Computational Economics
https://doi.org/10.1007/s10614-022-10310-1

Price Prediction of Cryptocurrency Using a Multi-Layer Gated Recurrent Unit Network with Multi Features

Gyana Ranjan Patra[1] · Mihir Narayan Mohanty[1]

Accepted: 11 August 2022

Abstract

In today's world the cryptocurrencies have taken a special place in the financial market with the aid of time series forecasting and with a market value of almost USD 1.5 Trillion as of August 2021. These financial commodities show more volatility and effect of different internal and external forces. The prediction of such cryptocurrencies is a time consuming and difficult affair. In this paper, an attempt is made with a multilayer Gated Recurrent Unit based model for price prediction of cryptocurrencies. The currencies considered in this work are Bitcoin, Ethereum and Dogecoin and are pre-processed to deal with the NaN values. The LSTM model, one variant of recurrent neural network is used initially for predication. Further, the Gated Recurrent Unit is used with single feature. However, it is observed that for multiple features with three layers of Gated Recurrent Units based model is working well with error minimization. The performance of the proposed model is compared with other two models over a 21-day forecasting window. The proposed model is found to provide better performance in terms of different parameters like mean square error, root mean square error, mean absolute error, mean absolute percentage error, p-value, and precision values than the other two models.

Keywords Crypto-currency · Prediction · Gated recurrent unit · Long short-term memory · Mean absolute percentage error

✉ Mihir Narayan Mohanty
mihir.n.mohanty@gmail.com

Gyana Ranjan Patra
gyana.patra@gmail.com

[1] Department of Electronics and Communication Engineering, Siksha 'O' Anusandhan (Deemed to Be University), Bhubaneshwar, Odisha, India

Published online: 29 August 2022

1 Introduction

The first cryptocurrency Bitcoin was introduced in 2008 (Nakamoto, 2008) which paved the path for blockchain technology and secure transactions using advanced encryption. The huge success of Bitcoin has resulted in introduction of numerous types of cryptocurrencies using a variety of technologies. The global market capitalization of cryptocurrencies is more than US$ 1.5 Trillion as on August 2021 according to *CoinMarketCap* website (Coinmarket, xxxx), one of the leading cryptocurrency information portals. Hence forecasting of cryptocurrencies have profound applications for financial investors and advisors.

Most of the cryptocurrency market is not regulated by any authority and the authenticity and security of transactions depend on secure peer-to-peer communication (Corbet et al., 2019; Zheng et al., 2018).

It has been observed that the price of cryptocurrencies tends to fluctuate more than any other financial assets (Dyhrberg, 2016a). In 2020, Bitcoin made headlines when it surpassed USD 60,000 mark. But within months the price of Bitcoin fell sharply and was nearly halved by July 2021. Similarly, another cryptocurrency called Dogecoin was in the news for rising its value by nearly 1000% in-between Dec 2020 to May 2021 period. The cryptocurrency market has been sometimes greatly influenced by external manipulations. For example, recently the price of Bitcoin took a hit and the prices of Dogecoin increased massively with tweets from Elon Musk. These type of large fluctuations in the value of the cryptocurrencies have attracted a lot of interest from both financial investors and researchers. Prediction of such price increase or decrease will certainly help the investors to decide whether to sell immediately or to keep it for future transactions if the price rises steadily.

It has been observed that the studies using machine learning methods for cryptocurrencies is increasing but is still inadequate. There are several key issues that can be reported regarding the prediction of cryptocurrencies. It can be seen that the accurate prediction of cryptocurrency prices is a challenging task. The interdependence of the multivariate crypto data can provide new insights into the forecasting of such financial commodities.

The main motivation of the study comes from the fact that cryptocurrency price time series data are multivariate in nature, very volatile and are difficult to predict. In multivariate time series prediction cases, it becomes challenging to use the dynamic dependence relationship among variables reasonably and effectively.

In this regard, the main objective of this paper is to develop a recurrent neural network-based model that can be employed to predict the cryptocurrency price with a low forecast error and higher accuracy of forecast. The RNN structures can take longer time for training, so the stress is given to design a more computationally efficient structure. The corresponding model must perform better against the state-of-the-art methods found in literature.

In this paper a multiple layer GRU based network has been used towards prediction of three cryptocurrency prices, namely, Bitcoin, Ethereum and Dogecoin. The data is suitably pre-processed and fed to the proposed model. Different performance indicators like mean square error (MSE), root mean square error

(RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), p-value, and precision values have been found for the proposed model and have been compared with two other models that use single layer LSTM and GRU networks.

The rest of the paper is organized as follows. In Sect. 2, the literature survey is provided. Section 3 deals with the problem formulation. The system model is presented in Sect. 4. The results and discussion are presented in Sect. 5. Section 6 outlines the conclusions drawn and scope of future work of this study.

2 Literature Survey

There are a few studies (Burniske & White, 2017; Dyhrberg, 2016b; Glaser et al., 2014; Hu et al., 2019) which have shown that the Bitcoin and the other cryptocurrencies can be considered as a new type of financial asset and the weak efficiencies shown by other such entities may not be applicable to the cryptocurrency market. Another study undertakes six years of Bitcoin data and has found that the market becomes shows increase in efficiency if the study period is divided into smaller units (Urquhart, 2016).

Many researchers have attempted to predict the price of cryptocurrencies using different statistical and machine learning techniques (Chen et al., 2020). Some interesting articles based on univariate autoregressive (AR) and moving average have been proposed (Siami-Namini & Namin, 2018). Similarly, the use of simple exponential smoothing (SES) and autoregressive integrated moving average (ARIMA) was carried out for extraction of seasonality in Bitcoin exchange (Kaiser, 2019). These studies demonstrated the effectiveness of ARIMA and SARIMA models in prediction of such cryptocurrencies but they were not able to model the long-term dependencies present in these type of highly volatile time series data. These statistical based methods are linear in nature, so work only for stationary time series data and hence fail to model the non-linear properties that are hidden in the complex time series data like the cryptocurrencies.

A study using Logistic Regression and Linear Discriminant Analysis for prediction of daily prices of Bitcoin has been attempted with 66% of accuracy (McNally et al., 2018). Forecasting of Bitcoin using support vector machines (SVM), random forests and generalized linear models (GLM) has been presented with 50–55% accuracy in 10 min time prediction intervals (Madan et al., 2015). A model that uses twitter trends to predict the Bitcoin price has been presented in Abraham et al. (2018). A prediction model using Bayesian Neural Networks for prediction of Bitcoin has been presented in Jang and Lee (2017) in which a Bayesian neural networks (BNNs) is shown to perform satisfactorily in prediction of Bitcoin and explaining the high volatility of such cryptocurrencies. Similarly, an application of Deep reinforcement learning was able to surpass the performance of buy and hold techniques for 12 different cryptocurrencies (Jiang & Liang, 2017a; Shilling, 1992). A work based upon ANN and SVM has also been used for prediction of daily closing prices of Bitcoin (Mallqui & Fernandes, 2018). Similarly it has been shown that the Bitcoin prices have relationship with many other currencies like US$, Euro and Nikkei index using a Generalized Autoregressive Conditional Heteroskedastic

(GARCH) model and other machine learning models (Chen et al., 2021). The prediction of cryptocurrencies like Bitcoin, Ethereum, Litecoin and Ripple using Markov chain models have been attempted successfully (Nascimento et al., 2022). A very common issue with these models is that they are not able to capture the nonlinear relationship between the features of the multivariate time series data and perform poorly in these scenarios.

Forecast of daily prices of 1681 cryptocurrencies in three different methods using Long Short-Term Memory networks (LSTM), Gradient Boosting, and Random Forests has been presented in Alessandretti et al. (2018). The prediction using Random Sampling Method (RSM) is able to outperform LSTM for prediction in Shintate and Pichl (2019). A method based upon Multi-Layer Perceptron (MLP) and LSTM was used for prediction of multivariate data for prediction of closing prices of Bitcoin, Litecoin and Ethereum (Uras et al., 2020). A similar work that shows the applications of Gated Recurrent Unit (GRU) networks which outperforms LSTM networks has also been presented in Dutta et al. (2020). A comparison of ARIMA time series model with that of LSTM has been proposed in Karakoyun and Cibikdiken (2018). The performance of RNN and LSTM was found to be superior to MLPs due to their memory capabilities in prediction of Bitcoin prices (Lukas & Kaizoji, 2017). An approach using NARX model along with genetic algorithm to predict the daily prices of bitcoin has been attempted where it showed improved performance when compared to the feed forward neural network (Han et al., 2020). There have been other implementations of neural networks in the time series prediction application like LSTM (Felizardo et al., 2019; Shin et al., 2021), RNN (Kavitha et al., 2020), GRU (Rizwan et al., 2019) and Multi-Head Self-Attention Transformer (Sridhar & Sanagavarapu, 2021) for the prediction of different types of cryptocurrencies.

From the literature survey it has been found that for an LSTM network it can be seen that extraction of the long and short-term dependencies from the time series data but the same cannot be feedback to the periodic information of the series. Also, LSTM models have issues like time consuming iterations, complex gate operations and are generally slow in response to the dynamical changes of the time series data.

The Table 1 displays the different methodologies that have been used in literature and their corresponding performances.

3 Problem Formulation

A brief description of the time series forecasting problem is presented in this section.

The time series prediction aims to find an estimation of the future samples of a time series $\mathbf{x}_{t+1:T}$ given its past $\mathbf{x}_{t_0:t}$ where $t_0 \leq t \leq T$ are time indices. This may or may not be associated with some context information given as $\mathbf{c}$. The time series prediction can be either univariate or multivariate in nature.

Table 1 Comparison of different methods used for cryptocurrency prediction present in literature

Study	Cryptocurrency	Methodology	Prediction interval	Observations
Chen et al. (2020)	Bitcoin	Quadratic discriminant analysis SVM Random Forest XG Boost LSTM	Daily and 5 min interval	Accuracy=55.1% Accuracy=65.3% Accuracy=51.0% Accuracy=48.3% Accuracy=57.0%

McNally et al. (2018)	Bitcoin	RNN
		LSTM
			Daily	Accuracy=52.8%
				Accuracy=50.2%

Madan et al. (2015)	Bitcoin	Random forest Generalized linear model	10 min	Accuracy=50–55%

Mallqui and Fernandes (2018)		SVM
		Ensemble Model (RNN+DT)
			Daily	Accuracy=59.4%
				Accuracy=62.9%

Bouri et al. (2019)	Bitcoin	ARIMA (Parametric, semiparametric estimations)	Daily	Inefficiency

Greaves and Au (2015)	Bitcoin	Logistic regression
		Support vector machine
		Neural Network
			Hourly	Accuracy=54.3%
				Accuracy=53.7%
				Accuracy=55.1%

Zoumpekas et al. (2020)	Ethereum	CNN
		LSTM
		sLSTM
		BiLSTM
		GRU
			5 min	MAE=1.97
				MAE=0.50
				MAE=3.31
				MAE=0.55
				MAE=0.53

Jiang and Liang (2017b)	12 most-volumed cryptocurrencies exchange data	Model-less convolutional Neural Network	30 min	ROI portfolio value=16.3%

Żbikowski (2015)	Bitcoin/USD exchange data	BOX-SVM
		VW-SVM
			Daily	Accuracy=10.58%
				Accuracy=33.52%

Akyildirim et al. (2021)	12 most liquid cryptocurrencies exchange data	Random Forest	15, 30 and 60 min	Accuracy=53.0%

3.1 Univariate Time Series Prediction

For the univariate time series prediction only the past values of the time series data are used for predicting the future values. This type of prediction is most applicable when the time series data is a single time-dependent variable. Mathematically

$$\hat{\mathbf{x}}_{t+1:T} = f_u(\hat{\mathrm{x}}_{t_0:t}, \mathbf{c}_i; \theta_u) \tag{1}$$

Here $\hat{\mathbf{x}}$ is the values of the estimated samples and $f_u(\cdot)$ represents the estimator function having θ_u tuneable parameters that are shared across the time series.

3.2 Multivariate Time Series Prediction

Some time series data are known as multivariate since they contain N features at a time such that

$$\mathbf{x}_{t_0:T} = \left\{\mathbf{x}_{1,t_0:T}, \mathbf{x}_{2,t_0:T}, \ldots, \mathbf{x}_{N,t_0:T}\right\} \in \mathbb{R}^{N \times T - t_0 + 1} \tag{2}$$

Multivariate methods make the prediction using all the N variates of the data as depicted by Eq. 3.

$$\hat{\mathbf{x}}_{t+1:T} = f_m(\hat{\mathrm{x}}_{t_0:t}, \mathbf{c}) \tag{3}$$

These N features may have some correlation with each other. For the cryptocurrency price prediction case the different features are opening, closing, highest, lowest values on a trading day, the volume of cryptocurrency and the amount in US$ traded in the day.

4 System Model

A description of the development of the system model which is applied to the time series forecasting is presented in this section. Figure 1 shows the general model. As a pre-processing step, the null values have been omitted and it is applied through a Min-Max normalization as shown in Eq. 3 which ensures the limit of the data variations within 0–1 range.

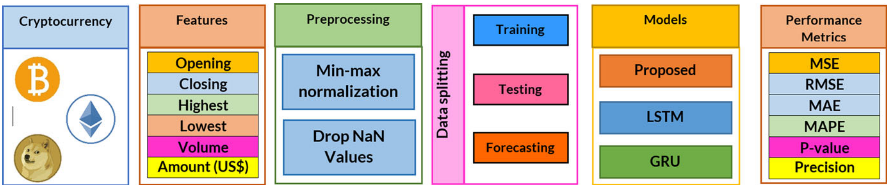


Fig. 1 System architecture for the time series forecasting problem

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

The normalized data is divided into the train and test datasets. The forecasting models are trained with the train data. The performance and the robustness of the model is evaluated using the test data.

4.1 Long Short-Term Memory

In the study of neural networks, the Recurrent Neural Network (RNN) is a special network where the output of one stage is fed to the input of the next stage. This arrangement ensures the RNN to have internal memory and thereby enabling it in extracting the patterns that have temporal characteristics present in the data. When training the RNN, several difficulties exist when the backpropagation algorithm is used. The issues like problems in convergence, the presence of vanishing gradients, etc. arise due to the cyclic structure of the RNN and therefore it is unable to deal with long-term dependencies present in time series prediction applications. The solution to this issue is present in the form of two other networks known as LSTM and GRU (Hochreiter, 1998; Jianwei et al., 2019).

The working of the LSTM has a lot of similarity to that of RNN but the repeating structure here has four neurons which provide the ability to learn the long-term dependencies. These networks are known as input, forget, and output gates which acts as a control structure for the flow of information inside the model (Cao et al., 2019; Hochreiter, 1998; Wang et al., 2021). The gates are used to generate the output in the form of a weighted sum where the weights are assigned using the backpropagation algorithm.

The structure of the LSTM and the mathematical equations explaining its working are provided in the Fig. 2 and Eqs. 5–10 respectively.

Given the input time-series input x_t and h_t as the current cell output, the gates in the LSTM network have the following equations.

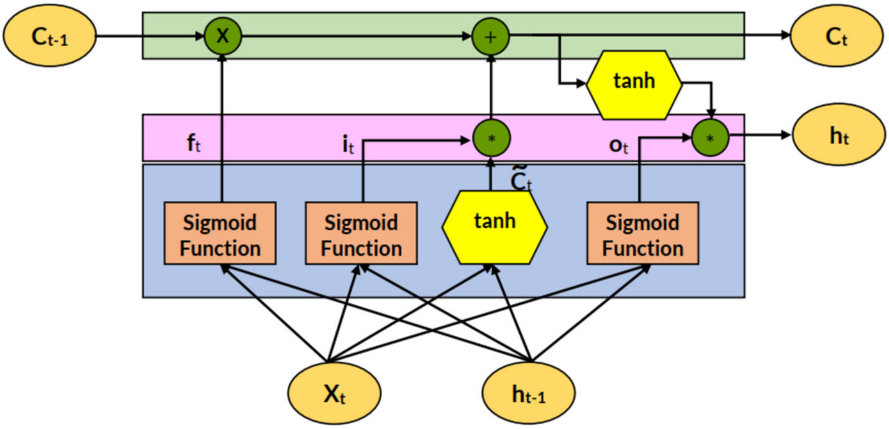


Fig. 2 Schematic representation of LSTM architecture

$$\text{Input gate}:\ i_t = \sigma(x_t V_i + h_{t-1} W_i) \tag{5}$$

$$\text{Forget gate}:\ f_t = \sigma\left(x_t V_f + h_{t-1} W_o\right) \tag{6}$$

$$\text{Output gate}:\ o_t = \sigma(x_t V_o + h_{t-1} W_o) \tag{7}$$

$$\text{Intermediate cell state}:\ \tilde{C}_t = \tanh\left(x_t V_g + h_{t-1} W_g\right) \tag{8}$$

$$\text{Cell state (next memory input)}:\ \ C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t) \tag{9}$$

$$\text{New state}:\ h_t = \tanh(C_t) * O_t \tag{10}$$

where, input is represented by x_t, current cell output is given by h_t, h_{t-1} is given as the previous cell output, C_{t-1} represents the previous cell memory, C_t represents the current cell memory, and finally W, V represent the weight matrices.

Here the input from the previous hidden state and the current state are applied to a sigmoid function to generate the forget state output f_t. The values that are close to 1 are retained only. The input gate is utilized for updating of the cell state. The data from previous hidden state and the current state are combined and passed through sigmoid and tanh functions and then these outputs are multiplied. This process ensures retaining the important information from the tanh section.

The previous cell state and the output of the forget state are multiplied and the result is added with the output of the input gate in order to generate the new cell state C_t. The output gate makes the previous hidden state and the current input state are given as the inputs to one *tanh* function, and the last obtained cell state through another *tanh* function Finally the *tanh* and the sigmoid outputs are multiplied to generate the new hidden state to be sent to the next step. In this way the gates decide the flow of information, what information to forget, what information to add and what information to carry forward.

For an LSTM network let m be the number of inputs, o be the number of outputs and n be the number of cells in the hidden layer. The number of computational complexity is given as (Rezk et al., 2020)

$$N = 4mn + 4n^2 + 3n \tag{11}$$

4.2 Gated Recurrent Unit

The GRU has a simpler structure when compared to LSTM network (Jianwei et al., 2019; Cho et al., 2014). This network has two gates, namely, update and forget gates. GRU combines the input and forget gates of the LSTM to create an update gate which is used to decide the volume of past information to be retained and transmitted to next stage.

The forget gate determines the quantity of information that need to be forgotten. This simpler arrangement enables the GRU capable of identifying the long-term dependencies as well as effectively tackling the vanishing gradient problem.

The structure of the GRU and the mathematical equations explaining its working are provided in the Fig. 3 and Eqs. 12–15 respectively.

$$\text{Update gate}: \; z_t = \sigma(U_{zh}x_t + W_{zx}h_{t-1} + b_z) \tag{12}$$

$$\text{Reset gate}: \; r_t = \sigma(U_{rh}x_t + W_{rx}o_{t-1} + b_r) \tag{13}$$

$$\text{Cell state}: \; \tilde{h}_t = \tanh(W_x x_t + U_h(r_t \odot h_{t-1}) + b) \tag{14}$$

$$\text{New state}: \; h_t = z_t \odot \tilde{h}_t + (1 - z_t) \odot h_{t-1} \tag{15}$$

where, x_t represents the input, h_t represents the output, z_t represents the update gate output, r_t represents the reset gate output, $\odot$ denotes the Hadamard product, and U, W, and b represent the feedforward weight matrix, recurrent weight matrix and bias parameters.

As shown in Fig. 3 the GRU contains three parts, namely, update and reset gates and current memory which permit it to store for a certain amount of time. The update gate multiplies a weight with the present input and the previous unit outputs, adds them and passes it through a sigmoid function. The reset gate, similar to the update gate, linearly combines present input and previous outputs with different weights and passes through another sigmoid function. These operations allow the update gate to decide the quantity of information to pass and the reset gate to decide quantity of information that need to be omitted. Then the cell state is generated by taking the Hadamard product of reset gate output with previous state output multiplying with a weight vector, multiplying the current memory state with present input, adding both and then passing it through a *tanh* function. The final output is generated by adding the result of two Hadamard product operations, one between the update state and previous output state, and the other with cell state and the difference of the update state from unity.

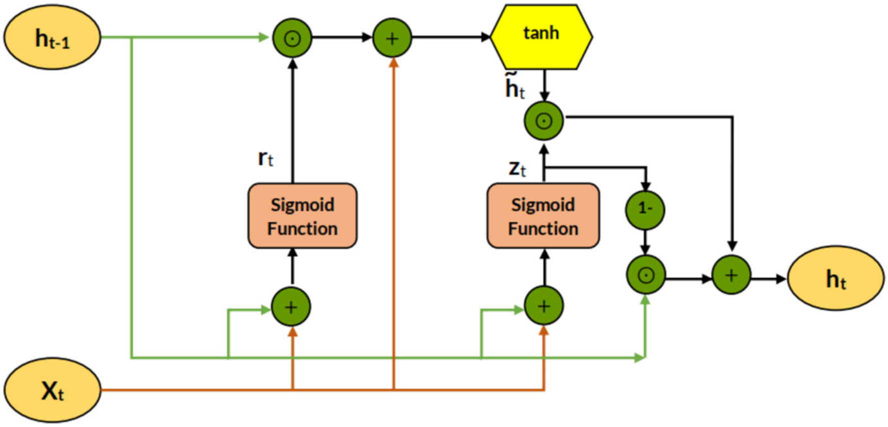


Fig. 3 Schematic representation of GRU architecture

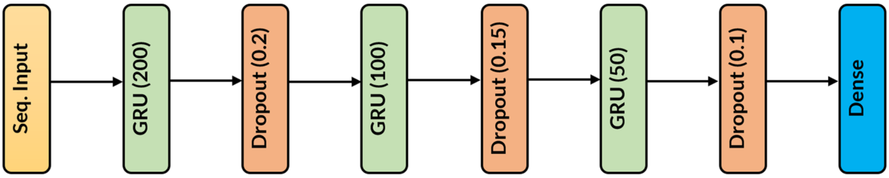


Fig. 4 Architecture of the proposed model

Considering W is an $n \times m$ matrix, U is an $n \times n$ matrix and b is an $n \times 1$ matrix (or vector) for a hidden state of size n and an input of size m, the total computational complexity is given as (Żbikowski, 2015)

$$N = 3\left(n^2 + mn + 2n\right). \tag{16}$$

By comparing the Eqs. 10 and 15 it can be observed that GRUs incur a 25% less number of multiplications and divisions compared to LSTM models.

4.3 Proposed Model

The proposed model is as shown in Fig. 4. The forecast period was chosen as a 21-day period. The model was implemented using the Tensorflow Library 2.3.2 in Python 3.8.5. The ADAM training algorithm was used to update the weights with the Mean Absolute Error as the error function. The model contains three of GRU layers with 200, 100 and 50 number of hidden units. Three dropout layers were introduced, each following one GRU layer with dropout rates of 0.2, 0.15 and 0.1 respectively, in order to prevent overfitting. The batch size and number of epochs were chosen to be 50 and 300 respectively as higher values did not result in further reduction of training and testing loss. Rest of the parameters were kept at their default as defined in the Keras package.

For comparison of performance purposes, this model was compared with two other networks containing LSTM and GRU models (Zoumpekas et al., 2020)The LSTM model contains a single layer LSTM network with 100 nodes and similarly the GRU model has a single layer with 100 nodes in contrast to Zoumpekas et al. (2020) where 50 nodes were used in both. The performance was improved when 100 nodes were taken but did not show further improvements after 100. The batch size and number of epochs were chosen to be 50 and 300 respectively and rest of the parameters were kept as default as defined in the Keras package for both of these networks.

The input data contained six features available with the data as shown in Fig. 1. The lookback period was chosen to be 1 day. The details of the all the three networks are represented in Table 2. The algorithm developed for the GRU model is listed in Algorithm 1. Similar steps are followed for the rest of the models.

Algorithm 1: The algorithm of the implementation of the GRU model

```
Function Pre-processing (Multivariate Cryptocurrency Data)
Input: Multivariate Cryptocurrency Data (Dataset)
Output: The Data is normalized and split to Train and Test

Data ≤ windows(Multivariate Cryptocurrency Data)

Function Normalise(Data_windows)
Normalised_Data = []
For i in Data:
    Normalised_Data [i] = (Data(i) – Min(Data))/(max(Data) – Min(Data))
Return Normalised_Data

row ≤ shape of Normalised Data – 21
Train_X ≤ Normalised_Data [: row, : 1]
Train_y ≤ Normalised_Data [: row, 1]          //as sequence prediction is done the same data is
                                              //divided to train_X, train_y, test_X and test_y.
Test_X ≤ Normalised_Data [row:, : 1]
Test_y ≤ Normalised_Data [row:, 1]

Function model(a, b, epoch, batch_size)
Input: a and b are the Train_X and Train_y respectively.
Output: the Trained model is obtained.
Network = sequential ()
Network.add(GRU(input, output, dropout))              //Input Layer of GRU RNN
Network.add(GRU(cells, activation, dropout))          //Hidden Layers of GRU RNN
Network.add(GRU(cells, activation, dropout))
Network.add(Dense(output, output_activation)          //Output Layer
Network.compile(loss_function, optimizer)             //Defining Optimization of the model
Network.fit(a, b, epoch, batch_size, validation)      //Training the model

Function forecast(a, b, Network)
Input: a and b are the Test_X and Test_y respectively, Network is the trained model.
Output: the predicted data for 21 days window and error parameters MSE, RMSE, MAE, MAPE
y_hat = Network.predict(a, b)
MSE = mean_squared_error (Test_y, y_hat)
RMSE  = sqrt(MSE)
MSE = mean_absolute_error (Test_y, y_hat)
MAPE = mean(abs((Test_y - y_hat)/ Test_y))*100
```

5 Results and Discussions

5.1 Data description

To obtain the data the Quandl marketplace (quandl, xxx) was used which has a rich collection of financial and economic datasets. This repository has been recently acquired by NASDAQ and has more than 300,000 users with over 10 million daily downloads (as of August 2021). Three cryptocurrencies, namely, Bitcoin, Ethereum and Dogecoin were obtained from Quandl and were associated with 6 features, which

Table 2 Parameter setup for the networks

Parameters	Proposed model	LSTM (Han et al., 2020)	GRU (Han et al., 2020)
Stages	Sequence input layer GRU layer (200) Dropout layer (0.2) GRU layer (100) Dropout layer (0.15) GRU layer (50) Dropout layer (0.1) Dense layer	Sequence input layer LSTM layer (100) Dense layer	Sequence input layer GRU layer (100) Dense layer
Input features	Open, Close, High, Low, Volume, Volume (USD)		
Training algorithm	ADAM	ADAM	ADAM
Batch size	50	50	50
Learning rate	0.0001	0.0001	0.0001
No. of Epochs	300	300	300
Error measure for training	MAE	MAE	MAE
Number of parameters	236,451	41,701	31,601

represent the opening, closing, highest, lowest values on a trading day, the volume of cryptocurrency and the amount in US$ traded in the day.

Bitcoin was created in 2009 by a person with the alias Satoshi Nakamoto. Currently it boasts a market capitalization of more than US$ 646 billion (as of August 2021). In 2020 it boasted an all-time high of US$ 63,659 and crashing to half of that within a matter of weeks.

Ethereum was founded in 2015 as a programmable blockchain which provides different services to a host of different types of transactions. This cryptocurrency is worth of more than US$ 256 billion valuation (as of August 2021).

Dogecoin was created as a sarcastic meme coin in 2013. The name is derived from its logo which is a Japanese Shiba Inu dog. Dogecoin was in the limelight in 2020 when it made a 1000% jump in its price. Currently it has a market capitalization of US$ 611 million and the rank of 43 (as of August 2021).

5.2 Data Pre-Processing

a. *Normalization* The input prices of has been normalised using the min–max normalization technique.
b. *Missing time series Data* In case of missing time series data (which appear as Not a Number (NaN)), the data point was dropped.
c. *Dividing Data into Three Sets* The data has been used to predict the price of the cryptocurrency 21 days into the future. Out of the rest of the samples 70% data is used for training and 30% data is used for testing.

The key details of the data has been presented in Table 3.

5.3 Evaluation Metrics

The following performance measure parameters defined in Eqs. 16–19 were selected for assessment of the all the three prediction models.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \widehat{y_t})^2 \tag{17}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \widehat{y_t})^2} \tag{18}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}(|y_t - \widehat{y_t}|) \tag{19}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|y_t - \widehat{y_t}|}{y_t}\right)\% \tag{20}$$

$$Accuracy = \frac{1}{n}\sum_{t=1}^{n}\left(1 - 100\frac{|y_t - \widehat{y_t}|}{y_t}\right) \tag{21}$$

$$Precision = \frac{\sum \quad Normalised \quad Absolute \quad Error}{Total \quad number \quad of \quad predicted \quad data} \tag{22}$$

For representing better results, parameter values for Eqs. 17–20 and 22 need to be small positive values and Eq. 21 should be close to 100%.

Table 3 Cryptocurrency data details

Cryptocurrency	Data starting date	Data ending date	Total no. of observations	No. of training samples	No. of testing samples	No. of prediction samples
Bitcoin	7th January 2014	3rd June 2021	2096	1453	622	21
Ethereum	9th February 2016	15th April 2020	1438	992	425	21
Dogecoin	17th September 2014	6th July 2021	2485	1725	739	21

Table 4 Performance comparison of the proposed method

Cryptocurrency	Performance measure	Proposed model	LSTM	GRU
Bitcoin	MSE	**63,307.3120**	1,469,298.6425	1,135,183.4342
	RMSE	**251.6094**	1212.1460	1065.4498
	MAE	**164.4882**	1044.8451	899.3333
	MAPE	**0.0031**	0.0237	0.0184
	Accuracy	**99.69**	97.63	98.16
Ethereum	MSE	**0.3663**	19.7456	4.7834
	RMSE	**0.6052**	4.4436	2.1871
	MAE	**0.5121**	2.1376	4.4421
	MAPE	**0.0037**	0.0147	0.0312
	Accuracy	**99.63**	98.53	96.88
Dogecoin	MSE	**9.5562 e-07**	3.1542 e-03	1.2291 e-05
	RMSE	**9.7758 e-04**	5.5214 e-02	3.5059 e-03
	MAE	**7.6192 e-04**	3.9754 e-02	3.4105 e-03
	MAPE	**2.8281 e-03**	1.7459 e-01	1.3423 e-02
	Accuracy	**99.99**	99.82	99.98

*Best results are highlighted as bold letters

Table 5 Performance comparison of ***p***-value for a forecast window of 21 days

Cryptocurrency	Proposed model	LSTM	GRU
Bitcoin	**4.9973 e-04**	0.0311	0.0278
Ethereum	**2.0641 e-04**	0.0415	0.0338
Dogecoin	**8.2553 e-04**	0.0432	0.0357

*Best results are highlighted as bold letters

Table 6 Performance comparison of ***Precision*** for a forecast window of 21 days

Cryptocurrency	Proposed model	LSTM	GRU
Bitcoin	**0.002878758**	0.029097022	0.008772406
Ethereum	**0.003272181**	0.028232053	0.025945023
Dogecoin	**0.003047922**	0.018764531	0.008767346
Average	**0.003066287**	0.025364535	0.014494925

*Best results are highlighted as bold letters

5.4 Observations

Table 4, 5 and 6 present the comparison of performance of different performance measure for the proposed model and the two other networks. The error plots as well

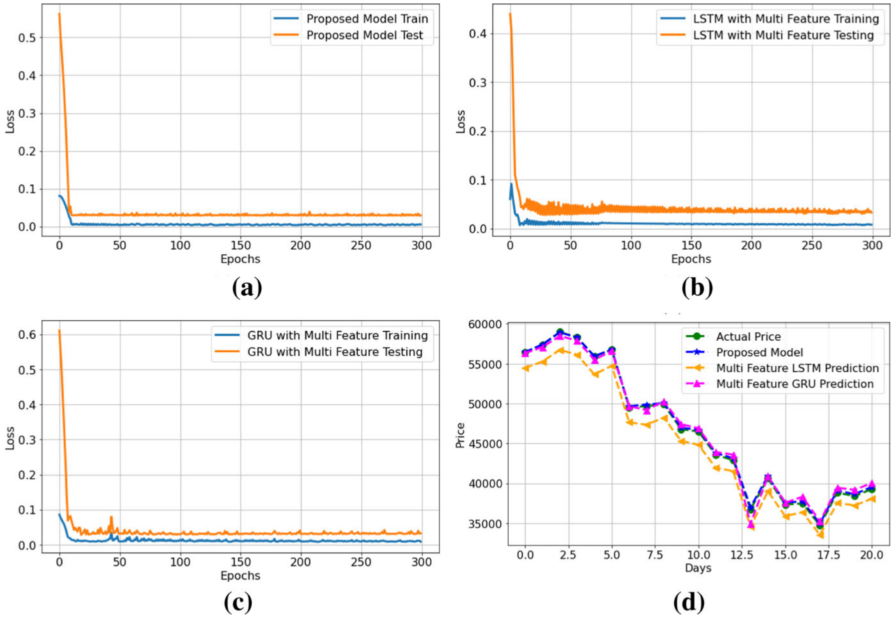


Fig. 5 Bitcoin Price prediction using **a** Proposed model **b** LSTM and **c** GRU **d** Prediction results for a 21-day window

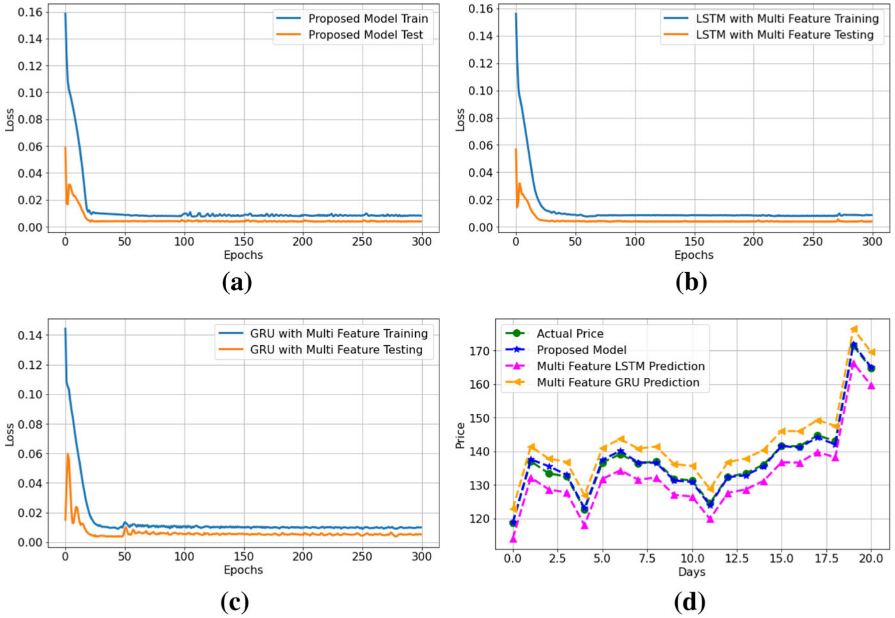
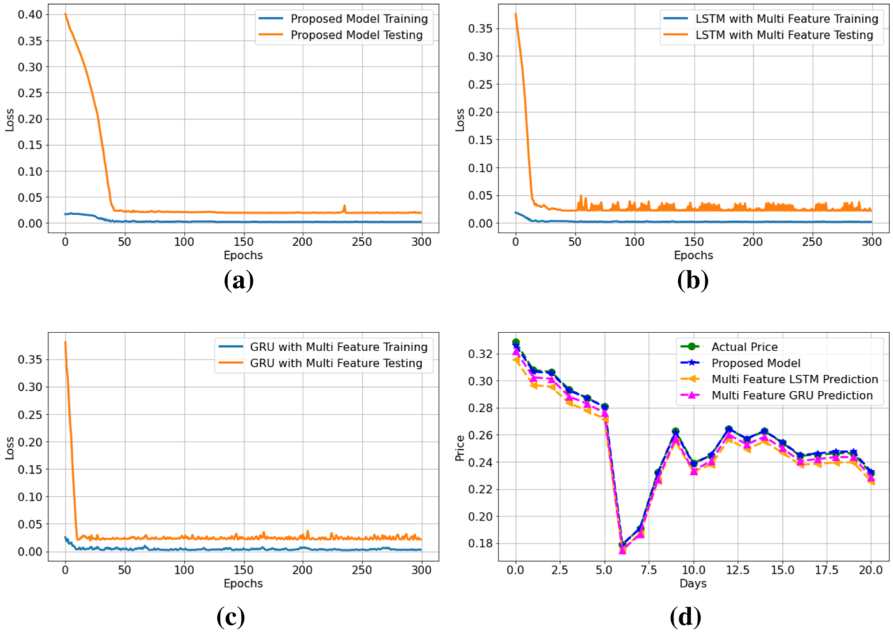


Fig. 6 Ethereum Price prediction using **a** Proposed model **b** LSTM and **c** GRU **d** Prediction results for a 21-day window



(a)

(b)

(c)

(d)

Fig. 7 Dogecoin Price prediction using **a** Proposed model **b** LSTM and **c** GRU **d** Prediction results for a 21-day window

as in the comparison of predicted prices of all the cryptocurrencies have been presented in Figs. 5, 6 and 7.

From the Table 4 it is seen that the proposed model outperforms the LSTM and the GRU models. The *MSE*, *RMSE*, *MAE* as well as *MAPE* values of the proposed model are almost one order less than that of the other two models. The *MAPE* values for the proposed model are found to be 0.0031, 0.0037 and 0.0028 for the Bitcoin, Ethereum and Dogecoin respectively. The same is found to be 0.0184, 0.0312, 1.3423 e-02 for the GRU model and 0.0237, 0.0147 and 1.7459 e-01 for the LSTM model. Omitting the outlier value of obtained in the Dogecoin case it has been observed that the proposed model has an average MSE which is 38 times less than that of LSTM performance and 15 times less than the GRU performance. Similarly in case of RMSE the proposed model on the average has approximately 23 times less than the LSTM and 4 times less than the GRU model. The proposed model is seen to have 21 times less MAE than the LSTM and 6 times less than the GRU model. Similarly the proposed model has 24 times less MAPE than the LSTM and 6 times less than the GRU model. This shows the effectiveness of the proposed model over the rest of the models used in this paper.

The best Accuracy values that have been obtained in this study is by the proposed method where it has been found to be more than 99% for all cases. The LSTM and GRU methods also perform well as their accuracy is more than 95%.

The *p*-values were found out for the models to display statistically significant results for the prediction window of 21 days and has been presented in Table 5. The

Table 7 Performance comparison of state-of-the-art models with the proposed method

Work	Cryptocurrency	Performance parameters				
		Accuracy	RMSE	MSE	MAE	MAPE
Chen et al. (2020)	Bitcoin	65.30	–	–	–	–
Zoumpekas et al. (2020)	Ethereum	97.20	–	–	–	0.28
Shin et al. (2021)	Dogecoin	95.86	31.75	1008.06	–	–
Felizardo et al. (2019)	Dogecoin	–	76.763	8990.969	64.854	–
Kavitha et al. (2020)	Dogecoin	–	95.067	9037.74	64.389	–
Rizwan et al. (2019)	Dogecoin	94.70	–	–	–	–
Sridhar and Sanagavarapu (2021)	Dogecoin	98.47	–	–	–	–
Proposed method	Bitcoin	**99.69**	**251.6094**	**63,307.3120**	**164.4882**	**0.0031**
	Ethereum	**99.63**	**0.6052**	**0.3663**	**0.5121**	**0.0037**
	Dogecoin	**99.99**	**9.7758e-4**	**9.5562e-7**	**7.6192e-4**	**2.8281e-3**

*Best results are highlighted as bold letters

p-values for Bitcoin, Ethereum and Dogecoin were found to be 4.9973 e-04, 2.0641 e-04, 8.2553 e-04 respectively. The same for LSTM and GRU models are found to be almost 50 to 200 times higher than the proposed model. Thus, it is seen that the proposed model offers more statistically significant results than the rest two models undertaken in this paper.

The precision values are tabulated in Table 6. The *Precision* values for Bitcoin, Ethereum and Dogecoin were found to be 0.002878758, 0.003272181, and 0.003047922 respectively. The average *Precision* was found to be 0.003066287, 0.025364535, and 0.014494925 for the proposed, LSTM and GRU models. Here it can be seen that the precision values achieved by the proposed model are 3 to 10 times higher compared to the other models undertaken in this paper.

The Table 7 displays the performances of some of the results from recently published literatures. It can be seen that the performance of the proposed method is superior in all three cryptocurrencies than the others.

Thus, it can be concluded that for a forecasting period of 21 days the performance of the proposed model is better than the rest of the models undertaken in this study.

6 Conclusion

The price prediction of different cryptocurrencies is a challenging task as the data shows significant amount of non-stationarity and volatility. In this paper the proposed model with a three stage GRU model has been implemented to predict the prices of Bitcoin, Ethereum and Dogecoin for a 21-day period. It has been found that the proposed model outperforms the models created using the LSTM and GRU networks. In case of MSE, the proposed model results 20–50 times less, for RMSE, 5–50 times less, for MAE 4–50 times less error and for MAPE 4–60 times less error than the LSTM and GRU models. These results establish the fact that the proposed model is a better alternative to the simple LSTM and GRU networks. The multiple GRU and the dropout layers used in the proposed model properly manage the information flow and at the same time avoid overfitting effectively.

The performance enhancement of the proposed network comes at higher computational costs. By comparing the number of tuneable parameters listed in Table 2 we see that the proposed model has 236,451 parameters whereas the LSTM and GRU models have 41,701 and 31,601parameters respectively. The proposed model takes about 5.6 times more computation compared to the LSTM and 7.5 times more than the GRU model. But the performance of the improved performance proposed network overcomes this limitation.

It can also be inferred that with availability of limited amount of data, LSTM and GRU models can perform better than many traditional machine learning models as they are better at learning the non-linear patterns and understanding the long-term dependencies present in the data. Using such in shallow learning networks does not require a lot of computational resources and it can be implemented easily.

Although the proposed model performs well for the 21-day window chosen in this paper, it struggles to show similar performance for larger forecasting windows. There are several attempts present in literature where a hybrid model is able to overcome this problem. The next aim of the authors is to develop such a model that can be applied to a longer prediction period ranging from 6 months to 1 year.

Funding This work is our own research, We have not received any financial support for the same.

Declarations

Conflict of interest The authors declare that they have no conflict of interest and the manuscript has not been submitted elsewhere.

References

Abraham, J., Higdon, D., Nelson, J., & Ibarra, J. (2018). Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review, 1*(3), 1.

Akyildirim, E., Goncu, A., & Sensoy, A. (2021). Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research, 297*, 3–36. https://doi.org/10.1007/s10479-020-03575-y

Alessandretti, L., ElBahrawy, A., Aiello, L. M., & Baronchelli, A. (2018). Anticipating cryptocurrency prices using machine learning. *Complexity, 2018*, 8983590.

Bouri, E., Gil-Alana, L. A., Gupta, R., & Roubaud, D. (2019). Modelling long memory volatility in the Bitcoin market: Evidence of persistence and structural breaks. *International Journal of Finance & Economics, 24*, 412–426.

Burniske, C., & White, A. (2017). Bitcoin: Ringing the bell for a new asset class, Working Paper.

Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications, 519*, 127–139. https://doi.org/10.1016/j.physa.2018.11.061

Chen, T. H., Chen, M. Y., & Du, G. T. (2021). The determinants of bitcoin's price: Utilization of GARCH and machine learning approaches. *Computational Economics, 57*, 267–280. https://doi.org/10.1007/s10614-020-10057-7

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics, 365*, 112395.

Cho, K., Merrienboer, B., Gulcehre, C., Bahdanau, D., Fethi, B., Holger, S., & Bengio, Y. (2014). Learning phrase representations using RNN encoder– decoder for statistical machine translation, arXiv:1406.1078.

https://www.coinmarketcap.com

Corbet, S., Lucey, B., Urquhart, A., & Yarovaya, L. (2019). Cryptocurrencies as a financial asset: A systematic analysis. *International Review of Financial Analysis, 62*, 182–199.

Dutta, A., Kumar, S., & Basu, M. (2020). A gated recurrent unit approach to bitcoin price prediction. *Journal of Risk and Financial Management, 13*, 23. https://doi.org/10.3390/jrfm13020023

Dyhrberg, A. H. (2016a). Bitcoin, gold and the dollar—A GARCH volatility analysis. *Finance Research Letters, 16*, 85–92.

Dyhrberg, A. H. (2016b). Bitcoin, gold and the dollar—A GARCH volatility analysis. *Finance Research Letters, 16*, 85–92. https://doi.org/10.1016/j.frl.2015.10.008

Felizardo, L., Oliveira, R., Del-Moral-Hernandez, E., & Cozman, F. (2019). Comparative study of Bitcoin price prediction using WaveNets, recurrent neural networks and other machine learning methods. In *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC),* pp. 1–6.

Glaser, F., Zimmermann, K., Haferkorn, M., Weber, M.C., Siering, M. (2014). Bitcoin-asset or currency? Revealing users' hidden intentions. In *Proceedings of 22nd European Conference on Information Systems,* vol. 2014, pp. 1–15.

Greaves, A., & Au, B. (2015) Using the bitcoin transaction graph to predict the price of bitcoin, stanford.edu. Available at http://snap.stanford.edu/class/cs224w. Accessed 8 Aug 2022

Han, J. B., Kim, S. H., Jang, M. H., et al. (2020). Using genetic algorithm and NARX neural network to forecast daily bitcoin price. *Computational Economics, 56*, 337–353. https://doi.org/10.1007/s10614-019-09928-5

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6*, 107–116.

Hu, A. S., Parlour, C. A., & Rajan, U. (2019). Cryptocurrencies: stylized facts on a new investible instrument. *Financial Management, 48*, 1049–1068. https://doi.org/10.1111/fima.12300

Jang, H., & Lee, J. (2017). An empirical study on modelling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. *IEEE Access, 6*, 5427–5437.

Jiang, Z., & Liang, J. (2017a). Cryptocurrency portfolio management with deep reinforcement learning. In *Proceedings of the 2017a Intelligent Systems Conference (IntelliSys)*, pp. 905–913, London.

Jiang, Z., & Liang, J. (2017b). Cryptocurrency portfolio management with deep reinforcement learning. In *Intelligent Systems Conference (IntelliSys)*, 2017b, pp. 905-913, Doi: https://doi.org/10.1109/IntelliSys.2017b.8324237.

Jianwei, E., Ye, J., & Jin, H. (2019). A novel hybrid model on the prediction of time series and its application for the gold price analysis and forecasting. *Physica A: Statistical Mechanics and its Applications, 527*, 121454. https://doi.org/10.1016/j.physa.2019.121454

Kaiser, L. (2019). Seasonality in cryptocurrencies. *Finance Research Letters, 31*, 232–238.

Karakoyun, E. S., & Cibikdiken, A. O. (2018). Comparison of ARIMA time series model and LSTM deep learning algorithm for bitcoin price forecasting. In *The 13th Multidisciplinary Academic Conference in Prague.*

Kavitha, H., Sinha, U. K., & Jain, S. S., (2020). Performance evaluation of machine learning algorithms for bitcoin price prediction. In *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, pp. 110–114.

Lukas, P., & Kaizoji, T. (2017). Volatility analysis of bitcoin price time series. *Quantitative Finance and Economics, 1*, 474–485.

Madan, I., Saluja, S., & Zhao, A. (2015). *Automated Bitcoin trading via machine learning algorithms.* Comput. Sci., Stanford Univ.

Mallqui, D., & Fernandes, R. (2018). Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. *Applied Soft Computing, 75*, 596–606. https://doi.org/10.1016/j.asoc.2018.11.038

McNally, J. R. S., & Caton, S. (2018). Predicting the price of bitcoin using machine learning. In *Proceedings of the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, pp. 339–343, Cambridge.

Nakamoto, S. (2008). Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

Nascimento, K. K. F., Santos, F. S., Jale, J. S., et al. (2022). Extracting rules via markov chains for cryptocurrencies returns forecasting. *Computational Economics*. https://doi.org/10.1007/s10614-022-10237-7

Rezk, N. M., Purnaprajna, N. M. T., & Ul-Abdin, Z. (2020). Recurrent neural networks: An embedded computing perspective. *IEEE Access, 8*, 57967–57996. https://doi.org/10.1109/ACCESS.2020.2982416

Rizwan, M., Narejo, S., & Javed, M. (2019). Bitcoin price prediction using deep learning algorithm. In *2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, pp. 1–7.

Shilling, A. G. (1992). Market timing: Better than a buy-and-hold strategy. *Financial Analysts Journal, 48*(2), 46–50.

Shin, M., Mohaisen, D., & Kim, J. (2021). Bitcoin price forecasting via ensemble-based LSTM deep learning networks. In *2021 International Conference on Information Networking (ICOIN)*, pp. 603–608.

Shintate, T., & Pichl, L. (2019). Trend prediction classification for high frequency bitcoin time series with deep learning. *Journal Risk Financial Management, 12*(1), 17.

Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. Retrived July 15, 2021, Available online: https://arxiv.org/abs/1803.06386v1.

Sridhar, S., & Sanagavarapu, S. (2021). Multi-head self-attention transformer for dogecoin price prediction. In *2021 14th International Conference on Human System Interaction (HSI)*, pp. 1–6, Doi: https://doi.org/10.1109/HSI52170.2021.9538640.

Uras, N., Marchesi, L., Marchesi, M., & Tonelli, R. (2020). Forecasting Bitcoin closing price series using linear regression and neural networks models. *PeerJ Computer Science, 6*, e279. https://doi.org/10.7717/peerj-cs.279

Urquhart, A. (2016). The inefficiency of bitcoin. *Economic Letters, 148*, 80–82. https://doi.org/10.1016/j.econlet.2016.09.019

Wang, K., Ma, C., Qiao, Y., Lu, X., Hao, W., & Dong, S. (2021). A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction. *Physica A: Statistical Mechanics and its Applications, 583*, 126293. https://doi.org/10.1016/j.physa.2021.126293

Żbikowski, K. (2015). Using Volume Weighted Support Vector Machines with walk forward testing and feature selection for the purpose of creating stock trading strategy. *Expert Systems with Applications, 42*(4), 1797–1805.

Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services, 14*, 352–375.

Zoumpekas, T., Elias, H., & Manolis, V. (2020). ETH analysis and predictions utilizing deep learning. *Expert Systems with Applications, 162*, 113866.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.