PAPER DETAILS

TITLE: Bitcoin Cryptocurrency Price Prediction Using Long Short-Term Memory Recurrent Neural Network

AUTHORS: Ahmad Bilal WARDAK,Jawad RASHEED

PAGES: 47-53

Avrupa Bilim ve Teknoloji Dergisi
Sayı 38, S. 47-53, Ağustos 2022
© Telif hakkı EJOSAT'a aittir
Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 38, pp. 47-53, August 2022
Copyright © 2022 EJOSAT
Research Article

Bitcoin Cryptocurrency Price Prediction Using Long Short-Term Memory Recurrent Neural Network

Ahmad Bilal Wardak[1], Jawad Rasheed[2*]

[1] Istanbul Aydin University, Faculty of Engineering, Department of Software Enginering, Istanbul, Türkiye (ORCID: 0000-0002-7928-5234), ahmadwardark@stu.aydin.edu.tr

[2*] Nisantasi University, Faculty of Engineering and Architecture, Department of Software Engineering, Istanbul, Türkiye (ORCID: 0000-0003-3761-1641), jawad.rasheed@nisantasi.edu.tr

(First received 26 February 2022 and in final form 29 June 2022)

(**DOI:** 10.31590/ejosat.1079622)

ATIF/REFERENCE: Wardak, A. B., & Rasheed, J. (2022). Bitcoin Cryptocurrency Price Prediction Using Long Short-Term Memory Recurrent Neural Network. *European Journal of Science and Technology*, (38), 47-53.

Abstract

Due to its growing popularity and commercial acceptance, cryptocurrency is playing an increasingly essential role in altering the financial system. While many people are investing in cryptocurrency, the dynamic characteristics and predictability of crypto currency are still largely unknown, putting investments at risk. In this paper, we attempt to anticipate the Bitcoin price by taking into account a variety of factors that influence its value with the highest possible accuracy using (LSTM) Recurrent Neural Network. The data we use in this work includes updated daily records of many aspects of Bitcoin pricing over a five-year period. Since the cryptocurrency (Bitcoin) data is so volatile, we implement an effective pre-processing of the data in order to have a better prediction result. With this solution, we gain accuracy of 95.7% and RMSE of 0.05. Furthermore, we compare this work with other existing methods based on performance and accuracy. This comparison demonstrates that utilizing LSTM with adequate hyperparameter tweaking is one of the most efficient ways for cryptocurrency price prediction.

Keywords: Cryptocurrency, Bitcoin, Blockchain, Neural Networks, Deep Learning, RNN, LSTM.

Uzun Kısa Vadeli Bellek Tekrarlayan Sinir Ağı Kullanarak Bitcoin Kripto Para Birimi Fiyat Tahmini

Öz

Artan popülaritesi ve ticari kabulü nedeniyle, kripto para birimi finansal sistemi değiştirmede giderek daha önemli bir rol oynamaktadır. Birçok kişi kripto para birimine yatırım yaparken, kripto para biriminin dinamik özellikleri ve öngörülebilirliği hala büyük ölçüde bilinmemektedir ve bu da yatırımları riske sokmaktadır. Bu yazıda, Tekrarlayan Sinir Ağını (LSTM) kullanarak değerini mümkün olan en yüksek doğrulukla etkileyen çeşitli faktörleri dikkate alarak Bitcoin fiyatını tahmin etmeye çalışıyoruz. Bu çalışmada kullandığımız veriler, beş yıllık bir süre boyunca Bitcoin fiyatlandırmasının birçok yönünün güncellenmiş günlük kayıtlarını içermektedir. Kripto para birimi (Bitcoin) verileri çok değişken olduğundan, daha iyi bir tahmin sonucuna sahip olmak için verilerin etkili bir ön işlemesini uyguluyoruz. Bu çözümle %95.7'lik bir doğruluk ve 0.05'lik bir RMSE elde ediyoruz. Ayrıca, bu çalışmayı performans ve doğruluğa dayalı olarak mevcut diğer yöntemlerle karşılaştırıyoruz. Bu karşılaştırma, LSTM'yi yeterli hiperparametre ayarlaması ile kullanmanın, kripto para birimi fiyat tahmini için en etkili yollardan biri olduğunu göstermektedir.

Anahtar Kelimeler: Kripto para, Bitcoin, blok zinciri, Nöral Ağlar, Derin Öğrenme, RNN, LSTM.

* Corresponding Author: jawad.rasheed@nisantasi.edu.tr

1. Introduction

Neural network and Deep learning-based cryptocurrency prediction and trading is a new and emerging field of research. It has been extensively studied and used for stock price prediction and trading. Similarly, different algorithms have been used to analyze the demand for cryptocurrencies, especially Bitcoin. Cryptocurrency (Greenberg, 2011) is a digital asset that is maintained in a computerized database system and uses encryption technology to safeguard transaction records, governs coin production, and authenticates coin ownership transfers. Bitcoin (Vigna Paul, 2016) is a type of electronic cash that is the most valued in the world and the first decentralized cryptocurrency. It was established in 2008 by Satoshi Nakamoto, an unnamed individual or group of individuals. It was deployed once the currency's architecture was made public as open-source software in 2009. Bitcoin is a decentralized digital currency that may be moved from one user to another over the peer-to-peer bitcoin network without the use of an intermediary.

A database framework called blockchain (Kharpal, 2018), which is similar to an encrypted registry with bitcoin transactions, secures and keeps track of all bitcoin transactions. In a blockchain, a block corresponds to a single record. Any block in the blockchain is linked to the previous block and shapes the block's substance. By default, all data contained inside a blockchain is encrypted. With the incorporation of automated cash systems into the network, any two involved parties will now transact with each other. With the integration of electronic cash systems into the network, any two involved parties may conduct safe transactions without the intervention of a third party or device. Each exchange includes the receiver's public key, which the owner verifies using his own private key.

Figure 1. Bitcoin's price fluctuation since it was formed.

Since the features of Bitcoin are so complex, and its price fluctuates over time, there are only a few algorithms that can forecast the value of bitcoin cryptocurrencies. (Figure 1) shows how the price of Bitcoin has varied in all-time (Bitstamp, 2022). So, in order to greatly support today's investment decisions, it has become necessary to accurately forecast the value of Bitcoin, which is the most common cryptocurrency. As this problem falls under the time series prediction category, deep learning mechanisms and neural network models can be considered, especially when considering price prediction of data that is temporal in nature. This paper uses (LSTM) Recurrent Neural Network mechanism for Bitcoin price prediction.

LSTM is an RNN architecture; both are the most efficient and well-known artificial neural network methods for identifying patterns in data sequences such as numerical time series data, financial markets, and government agencies. What differentiates RNNs and LSTMs from other neural networks is that they consider time and sequences and have a temporal component. On the other hand, Recurrent networks take both current input examples and what they have perceived previously in time as their input.

Due to the highly non-linear and volatile character of financial market proceedings, it is unsurprising that neural networks have been gaining increased attention in the world of finance, particularly for time series forecasting. Because Bitcoin is a relatively new technology, there are only a few price predictions models available. In this section, we will go over prior research on bitcoin price prediction using deep learning in a nutshell. Researchers (Patel et al., 2020) offer a hybrid cryptocurrency prediction approach based on LSTM and GRU that works on only two coins, Litecoin and Monero. The results show that this approach accurately predicts prices, and the strategy may be utilized for numerous cryptocurrency price estimations. Authors in (McNally et al., 2018) developed two models for bitcoin price prediction based on RNN and LSTM and compared them to an autoregressive integrated moving average (ARIMA) model (*Th15 Week ' s Citation Classic ®*, 1989). The RNN and LSTM models outperformed the ARIMA model in the study (McNally et al., 2018).

In the study (Lahmiri & Bekiros, 2021), a deep forward neural network (DFFNN) was built and deployed for the modeling and forecasting of Bitcoin high-frequency price data. The authors of this paper assessed the effect of the robust method and the Levenberg-Marquardt algorithm on DFFNN accuracy. DFFNN trained with the Levenberg-Marquardt algorithm outperforms DFFNN trained with the Powell-Beale restarts technique and DFFNN trained with the robust algorithm, according to the simulation results. As a consequence, the robust algorithm is faster, implying that it might be effective in online training and trading. The predictability of three main cryptocurrencies (Bitcoin, Ethereum, and Litecoin) and the profit growth of investment strategies the machine learning approach proposed by (Sebastião & Godinho, 2021) with an accuracy of 80.17 percent and 91.35 percent, respectively.

Paper (Ferdiansyah et al., 2019) studies bitcoin and stock market projections, methodology, tactics, and tools from a variety of sources, including books, journals, and other publicly available sources. This research study demonstrates how to create a model prediction of the bitcoin stock market using LSTM, but the prediction output is insufficient in terms of RMSE. Scientists in (Albariqi & Winarko, 2020) evaluated the efficacy of MLP and RNN models for forecasting bitcoin price fluctuations. Long-term price prediction outperforms short-term price prediction in both MLP and RNN. The top-performing model in this experiment was a Multilayer Perceptron with a time window of 3 and 200 epochs, an accuracy of 81.3 percent, a precision of 81 percent, and a recall of 94.7 percent.

To support the development of Intelligent Systems (Alkaya, 2013) introduced an optimization technique for the number of processing features of hidden layers to predict a stock price measurement using Evolutionary Artificial Neural Networks. The researchers in this paper demonstrated that when the number of neurons employed in hidden layers is improved, EANN performs better with a fixed neural network design and optimized parameters. (Raşo, H. & Demirci, 2019) provides a deep learning model feeding with statistical features computed from historical benchmark price data, and their approach outperforms and achieves high accuracy. (Rasheed et al., 2020) suggested two deep learning approaches, LSTM and 1D-CNN, to increase stock prediction accuracy using 3 independent datasets. They investigated how feature extraction using principle component analysis (PCA) influenced the accuracy rate of both 1D-CNN and LSTM. This study revealed that LSTM with PCA produced much better results on the dataset, despite the fact that the 1D-CNN model performed well in terms of computational complexity during training.

Furthermore, (Kemalbay G., 2021) reported the development of a machine learning tool based on genetic programming for the forecasting of the Istanbul Stock Exchange National market index and compared the results to traditional seasonal ARIMA (SARIMA) and ARCH models. These findings show that the GP technique outperforms standard methods in terms of predicting the XU100 index's financial time series data. However, inspired by the accurate result and best performance of mentioned research studies, for predicting Bitcoin cryptocurrencies price, this paper will also be using the RNN (LSTM) method but with various data pre-processing methods, using different hyperparameters and layers, using a real-time bitcoin cryptocurrency dataset. And also, we provide in-depth comparisons between the existed mentioned models and the proposed model.

This method not only process single data points, but also entire sequences of data. After training, the model showed the final results of 95.7 accuracies and 0.05 RMSE. Real-time cryptocurrency data (CryptoCompare, n.d.) has been used for training and testing the developed model. Also, we compared our proposed work with existing works based on accuracy, performance, and methods. The rest of the paper is organized as follows: Section 2 explains the material and methodology used in this study, whereas Section 3 presents the experiments, analysis, and performance of the work. Section 4 contains the conclusions.

2. Material and Method

The purpose of this paper is to develop a model that takes a sequence of time series data as input, processes it, trains the recurrent neural network, and predicts the future price for the Bitcoin cryptocurrency. The complete process is divided into four major sections: Getting real-time cryptocurrency data, preparing the data for training and testing, creating and training an (LSTM) Recurrent Neural Network model for predicting the price of a cryptocurrency, and predicting the prices using the trained model. We utilized the deep learning framework TensorFlow (Kanagachidambaresan, 2021) to address the bitcoin price prediction challenge. A real-time dataset from (CryptoCompare, n.d.) was used for training and testing. The created approach can forecast prices based on the last 30 days.

2.1. Data Preparation

Data in the LSTM model requires to be in the form of sequences of Xs and ys. Where in this model X represents the last 30 days' prices and y represents the 31st-day (next day) prices. Because the LSTM algorithm is based on neural networks, standardizing or normalizing the data is required for a faster and more accurate fit.

2.1.1. Data Normalization

Our model is built on an RNN with LSTM layers that transforms input vectors into vectors with entries in the [0,1] range using the sigmoid activation function. Because the denominator will always be greater than the nominator, we apply normalization to scale model features. The output value will always be a number between 0 and 1.

2.1.2. Data Transformation

We make an assumption for the RNN in terms of the number of time periods the model will need to learn about the present time period. Given that 30 days, the time span for our RNN model to learn from Bitcoin values in the past appears to be an acceptable assumption. As a result, per price, the X train will contain the last 30 days' Bitcoin prices, while the Y train will be the next day’s Bitcoin price.

2.1.3. Data Reshaping

We add a new dimension to the model to allow for the use of additional indicators to assist estimate the pricing. We convert the data from 2D to 3D in this method. The input that LSTM expects is in 3D format. The shape of our training data is (1571, 30, 1), which is in the form of (number of samples, time steps, number of features). This means that in the training data, we have 1571 examples to learn from, with each example looking back 30-steps in time, such as what the stock price was yesterday, the day before yesterday, and so on until the last 30 days. The second number (30) is called time steps. The last number (1) represents the number of features. Here we are using just one column ‘Closing Price’ which is why we set it to 1.

2.2. Prediction Model

Since the goal is to anticipate stock values, which is a continuous output value, predicting cryptocurrency prices is a regression problem rather than a classification one. The Sequential module is used to generate a Neural Network object with sequential layers for this predictor. We next add LSTM layers, and because forecasting the price of a financial product is difficult, we choose a model with high dimensionality that can catch both upward and downward trends in the stock price, thus we employ a large number of LSTM units per LSTM layer. For regularization, we additionally include a Dropout layer, which lets us to ignore a percentage of the neurons in the LSTM layers. Finally, we add an output layer with the Dense module. Whereas, for the compiling of the model we used Adam optimizer using mean squared error loss value with the experimentally 0.001 value of learning rate and a batch size of 90.

This project is developed using the Python v3 programming language in a Windows 10 environment on a machine with an Intel(R) Core (TM) i7 processor running at 2.90GHz. In this project, the PyCharm Integrated Development Environment (IDE) is used to write the code. PyCharm is a Python programming language integrated development environment (IDE). It was created by JetBrains (*JetBrains Strikes Python Developers with PyCharm 1.0 IDE*, n.d.), a Czech firm. It provides code analysis, a graphical debugger, an integrated unit tester, version control system (VCSys) integration, and compatibility for web development with Django and data science with Anaconda (Haagsman, 2019). PyCharm is cross-platform, with versions available for Windows, Mac OS, and Linux.

2.3. Deep Neural Network Architecture

Convolutional neural networks (CNNs), unsupervised pre-trained networks (UPNs), and recurrent neural networks (RNNs) are the three major groups of deep neural network designs (A. Canziani, Adam Paszke, 2016). The architectural overview explains how to put these networks into operation. RNNS are the state-of-the-art method for sequential data, particularly time-series data, among several DNN designs (see Figure 2).

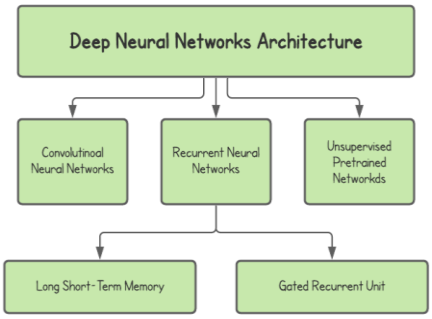

Figure 2. Deep neural networks architectures at a glance.

The estimated error is transmitted back through the network to update the weight in RNN, and information goes from the input layers to the output layer. Hidden layers of RNN, unlike UPNs, ANNs, and CNN's, not only provide output but also feed themselves. RNN (Singh, 2017) works by converting the specified input features into machine-readable vectors. The system then goes through each of these vectors one by one, moving from the beginning to the last in sequential order. During processing, the system sends the information to the next phase of this sequence via a hidden state (memory state). When the hidden state has gathered all of the system's existing information from previous stages, it is ready to move on to the next step and combine it with the current step's input to produce a new information vector. The architecture of recurrent neural networks is shown in Figure 3, where "i" is the input and "o" is the output at time step "t." The hidden state at time step "t" is "h." It's the network's memory, calculated from the previous hidden state and the current step's input. Weights for input and previous state value input are denoted by the letter's "W" and "U."

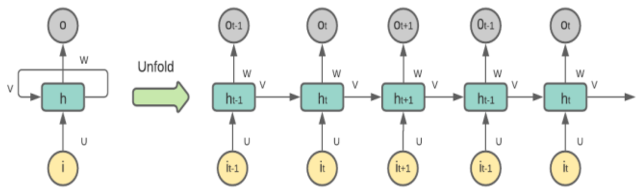

Figure 3. RNN architecture

During the optimization phase of training an RNN model, the weights are randomly chosen values in the first epoch, and if these chosen values are very small, multiplying them with the same recurrent weight multiple times minimizes the gradient until it disappears. Then, when the gradient decreases, updating weight becomes more complex, resulting in a slower optimization process. Furthermore, because all weights are connected, one incorrectly updated weight affects the calculation of the remaining weights and causes them to be wrong as well. This problem is known as the Vanishing Gradient Problem in RNNs, and LSTMs (Sepp Hochreiter, 1997) are used to solve it. The set of operations performed on the provided information and the input value in that precise step distinguishes the RNN from the LSTM.

The information in LSTMs is sent through the following gates: Forget gate, which determines whether or not information should be discarded. The input gate is used to update the state of the cell. The output gate decides what the next hidden state should be, while the cell state updates the cell state to new values that the network finds important.

2.4. Data Analyzation

To train our pricing model, we use real-time cryptocurrency price data in the period of January 2016 – to the current date, a total of 2001 days of financial data from (CryptoCompare, n.d.) which are shown in Figure 4. During this time period, the lowest close price was 448.00 USD in January 2016, and the maximum price was 38,373.90 USD on February 18, 2022, which was much greater than the lowest price.

Figure 4. Bitcoin (USD) daily close prices from January 2016 to February 2022

The dataset contains 8 features, 5 of them counted as main features which are depicted in Figure 5. The closure price of a currency is the price at which the market closes for a particular day. The highest currency price for the day is known as the High Price. Low Price refers to the day's lowest currency rate. The open price for that day's currency is the market open price. The volume of currency traded on a given day is referred to as volume. To evaluate our model, we use Bitcoin price data in USD and the target feature (value) is the close price. The created model predicts the Bitcoin close prices based on the last 30 days.

time	high	low	open	volumefrom	volumeto	close	conversionType	conversionSymbol
2016-09-01	704.97	569.37	573.88	25872.20	1.486016e+07	571.99	direct	
2016-09-02	577.60	569.30	571.99	26612.15	1.531576e+07	575.29	direct	
2016-09-03	604.97	572.64	575.29	35699.73	2.105551e+07	598.84	direct	
2016-09-04	615.20	590.78	598.84	30690.03	1.864601e+07	609.55	direct	
2016-09-05	610.61	598.77	609.55	25286.46	1.536983e+07	605.76	direct	
...	...	...	...	...	...	...	...	...
2022-02-18	40980.65	39496.70	40545.09	32514.79	1.308788e+09	39994.66	direct	
2022-02-19	40464.62	39664.71	39994.66	11674.81	4.677441e+08	40107.12	direct	
2022-02-20	40142.98	38019.97	40107.12	23028.66	8.907071e+08	38397.25	direct	
2022-02-21	39495.33	36862.52	38397.25	47549.32	1.819939e+09	37037.39	direct	
2022-02-22	37875.98	36371.68	37037.39	21929.46	8.118421e+08	37607.30	direct	

2001 rows × 8 columns

Figure 5. Bitcoin (USD) dataset sample with features and prices

2.5. Model Training

The initial dataset was partitioned into train and test sets with ratios of 80 and 20, respectively, to train the model. Following successful training, the accuracy was calculated in 100 iterations using all of the prices from the test dataset. Figure 6 shows that the model loss decreases as the number of training iterations increases. The result shows that the model has a loss of 0.00005 and an accuracy of 95.7 percent.

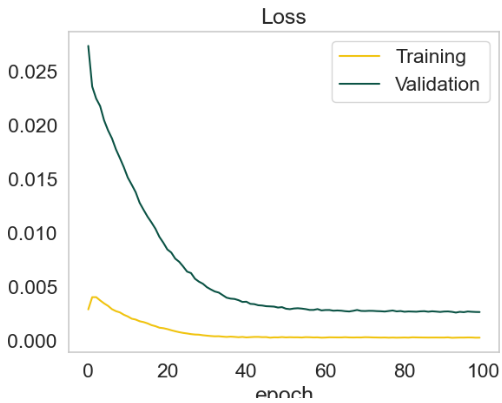

Figure 6. Training and validation loss decreases with each iteration.

At the training stage, the network processed all prices from the training dataset per each iteration, we added three callbacks, which are EarlyStopping, ModelCheckpoint, and TensorBoard callbacks. The EarlyStopping callback is used to avoid overfitting during data training. Using this callback, the training stops when there is no improvement in the monitored metric. The validation loss value would be the statistic to be tracked in this case. At the end of each epoch, the training loop will check to see if the loss is still decreasing; if it isn't, stop training is set to True, and the training will be terminated. We utilized the ModelCheckpoint callback to save the best model or weights (in a checkpoint file) at regular intervals so that the model or weights could be loaded later to continue training from the saved state and utilizing the best model. The TensorBoard callback is used to log TensorBoard events, which allow TensorBoard visualizations. TensorBoard is a visualization tool that comes standard with Tensor Flow.

3. Results and Discussion

As this paper emphasizes a solution to time-series data price prediction, it is necessary to evaluate processing and performance. To evaluate the model accuracy, we used the real-time Bitcoin dataset which contains 2001 days prices. The prices predicated by the proposed method in this paper are shown in Figure 7. The results show that the applied method gives satisfying prediction results. When testing the developed model, we used 20% of the dataset and 80% for training the model. For evaluating the model, we have applied different values of hyperparameters; We added 0.01 value for learning rate and also, we added a dropout layer with an experimental value of 0.5, which had a big impact in increasing the accuracy of the model and also gets rid of overfitting, with experimenting different batch size from 120, 90, 60 we have chosen 120 which was the most proper one. we gained 95.7% of accuracy with the validation loss of 0.00065 and 0.05 RMSE. Table 1 shows the values of the hyperparameters for the proposed model.

Figure 7. Original vs predicted close price of the bitcoin cryptocurrency.

Table 1. Proposed model hyperparameter tunning

	Proposed Model (LSTM)
Accuracy	95.7%
Validation loss	0.00065
Learning Rate	0.001
Learning Algorithm	Adam
Batch Size	90
Hidden State Size	50
Dropout	0.5
Window size	30
RMSE	0.05

To compare the proposed model with other developed models we have studied the (Albariqi & Winarko, 2020) and (McNally et al., 2018) papers. The paper (Albariqi & Winarko, 2020) discusses the Multilayer Perceptron (MLP) and Recurrent Neural Networks (RNN) models for predicting short-term and long-term Bitcoin price changes. The data utilized are from the Bitcoin blockchain from August 2010 to October 2017, with a 2-

day period and a total of 1300 data. The models anticipate both short-term and long-term price changes spanning from two to sixty days. The results show that long-term prediction beats short-term prediction, with Multilayer Perceptrons having a higher accuracy when forecasting the next 60-day change in prices and Recurrent Neural Networks getting the greatest accuracy when forecasting the next 56-day price change. With an accuracy of 81.3 percent, Multilayer Perceptron outperforms Recurrent Neural Networks. According to the findings of this experiment, the optimal learning rate for both MLP and RNN is 0.01, with validation losses of 0.58 and 0.62, respectively as shown in Table 2.

Table 2. Paper (Albariqi & Winarko, 2020) *models' comparison*

	Multilayer Perceptron	**Recurrent Neural Networks**
Short term highest accuracy	70.04%	67.56%
Long term highest accuracy	81.3%	77.3%
Validation loss	0.58	0.62
Learning Rate	0.01	0.01
Learning Algorithm	Adam	Adam
Batch Size	128	128
Hidden State Size	Obtained using formula (42)	20
Dropout	0.5	0.5
Window size	3	3

The CRISP data mining approach is used in paper (McNally et al., 2018). The Bitcoin dataset was utilized for this suggested technique, which spans from August 19th, 2013 to July 19th, 2016. This study created two models for bitcoin price prediction based on RNN and LSTM and compared them to an autoregressive integrated moving average (ARIMA) model. As a consequence, the LSTM had the best accuracy, while the RNN had the lowest RMSE (see Table 3). In terms of accuracy and RMSE, the ARIMA forecast did not perform well.

Table 3. Paper (McNally et al., 2018) *models' comparison*

	LSTM	**RNN**	**ARIMA**
Accuracy	52.78%	50.25%	50.05%
RMSE	6.87%	5.45%	53.74%
Learning Algorithm	RMSprop	RMSprop	-
Hidden State Size	20	20	-
Dropout	0.5	0.5	-
Window size	100	24	-

Table 4 shows the resulting accuracy of the proposed method compared to all methods in (McNally et al., 2018) and (Albariqi & Winarko, 2020). It shows that one of the most efficient methods for cryptocurrency price prediction is using LSTM with proper hyperparameters tuning. The approach given in this study outperforms the other methods in terms of accuracy and RMSE.

Table 4. Comparison between the proposed method and several existing methods.

	Algorithm	**Accuracy (%)**
Proposed Method	LSTM	95%
Paper (Albariqi & Winarko, 2020)	MLP	81.3%
	RNN	77.3%
Paper (McNally et al., 2018)	ARIMA	50.05%
	LSTM	52.78%
	RNN	50.25%

4. Conclusions

We introduced a recurrent neural network architecture with long-short term memory for forecasting Bitcoin cryptocurrency prices in this study. We evaluated the network under different parameters for tuning the model. It has been found that using the Relu activation function and appropriate dropout value, using gradient descent, a proper and experimented hidden node along with an effective learning rate value, and using the Adam learning algorithm considerably improved the overall result of prediction process, as well as it affected the performance of training time. The prediction is performed on Bitcoin real time dataset and after training we gained the accuracy of 95.7% with error loss of 0.00065 and 0.05 of RMSE which is quite satisfying for a time series data prediction. On the basis of performance and accuracy, we compared this study to other existing methods such as RNN, MLP, ARIMA, and other LSTM models. This comparison revealed that using LSTM with proper hyperparameter tweaking is one of the most effective methods for predicting bitcoin prices.

5. References

A. Canziani, Adam Paszke, E. C. (2016). An Analysis of Deep Neural Network Models for Practical Applications. *ArXiv, abs/1605.0*.

Albariqi, R., & Winarko, E. (2020). Prediction of Bitcoin Price Change using Neural Networks. *Proceeding - ICoSTA 2020: 2020 International Conference on Smart Technology and Applications: Empowering Industrial IoT by Implementing Green Technology for Sustainable Development*, 1–4. https://doi.org/10.1109/ICoSTA48221.2020.1570610936

Alkaya, A. (2013). NEURON OPTIMIZATION OF EVOLUTIONARY ARTIFICIAL NEURAL NETWORKS FOR STOCK PRICE INDEX PREDICTION.

International Journal of Economics and Finance Studies, 5 (1), 12–21. https://dergipark.org.tr/en/pub/ijefs/issue/26160/2

Bitstamp. (2022). *Bitcoin BTC/USD*. https://www.bitstamp.net

CryptoCompare. (n.d.). *The Premium API Solution*. https://min-api.cryptocompare.com

Ferdiansyah, Othman, S. H., Zahilah Raja Md Radzi, R., Stiawan, D., Sazaki, Y., & Ependi, U. (2019). A LSTM-Method for Bitcoin Price Prediction: A Case Study Yahoo Finance Stock Market. *ICECOS 2019 - 3rd International Conference on Electrical Engineering and Computer Science, Proceeding, June*, 206–210. https://doi.org/10.1109/ICECOS47637.2019.8984499

Greenberg, A. (2011). *Crypto Currency*. https://www.forbes.com/forbes/2011/0509/technology-psilocybin-bitcoins-gavin-andresen-crypto-currency.html?sh=7906850a353e

Haagsman, E. (2019). *Collaboration with Anaconda, Inc*. PyCharm Blog.

JetBrains Strikes Python Developers with PyCharm 1.0 IDE. (n.d.). https://www.eweek.com/development/jetbrains-strikes-python-developers-with-pycharm-1.0-ide

Kanagachidambaresan, K. B. P. R. R. (2021). Introduction to Tensorflow Package. *Programming with TensorFlow*, 1–4.

Kemalbay G., B. K. O. (2021). SARIMA-ARCH versus genetic programming in stock price prediction. *Sigma Journal of Engineering and Natural Sciences, 39(2)*, 110–122.

Kharpal, A. (2018). *Everything you need to know about the blockchain*.

Lahmiri, S., & Bekiros, S. (2021). Deep Learning Forecasting in Cryptocurrency High-Frequency Trading. *Cognitive Computation*, *13*(2), 485–487. https://doi.org/10.1007/s12559-021-09841-w

McNally, S., Roche, J., & Caton, S. (2018). Predicting the Price of Bitcoin Using Machine Learning. *Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018*, 339–343. https://doi.org/10.1109/PDP2018.2018.00060

Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. *Journal of Information Security and Applications*, *55*(May), 102583. https://doi.org/10.1016/j.jisa.2020.102583

Rasheed, J., Jamil, A., Ali Hameed, A., Ilyas, M., Ozyavas, A., & Ajlouni, N. (2020). Improving Stock Prediction Accuracy Using CNN and LSTM. *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy, ICDABI 2020*. https://doi.org/10.1109/ICDABI51230.2020.9325597

Raşo, H. & Demirci, M. (2019). Predicting the Turkish Stock Market BIST 30 Index using Deep Learning. *International Journal of Engineering Research and Development*, *11 (1)*, 253–265. doi: 10.29137/umagd.425560

Sebastião, H., & Godinho, P. (2021). Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financial Innovation*, *7*(1). https://doi.org/10.1186/s40854-020-00217-x

Sepp Hochreiter, J. S. (1997). Long Short-Term Memory. *Neural Computation*, *9 (8)*, 1735–1780. https://doi.org/https://doi.org/10.1162/neco.1997.9.8.1735

Singh, V. (2017). *Basic Architecture of RNN and LSTM*.

Th15 Week 's Citation Classic ®. (1989). *1970*, 1989.

Vigna Paul, C. M. J. (2016). *The Age of Cryptocurrency: How Bitcoin and the Blockchain Are Challenging the Global Economic Order*.