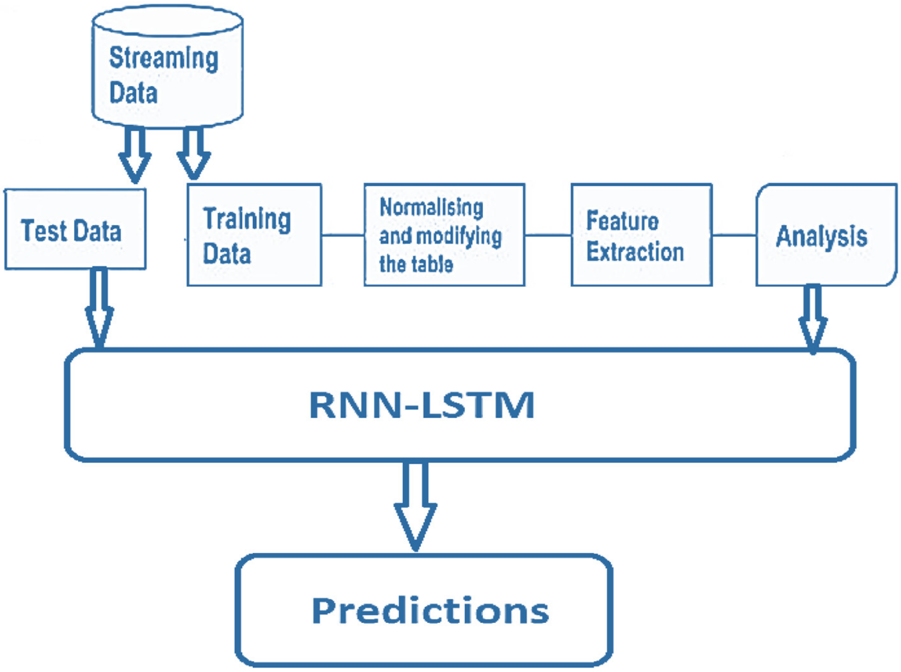

Fig. 11. The flowchart of the proposed model.

An LSTM cell, in a time-series data prediction task, can recall memories at arbitrary time intervals, thus setting up three gateways for information flow in and out of the cell. In this method, the output of each LSTM cell is used as the input of the next LSTM cell (Fig. 4). It means that the status of one LSTM cell affects the way the next cell works. The final output at the end of the sequence, represents the price trend classification labels (incremental or decremental). The LSTM neural network is more durable than the RNN neural network, which has long-term dependency problem, because LSTM neural network can control long-term memory.

The flowchart of the proposed model is shown in Fig. 11.

As shown in the proposed flowchart, in the first step, the stream data is input to the proposed system. Then the stream data set is divided into two categories: Training Data and Test Data. In the next step, the training data set is subjecting to normalization and data modification operations, which include processes such as removing null values, sorting, etc. In the next step, features that can play a more helpful role in decision-making are selected and extracted. After performing the above operations, the test data and the analyzed data are applied as input to the RNN-LSTM algorithm to predict the price. In fact, in the proposed method, a combination of RNN and LSTM algorithms is used.

Both LSTM and RNN networks use neural network algorithms to predict the price of cryptocurrencies. It means that it uses sequential data stored in memory that has already been received by the input at a specified time interval. As a result, it helps these networks to accurately predict the subsequent output (price).

As mentioned earlier, RNN has two inputs, one for current data and the other for previous data. In addition, RNN can hold some essential historical data in its memory, which is considered an essential element. The gradient in RNN is typically used to determine the change in weight relative to the change in error, but in this study, the Sigmoid function is used to achieve this goal, which produces a result between 0 and 1.

In the proposed method, first, the optimal size of the window for storing information is determined. This is done by analyzing the final price chart by the date factor for better forecasting. The price closest to the final price in the graph can be in the last three-day before the final day. Therefore, a 3-day time window length is used for RNN. In the LSTM network, two hidden layers are considered. Because in a larger time window, this network produces a better result.

After applying the data as input to the proposed model, the process of cleaning and normalizing the data is done in such a way that price and volume values vary from −1 to 1. The LSTM network used in this research is a two-way LSTM network with two hidden layers that these layers can produce better results in a larger time window. This type of network is similar to traditional LSTM models, which can improve model performance in sequence classification issues. In cases where all input sequence time steps are available, two-way LSTMs can train two sequences instead of one input sequence.

After entering the data into the proposed model, first, the process of cleaning and normalization of data is performed. Finally, its output includes close price and volume traded in the range between −1 and 1. This training data is then entered into the proposed algorithm, and finally, the result is obtained, which is the price forecast for the next two months.

4. Training data and evaluation criteria

In this research, training data and evaluation criteria have been used as follows:

4.1. Training data

In this study, the daily close price data sets of 4 cryptocurrencies BTC, Bitcoin Cash (BCH), Litecoin (LTC), and ETH, along with their fluctuations from 15 September 2016 to 5 November 2018 were received from CoinMarketCap, and the experiment was done [23].

4.2. Evaluation criteria

In this research, four standard criteria have been used to evaluate the forecast performance:

- Root Mean Square Error (RMSE).
- Mean Absolute Error (MAE).
- Mean Absolute Percentage Error (MAPE).
- R-Squared (R^2).

4.2.1. Root Mean Square Error (RMSE)

Root-Mean-Square Deviation (RMSD) or Root-Mean-Square Error (RMSE) measures the differences between values predicted and the values observed. RMSD The square root of the second instance shows the difference between the predicted values and the observed values, or the quadratic mean of these differences. These deviations are used when calculating on the estimated sample data and are considered as forecast errors. RMSD in a single measurement in the aggregation of large errors of different data points, is very accurate to predict. RMSD is an accuracy measurement tool for comparing the prediction errors of different models in a particular dataset and is not applicable to multiple datasets because it is scale dependent [24–26]. Note Equation (9):

How the output of h_t is calculated [26]

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}} \tag{9}$$

In Eq. (9), i is a variable, N is the number of data points which was not lost, x_i is the time series of actual observations, and $\hat{x}_i$ is estimated time series [26].

4.2.2. Mean Absolute Error (MAE)

Mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement [27,28]. MAE is calculated by Eq. (10):

How the output of h_t is calculated [28]

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \tag{10}$$

In Eq. (10), y_i is the prediction, x_i is the actual value, and n is the total number of data points. The mean absolute error is a common measure of forecast error in time series analysis [27–29].

4.2.3. Mean Absolute Percentage Error (MAPE)

The mean absolute percentage error (MAPE) or the mean absolute percentage deviation (MAPD) is the same measure of prediction accuracy of a forecasting method in statistics [30]. The calculation method is done according to Eq. (11):

How the output of h_t is calculated [30]

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \tag{11}$$

In Eq. (11), A_t is the actual value, F_t is the forecast value, and n is the total forecasted points [30].

4.2.4. R-Squared (R^2)

R^2 is a proportion of the variance in the dependent variable that is predictable from the independent variable(s). This coefficient is used in the discussion of statistical models in such a way that the purpose is either to predict future outputs or to test the hypothesis based on other relevant information [31–34]. The coefficient of determination will always be between 0 and 1, the number 0 indicates that the model has nothing to do with dependent and independent variables around the mean, and the number 1 indicates that the model has all kinds of response data variability around the mean. In sometimes, R^2 may be negative [31]. Note Eq. (12).

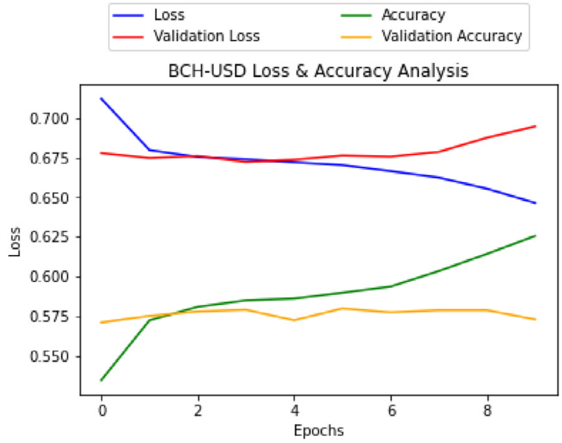

Fig. 12. The flowchart of the proposed model.

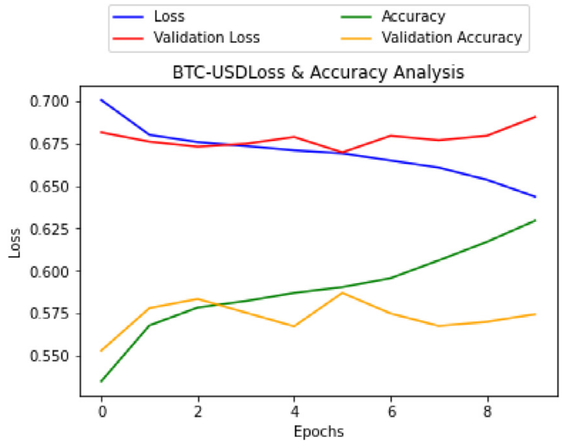

Fig. 13. The flowchart of the proposed model.

How the output of h_t is calculated [31]

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad SS_{res} = \sum_i (y_i - f_i)^2 \qquad SS_{tot} = \sum_i (y_i - \bar{y})^2$$
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{12}$$

In Eq. (12), $\bar{y}$ is the mean of the observed data, SS_{res} is the sum of squares of residuals of the data set, SS_{tot} is the total sum of squares (proportional to the variance of the data) [31].

In the best case, the modeled values exactly match the observed values, which results in $SS_{res} = 0$ and $R^2 = 1$. A baseline model, which always predicts y, will have $R^2 = 0$. Models that have worse predictions than this baseline will have a negative R^2 [31].

5. Simulation

Python 3.10.0 software version 64-bit was used to simulate this paper.

In Figs. 12 to 15, the effect of different parameters on the proposed method by factors such as the Amount of Loss, Validation Loss, Accuracy, and Validation Accuracy for BCH, BTC, ETH, and LTC are evaluated in the simulation.

To evaluate the proposed method in this study, this method was compared with the proposed method in the article [3] (CNN) listed in Table 1. In this Table, in the column titled Models all the methods that have been compared with the proposed method of this research are shown. The RMSE, MAE, MAPE, and R-squared evaluation criteria can also be seen in the following columns. Each row displays the values obtained from the implementation of the evaluation criteria on the mentioned methods. As can be seen, the proposed model has proven its superiority with the most minor prediction errors (RMSE, MAE, and MAPE) and the highest prediction accuracy (R-squared) in the ETH data set.

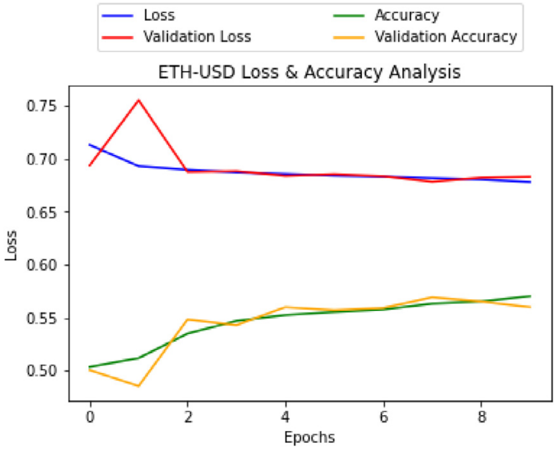

Fig. 14. The flowchart of the proposed model.

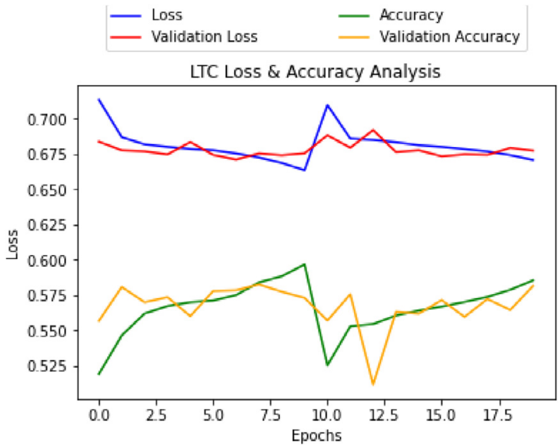

Fig. 15. The flowchart of the proposed model.

Table 1
Compare performance.

Models	RMSE	MAE	MAPE	R-squared
ARIMA	6.36e+01	5.24e+01	2.46e+01	1.05e+00
SVR	1.56e+01	1.16e+01	6.37e+00	8.76e−01
RF-Regressor	1.38e+01	9.88e+00	5.34e+00	9.03e−01
XGB-Regressor	1.72e+01	1.42e+01	8.35e+00	8.49e−01
MLP	2.06e+01	1.40e+01	7.79e+00	7.84e−01
LSTM	2.04e+01	1.63e+01	9.35e+00	7.89e−01
GRU	1.76e+01	1.35e+01	7.73e+00	8.43e−01
CNN	1.44e+01	1.02e+01	5.45e+00	8.94e−01
LSTM + CNN	1.37e+01	9.30e+00	5.17e+00	9.06e−01
GRU + CNN	1.21e+01	8.47e+00	4.74e+00	9.26e−01
Proposed method	**7.48e+00**	**5.93e+01**	**3.21e+00**	**9.43e−01**

The proposed model in this study proved its superiority over other models by obtaining the lowest RMSE at about 7.48 and the highest R-squared at about 9.43 in the cryptocurrency price forecast test of ETH. Meanwhile, the traditional ARIMA model performed the worst with an RMSE of around 6.36 and an R-squared of approximately 1.05. Among other solo models, CNN and RF-Regressor performed better than others. In particular, the prediction performance of the integrated models (LSTM + CNN and GRU + CNN) has been significantly improved, which may be due to the integration of memory and specific features of CNN. In addition, GRU + CNN has the best performance among the models. Finally, after comparing the performance of the methods in Table 1 in predicting the price of cryptocurrencies, it can be inferred that the proposed model in this study can predict the price of different cryptocurrencies relatively accurately.

6. Conclusion

In this paper, an RNN-LSTM-based model is proposed to predict the daily close price and fluctuations of cryptocurrencies. Extensive experiments were then performed using historical price data from the four cryptocurrencies (BTC, BCH, LTC, and ETH). The experimental results showed that the proposed model in this study has higher performance than other common methods due to less predictive errors (RMSE, MAE, and MAPE) and more accurate evaluation (R-squared).

To extend this research to the next phase, we intend to conduct experiments to examine the sensitivity of the parameters so that we can predict the acceleration of price increases or decreases. This means whether the price increases more or increases less.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Crypto-Currency Market Capitalizations. [Online]. http://coinmarketcap.com/.
[2] Ava Filipour, Exploration of Machine Learning Methods for Classifying Myocardial Sectors Based on Blood Flow Measurement, The Catholic University of America, 2021.
[3] Zhuorui Zhang, Hong-Ning Dai, Junhao Zhou, Subrota Kumar Mondal, Miguel Martínez García, Hao Wang, Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels, Elsevier J. (2021) 3–36, http://dx.doi.org/10.1016/j.eswa.2021.115378, Accepted 6 June 2021.
[4] David Enke, Suraphan Thawornwong, The use of data mining and neural networks for forecasting stock market returns, Expert Syst. Appl. 29 (4) (2005) 927–940.
[5] Wei Huang, Yoshiteru Nakamori, Shou-Yang Wang, Forecasting stock market movement direction with support vector machine, Comput. Oper. Res. 32 (10) (2005) 2513–2522.
[6] Phichhang Ou, Hengshan Wang, Prediction of stock market index movement by ten data mining techniques, Modern Appl. Sci. 3 (12) (2009) 28.
[7] Martin Gavrilov, Dragomir Anguelov, Piotr Indyk, Rajeev Motwani, Mining the stock market: Which measure is best? in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 487–496, http://dx.doi.org/10.1145/347090.347189.
[8] K. Senthamarai Kannan, P. Sailapathi Sekar, M. Mohamed Sathik, P. Arumugam, Financial stock market forecast using data mining techniques, in: Proceedings of the International Multiconference of Engineers and computer scientists, Vol. 1, 2010, pp. 4.
[9] Alaa F. Sheta, Sara Elsir M. Ahmed, Hossam Faris, A comparison between regression, artificial neural networks and support vector machines for predicting stock market index, (IJARAI) Int. J. Adv. Res. Artif. Intell. 4 (7) (2015).
[10] Pei-Chann Chang, Chen-Hao Liu, Chin-Yuan Fan, Jun-Lin Lin, Chih-Ming Lai, An ensemble of neural networks for stock trading decision making, in: International Conference on Intelligent Computing, Springer, 2009, pp. 1–10.
[11] Isaac Madan, Shaurya Saluja, Aojia Zhao, Automated bitcoin trading via machine learning algorithms, 2015.
[12] Huisu Jang, Jaewook Lee, An empirical study on modeling and prediction of bitcoin prices with Bayesian neural networks based on blockchain information, IEEE Access 6 (2018) 5427–5437.
[13] Sean McNally, Predicting the Price of Bitcoin using Machine Learning, IEEE, 2018, http://dx.doi.org/10.1109/PDP2018.2018.00060.
[14] Kareem Hegazy, Samuel Mumford, Comparitive automated bitcoin trading strategies, 2016.
[15] Tian Guo, Nino Antulov-Fantulin, Predicting short-term bitcoin price fluctuations from buy and sell orders, 2018, arXiv:1802.04065.
[16] A.G. Shilling, Market timing: Better than a buy-and-hold strategy, Financ. Anal. J. 48 (1992) (2018) 46–50, http://dx.doi.org/10.2469/faj.v48.n2.46.
[17] Z. Jiang, J. Liang, Cryptocurrency portfolio management with deep reinforcement learning, in: 2017 Intelligent Systems Conference (IntelliSys), 2017, pp. 905–913, http://dx.doi.org/10.1109/IntelliSys.2017.8324237.
[18] Nor Azizah Hitam, Amelia Ritahani Ismail, Comparative Performance of Machine Learning Algorithms for Cryptocurrency Forecasting, p-ISSN: 2502-4752, e-ISSN: 2502-4760.
[19] Harsh Jot Singh, Abdelhakim Senhaji Hafid, Prediction of transaction confirmation time in ethereum blockchain using machine learning, in: J. Prieto, A. Das, S. Ferretti, A. Pinto, J. Corchado (Eds.), Blockchain and Applications. BLOCKCHAIN 2019. Advances in Intelligent Systems and Computing, Vol. 1010, Springer, Cham, 2020, http://dx.doi.org/10.1007/978-3-030-23813-1_16.
[20] Vasily Derbentsev, Andriy Matviychuk, Vladimir N. Soloviev, Forecasting of cryptocurrency prices using machine learning, in: L. Pichl, C. Eom, E. Scalas, T. Kaizoji (Eds.), Advanced Studies of Financial Technologies and Cryptocurrency Markets, Springer, Singapore, 2020, http://dx.doi.org/10.1007/978-981-15-4498-9_12.
[21] Erdinc Akyildirim, Ahmet Goncu, Ahmet Sensoy, Prediction of cryptocurrency returns using machine learning, Ann. Oper. Res. 297 (2021) 3–36, http://dx.doi.org/10.1007/s10479-020-03575-y.
[22] Hui Hwang Goh, Ronghui He, Dongdong Zhang, Hui Liu, et al., Short-term wind power prediction based on preprocessing and improved secondary decomposition, J. Renew. Sustain. Energy (2021).
[23] CoinMarketCap, Coinmarketcap cryptocurrency price, 2020, URL: https://coinmarketcap.com/. Accessed: 2020-07-20.
[24] Rob J. Hyndman, Anne B. Koehler, Another look at measures of forecast accuracy, Int. J. Forecast. 22 (4) (2006) 679–688, http://dx.doi.org/10.1016/j.ijforecast.2006.03.001, CiteSeerX 10.1.1.154.9771.
[25] Saurav Agarwal, An Approach of SLA Violation Prediction and QoS Optimization using Regression Machine Learning Techniques, University of Windsor (Canada), 2020.
[26] en.wikipedia.org, Root mean square error, 2011, https://en.wikipedia.org/wiki/Root-mean-square_deviation.
[27] R. Hyndman, A. Koehler, Another look at measures of forecast accuracy, 2005.
[28] en.wikipedia.org, Mean absolute error, 2021, https://en.wikipedia.org/wiki/Mean_absolute_error.
[29] Jalal A. Al-Sudani, Hawazin K. Mustafa, Dania F. Al-Sudani, Hassan Falih, Analytical water saturation model using capacitance-resistance simulation: Clean and shaly formations, J. Nat. Gas Sci. Eng. (2020).
[30] en.wikipedia.org, Mean absolute percentage error, 2009, https://en.wikipedia.org/wiki/Mean_absolute_percentage_error.
[31] en.wikipedia.org, Coefficient of determination, 2019, https://en.wikipedia.org/wiki/Coefficient_of_determination#cite_note-1.
[32] R.G.D. Steel, J.H. Torrie, Principles and Procedures of Statistics with Special Reference to the Biological Sciences, McGraw Hill, 1960.
[33] Stanton A. Glantz, B.K. Slinker, Primer of Applied Regression and Analysis of Variance, McGraw-Hill, ISBN: 978-0-07-023407-9, 1990.
[34] N.R. Draper, H. Smith, Applied Regression Analysis. Wiley-Interscience, ISBN: 978-0-471-17082-2, 1998.

I. Nasirtafreshi: Researcher in Artificial intelligence & IoT Development Team Manager and Data Analysis. Interested in advancing software and hardware of intelligent machines that communicate, perceive and act. Experienced in Machine Learning, data manipulation and advanced analytics, software engineering and developing applied tools and models for scientific and engineering applications.

EDUCATION: Master of Science, Computer Engineering – Artificial Intelligence, ISLAMIC AZAD UNIVERSITY – Tehran – 2019.