JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

RNN-LSTM Model based Forecasting of Cryptocurrency Prices using Standard Scaler Transform

Ch. Shivani[1], B. Anusha[1], B. Druvitha[1], K. Kumara Swamy[2]

[1]UG Student, [2]Assistant Professor, [1,2]Department of Information Technology

[1,2]Malla Reddy Engineering College for Women (UGC-Autonomous), Maisammaguda, Secunderabad, Telangana, India

Abstract

Bitcoin is the worlds' most valuable cryptocurrency and is traded on over 40 exchanges worldwide accepting over 30 different currencies. It has a current market capitalization of 9 billion USD according to https://www.blockchain.info/ and sees over 250,000 transactions taking place per day. As a currency, Bitcoin offers a novel opportunity for price prediction due to its relatively young age and resulting volatility, which is far greater than that of fiat currencies. It is also unique in relation to traditional fiat currencies in terms of its open nature; no complete data exists regarding cash transactions or money in circulation for fiat currencies. Prediction of mature financial markets such as the stock market has been researched at length. Bitcoin presents an interesting parallel to this as it is a time series prediction problem in a market still in its transient stage. Traditional time series prediction methods such as Holt-Winters exponential smoothing models rely on linear assumptions and require data that can be broken down into trend, seasonal and noise to be effective. This type of methodology is more suitable for a task such as forecasting sales where seasonal effects are present. Therefore, the analysis of financial data for predicting the future bitcoin price has always been an important field of research with a direct and indirect effect on world economy. Due to the lack of seasonality in the Bitcoin market and its high volatility, these methods are not highly effective for this task. Given the complexity of the task, machine learning makes for an interesting technological solution based on its performance in similar areas. Hence, a time series analysis is utilized in this paper in order to find out the pattern of bitcoin price movement and forecasting the closing price of the next few days as well as analyzing the performance of the time series models.

Keywords: Bitcoin price, cryptocurrency prediction, time series analysis, recurrent neural networks, long-short term memory models.

1. INTRODUCTION

Digital transformation of economies is the most serious disruption that is taking place now in all economies and financial systems. The economies and financial systems of the world are becoming digital at an unprecedentedly fast pace. According to a recent report, the size of digital economy in 2025 is estimated to be 25% (23 trillion USD), consisting of tangible and intangible digital assets [1]. The most recent technology for establishing and spending digital assets is the distributed ledger technology (DLT), and its most well-known application being the cryptocurrency named Bitcoin [2]. Following these developments, blockchain technology has found its place in the intersections of Fintech and next-generation networks [3]. An important issue about the non-tangible digital assets, and especially cryptocurrencies, is price volatility. The price of Bitcoin (BTC) for the period of April

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

1, 2013, to December 31, 2019, can be seen in Fig. 1. BTC prices have exhibited extreme volatility in this period. The price has increased 1900% in the year 2017, consecutively losing 72% of its value in 2018 [4]. Prior to 2013, the popular interest in BTC, its usage in virtual transactions and its prices have been low. That period is not considered in our models. Although the BTC prices exhibit extraordinary volatility, BTC as a digital asset is quite resilient as it can regain its value after significant drops, and even when the uncertainty is high in the market such as during the COVID-19 pandemic [5].

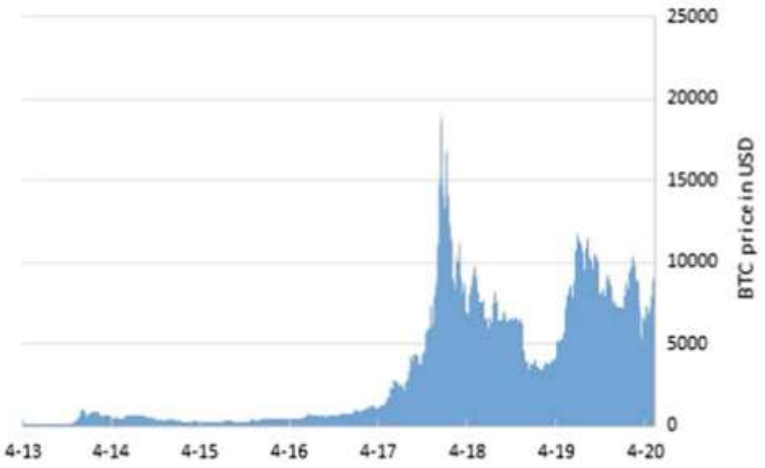

Fig. 1 Bitcoin (BTC) prices from April 2013 to April 2020.

Despite its rapidly changing nature, the price of BTC has been an area where various researchers have presented efforts for price forecast. A number of studies have discussed whether BTC prices are predictable using technical indicators and demonstrated the existence of significant return predictability [6, 7]. Other recent studies such as [8, 9] and [10] have applied various machine learning-related methods for end-of-day price forecast and price increase/decrease forecasting. [9] reported maximum accuracy up to 63% for forecasting of increase or decrease of prices. [10] reported 98% success rate for daily price forecast. However, the time periods of these studies have been limited by data—up to April 1, 2017 [10] and up to March 5, 2018 [9]. We believe that a current study is needed considering the volume of the BTC price movements that occurred after these dates. Secondly, the cited works focus on end-of-day closing price forecast and price increase/decrease forecasting for the next day prices.

2. RELATED WORK

When Bitcoin began to get worldwide attention at end of 2013, it witnessed a significant fluctuation in its value and number of transactions [11]. A strand of literature has examined the predictability of BTC returns through various parameters such as social media attention [12, 13] and BTC-related historical technical indicators [14]. One group studied the period from September 4, 2014, to August 31, 2018, by capturing the number of times the term "Bitcoin" has been tweeted. The results showed that the number of tweets on Twitter can influence BTC trading volume for the following day [15]. Moreover, [16] studied the influence of users comments in online platforms on price fluctuations and number of transaction of cryptocurrencies and found that BTC is particularly correlated with the number of positive comments on social media. They reported an accuracy of 79% along with Granger

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

causality test, which implies that user opinions are useful to predict the price fluctuations. When it comes to time-series forecasts, there are three different types of model-based approaches for time-series forecast according to [17]. The first approach, pure models, only uses the historical data on the variable to be predicted. Examples of pure time-series forecast models are Autoregressive Integrated Moving Average (ARIMA) [18] and Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) [19]. [20] presents an ARIMA-based time-series forecast for next-day BTC prices. However, we have not yet seen a study based on GARCH. Pure time-series models are more appropriate for univariate and stationary time-series data. In this paper, we focus on machine learning with higher level features rather than the traditional models for the following reasons. First of all, BTC prices are highly volatile and non-stationary. We demonstrate that BTC prices are non-stationary in the next section. Secondly, there are a large number of features in the data and the proposed machine learning methodology handles autocorrelation, seasonality and trend effects, while the training process of pure time-series models require manual tuning to address these effects. The second approach, explanatory models, uses a function of predictor variables to predict the target variable in a future time. Model-based time-series forecast approaches have the disadvantage of making a prior assumption about data distributions. For example, [20] and [21] are based on a log-transformation of the BTC prices. Similarly, [21] used daily BTC data from September 2011 up to August 2017 to conduct an empirical study on modeling and predicting the BTC price that compare the Bayesian neural network (BNN) with other linear and nonlinear benchmark models. They found that BNN performs well in predicting BTC log-transformed price, explaining the high volatility of the BTC price. However, the above-mentioned studies have used log-transformed prices for reporting performance metrics, which are misleading, as such values tend to be lower than performance metrics computed using real prices. We have analyzed this by calculating the performance metrics using log-normalized values and comparing against the non-log-normalized ones for our own results and found that although the log-normalized price forecast reports a much lower MAPE value, the actual error may be up to 10 times higher.

Since cryptocurrency prices are nonlinear and non-stationary, the assumptions on data distributions may have adverse effects on the forecast performance. Non-stationary time-series models exhibit evolving statistical distributions over time, which results in a changing dependency behavior between the input and output variables. Machine learning-based approaches utilize the inherent nonlinear and non-stationary aspects of the data. They can also take advantage of the explanatory features by taking into consideration the underlying factors affecting the predicted variable. There are several research studies on modeling and forecasting the price of BTC using machine learning, [22] used Bayesian regression method that utilizes latent source model which was developed by [23] to predict the price variation using BTC historical data. [24] used machine learning and feature engineering to investigate how the BTC network features can influence the BTC price movements. They obtained classification accuracy of 55%. [9] used artificial neural network (ANN) to achieve a classification accuracy of 65%. Furthermore, [25] predicted the BTC price using Bayesian optimized recurrent neural network (RNN) and long short-term memory (LSTM). The classification accuracy they achieved was 52% using LSTM with RMSE of 8%. They also reported that in forecasting, the nonlinear deep learning models performed better than ARIMA. [10] employed ANN and SVM algorithms in regression models to predict the minimum, maximum and closing BTS prices and reported that SVM algorithm performed best with MAPE of 1.58%. One of the latest studies in predicting BTC daily prices is by [8], which used high-dimensional features with different machine learning algorithms such as SVM,

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

LSTM and random forest. For next-day price forecast from July 2017 to January 2018, the highest accuracy of 65.3% was achieved by SVM.

3. PROPOSED SYSTEM

Below Figure gives an overview of the main ideas used in this paper. The ML-based time-series forecast method starts with the construction of a dataset. This is followed by the training of ML models and forecasts based on these models for different horizons of forecast. Time-series forecast on cryptocurrency prices has underlying interdependencies that are hard to understand and model. For example, there are statistical factors such as variance and standard deviation that change over time.

Fig. 2: Overview of proposed system for cryptocurrency prediction.

3.1 Dataset collection and upload dataset

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-09-17	465.864014	468.174011	452.421997	457.334015	457.334015	2.105680e+07
1	2014-09-18	456.859985	456.859985	413.104004	424.440002	424.440002	3.448320e+07
2	2014-09-19	424.102997	427.834991	384.532013	394.795990	394.795990	3.791970e+07
3	2014-09-20	394.673004	423.295990	389.882996	408.903992	408.903992	3.686360e+07
4	2014-09-21	408.084991	412.425995	393.181000	398.821014	398.821014	2.658010e+07
...	...	...	...	...	...	...	...
1927	2019-12-27	7238.141113	7363.529297	7189.934082	7290.088379	7290.088379	2.277736e+10
1928	2019-12-28	7289.031250	7399.041016	7286.905273	7317.990234	7317.990234	2.136567e+10
1929	2019-12-29	7317.647461	7513.948242	7279.865234	7422.652832	7422.652832	2.244526e+10
1930	2019-12-30	7420.272949	7454.824219	7276.308105	7292.995117	7292.995117	2.287413e+10
1931	2019-12-31	7294.438965	7335.290039	7169.777832	7193.599121	7193.599121	2.116795e+10

1932 rows × 7 columns

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

	Date	Open	High	Low	Close	Adj Close	Volume
1933	2020-01-02	7202.551270	7212.155273	6935.270020	6985.470215	6985.470215	2.080208e+10
1934	2020-01-03	6984.428711	7413.715332	6914.996094	7344.884277	7344.884277	2.811148e+10
1935	2020-01-04	7345.375488	7427.385742	7309.514160	7410.656738	7410.656738	1.844427e+10
1936	2020-01-05	7410.451660	7544.497070	7400.535645	7411.317383	7411.317383	1.972507e+10
1937	2020-01-06	7410.452148	7781.867188	7409.292969	7769.219238	7769.219238	2.327626e+10
...	...	...	...	...	...	...	...
2075	2020-05-23	9185.062500	9302.501953	9118.108398	9209.287109	9209.287109	2.772787e+10
2076	2020-05-24	9212.283203	9288.404297	8787.250977	8790.368164	8790.368164	3.251880e+10
2077	2020-05-25	8786.107422	8951.005859	8719.667969	8906.934570	8906.934570	3.128816e+10
2078	2020-05-26	NaN	NaN	NaN	NaN	NaN	NaN
2079	2020-05-27	8834.157227	8859.578125	8834.157227	8856.885742	8856.885742	2.914432e+10

147 rows × 7 columns

3.2 Dataset preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So, for this, we use data preprocessing tasks.

3.2.1 Need for Data Preprocessing: Real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

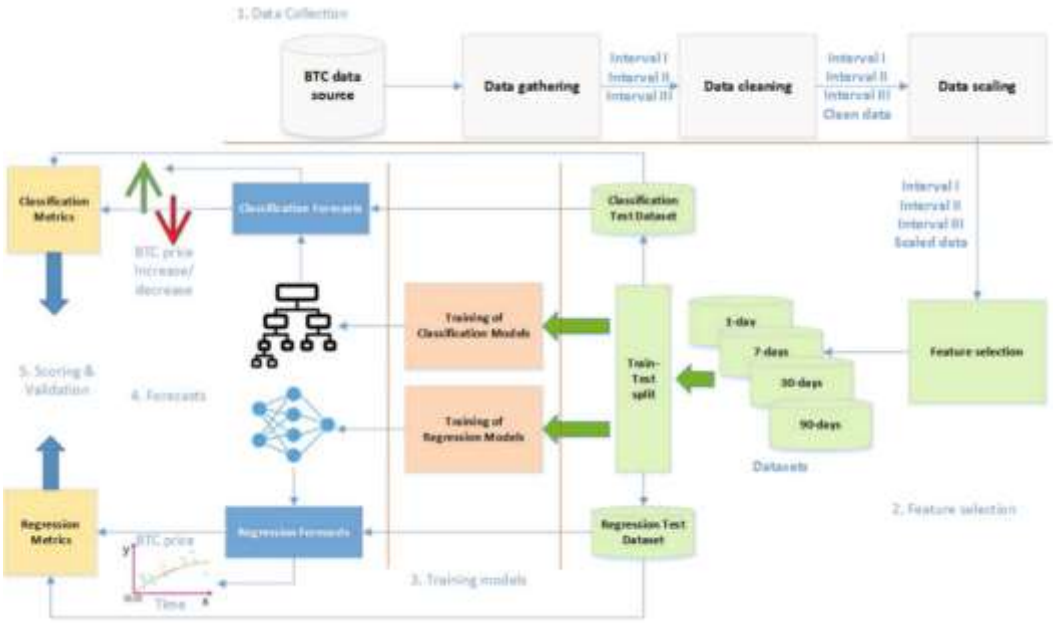

Figure 3. Data collection and splitting.

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

3.3 Splitting the Dataset

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
```

3.3.1 Explanation

- In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.
- In the second line, we have used four variables for our output that are
- x_train: features for the training data
- x_test: features for testing data
- y_train: Dependent variables for training data
- y_test: Independent variable for testing data
- In train_test_split() function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.
- The last parameter random_state is used to set a seed for a random generator so that you always get the same result, and the most used value for this is 42.

3.4 Recurrent Neural Networks

Recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

handwriting recognition or speech recognition. Recurrent Neural Network comes into the picture when any model needs context to be able to provide the output based on the input. Similarly, RNN remembers everything. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other. Let's say you have to predict the next word in a given sentence, in that case, the relation among all the previous words helps in predicting the better output. The RNN remembers all these relations while training itself. In order to achieve it, the RNN creates the networks with loops in them, which allows it to persist the information.

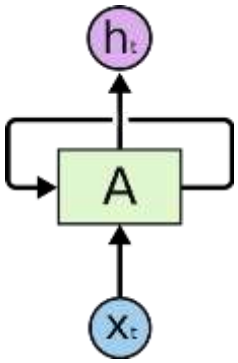

Figure 4. Recurrent unit.

This loop structure allows the neural network to take the sequence of input. If you see the unrolled version, you will understand it better.

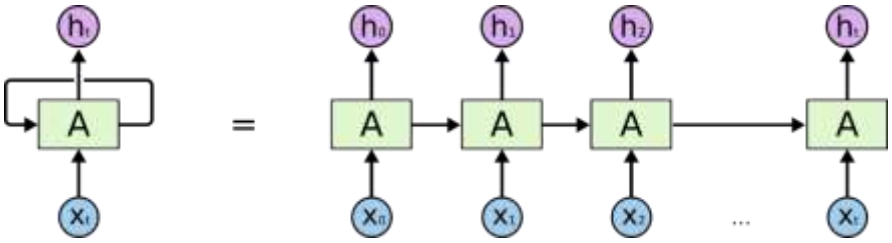

Figure 5. RNN structure.

As you can see in the unrolled version. First, it takes the x(0) from the sequence of input and then it outputs h(0) which together with x(1) is the input for the next step. So, the h(0) and x(1) is the input for the next step. Similarly, h(1) from the next is the input with x(2) for the next step and so on. This way, it keeps remembering the context while training. This way RNN works. RNN helps wherever we need context from the previous input.

The following are the few applications of the RNN:

- Next word prediction.
- Music composition.
- Image captioning
- Speech recognition
- Time series anomaly detection
- Stock market prediction

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

Now a days, RNN has become very popular as it helps in solving many real-life problems which the industries are facing. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network. Conversely, to handle sequential data successfully, you need to use recurrent (feedback) neural network. It can 'memorize' parts of the inputs and use them to make accurate predictions. These networks are at the heart of speech recognition, translation and more. So, let's dive into a more detailed explanation.

What is a Recurrent Neural Network?

Training a typical neural network involves the following steps:

- Input an example from a dataset.
- The network will take that example and apply some complex computations to it using randomly initialized variables (called weights and biases).
- A predicted result will be produced.
- Comparing that result to the expected value will give us an error.
- Propagating the error back through the same path will adjust the variables.
- Steps 1–5 are repeated until we are confident to say that our variables are well-defined.
- A predication is made by applying these variables to a new unseen input.

Here x_1, x_2, x_3, …, x_t represent the input words from the text, y_1, y_2, y_3, …, y_t represent the predicted next words and h_0, h_1, h_2, h_3, …, h_t hold the information for the previous input words.

Since plain text cannot be used in a neural network, we need to encode the words into vectors. The best approach is to use word embeddings (word2vec or GloVe) but for the purpose of this article, we will go for the one-hot encoded vectors. These are (V,1) vectors (V is the number of words in our vocabulary) where all the values are 0, except the one at the i-th position. For example, if our vocabulary is apple, apricot, banana, …, king, … zebra and the word is banana, then the vector is [0, 0, 1, …, 0, …, 0]. Typically, the vocabulary contains all English words. That is why it is necessary to use word embeddings. Let's define the equations needed for training:

$$
\begin{aligned}
&1)\quad h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)\\
&2)\quad y_t = softmax(W^{(S)}h_t)\\
&3)\quad J^{(t)}(\theta) = \sum_{i=1}^{|V|}(y_{t_i}^{'} log y_{t_i})
\end{aligned}
\tag{1}
$$

1) —holds information about the previous words in the sequence. As you can see, h_t is calculated using the previous h_{t-1} vector and current word vector x_t. We also apply a non-linear activation function f (usually tanh or sigmoid) to the final summation. It is acceptable to assume that h_0 is a vector of zeros.

2) — calculates the predicted word vector at a given time step t. We use the softmax function to produce a $(V, 1)$ vector with all elements summing up to 1. This probability distribution gives us the index of the most likely next word from the vocabulary.

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

3) — uses the cross-entropy loss function at each time step t to calculate the error between the predicted and actual word.

3.5 LSTM NEURAL NETWORK

Deep learning is a new research direction in the field of artificial intelligence. It is developed on the basis of shallow neural networks with the improvement of computer hardware levels and the explosive growth of the current data volume. Deep learning and shallow neural network structure are both layered. Each layer will process the data input to the model and combine low-level features into potential high-level features by learning data rules. Compared with shallow models, deep learning can express complex high dimensionality such as high-variable functions and find the true relationships within the original data better. In the 1980s, artificial neural network back propagation algorithm was born. This method can automatically learn data rules from a large amount of training data without manual intervention. At present, deep learning is the most concerned research direction in the field of artificial intelligence, which completely subverts the shallow model in traditional machine, proposes a deep learning network model, and elevates it to a new height from theory to application. CNN (convolutional neural network) and RNN are two types of classical deep learning network structures now.

Because there are connections between neurons in the RNN layer, the network can learn the change law of sequence data before and after, and the internal sequence rules of data is easy to be mined. Thus, RNN is widely used in the field of sequence data processing such as speech recognition and machine translation. However, this structure also has some problems. When data is transmitted backward, the problem of gradient disappearance or gradient explosion is unavoidable, which limits its processing of long-term dependencies. The LSTM network changes the way of gradient transmission during backpropagation by adding multiple special computing nodes in the hidden layer of RNN, which effectively slows the problem of gradient disappearance or gradient explosion. Its model structure is shown in Figure 6.

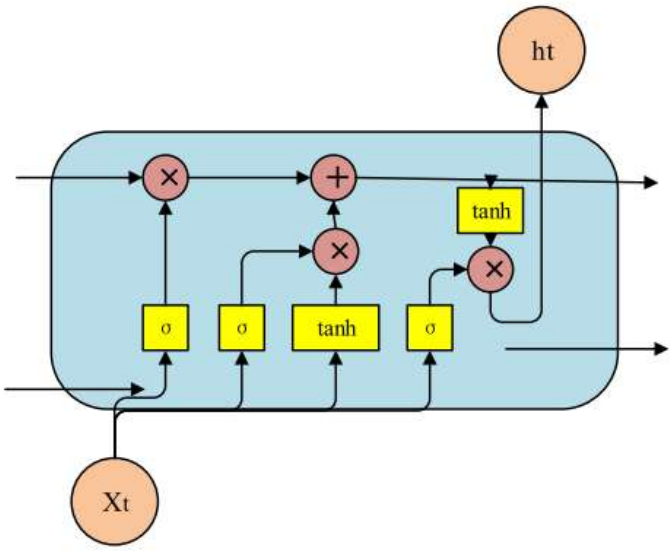

Fig. 6: LSTM model structure.

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

Where h_{t-1} represents the output of the previous cell, and x_t represents the input of the current cell. σ represents the sigmoid function. The difference between LSTM and RNN is that it adds a ‘‘processor’’ to the algorithm to determine the usefulness of the information. The structure of this processor is called a cell. Three gates are placed in a cell, which are called *Input gate*, *Forget gate*, and *Output gate*. A piece of information enters the LSTM network, and it can be judged whether it is useful according to the rules. Only the information that meets the algorithm authentication will be left, and the non-conforming information will be forgotten through the *Forget gate*.

3.6 FORGET GATE

The first step for data entering the LSTM is to decide what information should be lost and what retained. This decision is made by the Forget gate, which reads h and x and outputs a value between 0 and 1, where 1 means ‘‘complete reserved’’, 0 means ‘‘completely discarded’’. Forget gate is calculated as:

$$f_t = \sigma\left(W_f * [h_{t-1}, x_t] + b_f\right) \tag{2}$$

In the formula, f_t is the calculation result of the Forget gate which is mainly used to control the retention of the information transmitted from the unit state at the previous moment to the unit state at the current moment. [] indicates that the two vectors are spliced, h_{t-1} is the output of the unit at the previous moment, and are the weight and bias of Forget gate, W_f and b_f are Sigmoid activation functions.

3.6.1 INPUT GATE

Input gate determines the addition of new information, and its operation process includes sigmoid layer and tanh layer. The sigmoid layer determines the information that needs to be updated. The calculation formula is:

$$i_t = \sigma\left(W_i * [h_{t-1}, x_t] + b_i\right) \tag{3}$$

In the formula, i_t is the calculation result of the input gate, and the input gate also has independent weight and bias. The role of the tanh layer is to generate a vector of candidate update information. Its calculation formula is:

$$\tilde{C}_t = \tanh\left(W_c * [h_{t-1}, x_t] + b_c\right) \tag{4}$$

$\tilde{C}_t$ is the unit state of the current input, the unit state of the current moment is C_t, and its calculation formula is:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

3.6.2 OUTPUT GATE

Output gate is roughly the same as the Input gate, and its operation flow includes sigmoid layer and tanh layer. The sigmoid layer determines the output part of the information, and the calculation formula is:

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \tag{6}$$

Finally get the output of the current moment h_t:

$$h_t = o_t * \tanh\left(c_t\right) \tag{7}$$

The forward propagation of LSTM calculates the cell state C_t and h_t the output of the current moment and completes the forward propagation calculation of the network. The backpropagation of LSTM is like the back-propagation principle of RNN. Finally, the weights and biases of all parts of the network are updated to complete the model training.

4. RESULTS AND DISCUSSION

4.1 Sample Training Data

	Open	High	Low	Close	Volume
0	465.864014	468.174011	452.421997	457.334015	21056800.0
1	456.859985	456.859985	413.104004	424.440002	34483200.0
2	424.102997	427.834991	384.532013	394.795990	37919700.0
3	394.673004	423.295990	389.882996	408.903992	36863600.0
4	408.084991	412.425995	393.181000	398.821014	26580100.0

4.2 Outcome of Standard scaler transform

```
array([[1.49732345e-02, 1.29013200e-02, 1.49400698e-02, 1.44534769e-02,
        3.35749244e-04],
       [1.45066780e-02, 1.23321258e-02, 1.28489753e-02, 1.27508263e-02,
        6.33453324e-04],
       [1.28093283e-02, 1.08719155e-02, 1.13293978e-02, 1.12164013e-02,
        7.09650970e-04],
       ...,
       [3.70008086e-01, 3.67365217e-01, 3.78051927e-01, 3.74990337e-01,
        4.97548412e-01],
       [3.75325771e-01, 3.64390763e-01, 3.77862744e-01, 3.68279031e-01,
        5.07057851e-01],
       [3.68805505e-01, 3.58377151e-01, 3.72197021e-01, 3.63134123e-01,
        4.69226533e-01]])
```

4.3 Building LSTM model

```
regressor = Sequential()
regressor.add(LSTM(units = 50, activation = 'relu', return_sequences = True, input_shape = (X_train.shape[1], 5)))
regressor.add(Dropout(0.2))
```

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

```
regressor.add(LSTM(units = 60, activation = 'relu', return_sequences = True))
regressor.add(Dropout(0.3))

regressor.add(LSTM(units = 80, activation = 'relu', return_sequences = True))
regressor.add(Dropout(0.4))

regressor.add(LSTM(units = 120, activation = 'relu'))
regressor.add(Dropout(0.5))

regressor.add(Dense(units =1))
```

4.4 Model summary

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 60, 50)            11200
_________________________________________________________________
dropout (Dropout)            (None, 60, 50)            0
_________________________________________________________________
lstm_1 (LSTM)                (None, 60, 60)            26640
_________________________________________________________________
dropout_1 (Dropout)          (None, 60, 60)            0
_________________________________________________________________
lstm_2 (LSTM)                (None, 60, 80)            45120
_________________________________________________________________
dropout_2 (Dropout)          (None, 60, 80)            0
_________________________________________________________________
lstm_3 (LSTM)                (None, 120)               96480
_________________________________________________________________
dropout_3 (Dropout)          (None, 120)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 121
=================================================================
Total params: 179,561
Trainable params: 179,561
Non-trainable params: 0
_________________________________________________________________
```

4.4 Prediction results

Figure 20. Cryptocurrency Price Prediction using RNN-LSTM

ISSN- 2394-5125 VOL 10, ISSUE 01, 2022

5. CONCLUSION

In this project, we address short-term to mid-term BTC price forecasts using ML models. It is the first study that takes into consideration all the price indicators up to May 27, 2020, and provides highly accurate end-of-day, short-term (7 days) and mid-term (30 and 90 days) BTC price forecasts using machine learning. The LSTM showed the best overall performance. Performance evaluation results show an improvement over the latest literature in daily closing price forecast and price increase/decrease forecasting. The results are satisfactory and show potential for further applications in different areas such financial technology, blockchain and AI development. Our results show that it is possible to forecast the actual BTC price with very low error rates, while it is much harder to forecast its rise and fall. The classification model performance scores presented are the best in the literature. Having said that, the classification models for Bitcoin need to be further studied. As further work, hourly BTC prices and technical indicators may be utilized as well as using ensemble models that combined different types of models for making forecast. Further work which can be followed on the basis of this paper is investigating the use of artificial intelligence for modeling the price of cryptocurrencies as a basis for measuring the risk factor for the financial usage of blockchain technology. This model could also be useful in detecting fraudulent activities and anomalous behavior. When the actual behavior (price) changes significantly from the modeled behavior, this may indicate the effect of external factors such as major global events as well as fraudulent activities such as artificial pumps and dumps. While price modeling and forecast is not the only tool to detect such external factors, one of the possible applications of such models is in the detection and prevention of fraudulent activities. Our future research will be focusing on such application areas. Using external data inputs related to global events and global financial risk, a combination of machine learning-based price models and anomaly detection methods may be utilized to assess and predict the stability of cryptocurrencies.

REFERENCES

[1] Xu W, Cooper A (2017) Digital spillover: measuring the true impact of the digital economy. Huawei and Oxford Economics. https://www.huawei.com/minisite/gci/en/digital-spillover/files/gci-digital-spillover.pdf. Accessed 25 December 2022

[2] Oliver J (2013) Mastering blockchain distributed ledger technology, and smart contracts explained-packt publishing, vol 53. Packt Publishing Ltd, https://doi.org/10.1017/CBO9781107415324.004, arXiv:1011.1669v3

[3] Unal D, Hammoudeh M, Kiraz MS (2020) Policy specification and verification for blockchain and smart contracts in 5G networks. ICT Express 6(1):43–47. https://doi.org/10.1016/j.icte.2019.07.002

[4] Morris DZ (2017) Bitcoin hits a new record high, but stops short of \$20,000 | Fortune. https://fortune.com/2017/12/17/bitcoin-record-high-short-of-20000/

[5] Shukla S, Dave S (2020) Bitcoin beats coronavirus blues. https://economictimes.indiatimes.com/markets/stocks/news/bitcoin-beats-coronavirus-blues/articleshow/75049718.cms

[6] Liu L (2019) Are Bitcon returns predictable?: Evidence from technical indicators. Physica A: Stat Mech Appl 533:121950. https://doi.org/10.1016/j.physa.2019.121950

[7] Huang JZ, Huang W, Ni J (2018) Predicting Bitcoin returns using high-dimensional technical indicators. J Finance Data Sci 5(3):140–155. https://doi.org/10.1016/j.jfds.2018.10.001

[8] Chen Z, Li C, Sun W (2020) Bitcoin price prediction using machine learning: an approach to sample dimension engineering. J Comput Appl Math 365:112395

[9] Adcock R, Gradojevic N (2019) Non-fundamental, non-parametric Bitcoin forecasting. Physica A: Stat Mech Appl 531:121727. https://doi.org/10.1016/j.physa.2019.121727

[10] Mallqui DC, Fernandes RA (2019) Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques. Appl Soft Comput 75:596–606

[11] Urquhart A (2018) What causes the attention of Bitcoin? Econ Lett 166:40–44

[12] Philippas D, Rjiba H, Guesmi K, Goutte S (2019) Media attention and Bitcoin prices. Finance Res Lett 30:37–43.

[13] Abraham J, Higdon D, Nelson J, Ibarra J (2018) Cryptocurrency price prediction using tweet volumes and sentiment analysis. SMU Data Sci Rev 1(3):1

[14] Huang JZ, Huang W, Ni J (2019) Predicting Bitcoin returns using high-dimensional technical indicators. J Finance Data Sci 5(3):140–155.

[15] Shen D, Urquhart A, Wang P (2019) Does twitter predict Bitcoin? Econ Lett 174:118–122.

[16] Kim YB, Kim JG, Kim W, Im JH, Kim TH, Kang SJ, Kim CH (2016) Predicting fluctuations in cryptocurrency transactions based on user comments and replies. PloS one 11(8):e0161197. https://doi.org/10.1371/journal.pone.0161197

[17] Hyndman RJ, Athanasopoulos G (2018) Forecasting: principles and practice, 2nd edn. OTexts, Melbourne, Australia. https:// otexts.com/fpp2/. Accessed 25 December 2022

[18] Bakar NA, Rosbi S (2017) Autoregressive integrated moving average (ARIMA) model for fore-casting cryptocurrency exchange rate in high volatilityenvironment: A new insight of bitcoin transaction. Int J Adv Eng Res Sci 4(11):130–137. https:// doi.org/10.22161/ijaers.4.11.20

[19] Garcia RC, Contreras J, Van Akkeren M, Garcia JBC (2005) A GARCH forecasting model to predict day-ahead electricity prices. IEEE Trans Power Syst 20(2):867–874

[20] Munim ZH, Shakil MH, Alon I (2019) Next-day Bitcoin price forecast. J Risk Financ Manag 12(2):103. https://doi.org/10.3390/ jrfm12020103

[21] Jang H, Lee J (2017) An empirical study on modeling and prediction of Bitcoin prices with bayesian neural networks based on blockchain information. IEEE Access 6:5427–5437

[22] Shah D, Zhang K (2014) Bayesian regression and Bitcoin. In: 2014 52nd annual Allerton conference on communication, control, and computing (Allerton), IEEE, pp 409–414.

[23] Chen GH, Nikolov S, Shah D (2013) A latent source model for nonparametric time series classification. In: Advances in neural information processing systems, pp 1088–1096.

[24] Greaves A, Au B (2015) Using the Bitcoin transaction graph to predict the price of Bitcoin. https://doi.org/10.1109/CEC.2010.5586007

[25] McNally S, Roche J, Caton S (2018) Predicting the price of Bitcoin using machine learning. In: Proceedings—26th Euromicro international conference on parallel, distributed, and network-based processing, PDP 2018 pp 339–343. https://doi.org/10.1109/ PDP2018.2018.00060.

[26] Ammer, M.A. and Aldhyani, T.H., 2022. Deep Learning Algorithm to Predict Cryptocurrency Fluctuation Prices: Increasing Investment Awareness. Electronics, 11(15), p.2349.

[27] Mohil Maheshkumar Patel, Sudeep Tanwar, Rajesh Gupta, Neeraj Kumar, A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions, Journal of

Information Security and Applications, Volume 55, 2020, 102583, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2020.102583.

[28] S. Tanwar, N. P. Patel, S. N. Patel, J. R. Patel, G. Sharma and I. E. Davidson, "Deep Learning-Based Cryptocurrency Price Prediction Scheme with Inter-Dependent Relations," in IEEE Access, vol. 9, pp. 138633-138646, 2021, doi: 10.1109/ACCESS.2021.3117848.

[29] Lahmiri, S., Bekiros, S. Deep Learning Forecasting in Cryptocurrency High-Frequency Trading. Cogn Comput 13, 485–487 (2021). https://doi.org/10.1007/s12559-021-09841-w

[30] K. Rathan, S. V. Sai and T. S. Manikanta, "Crypto-Currency price prediction using Decision Tree and Regression techniques," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 190-194, doi: 10.1109/ICOEI.2019.8862585.

[31] Reaz Chowdhury, M. Arifur Rahman, M. Sohel Rahman, M.R.C. Mahdy, An approach to predict and forecast the price of constituents and index of cryptocurrency using machine learning, Physica A: Statistical Mechanics and its Applications, Volume 551, 2020, 124569, ISSN 0378-4371, https://doi.org/10.1016/j.physa.2020.124569.

[32] Z. Shahbazi and Y. -C. Byun, "Improving the Cryptocurrency Price Prediction Performance Based on Reinforcement Learning," in IEEE Access, vol. 9, pp. 162651-162659, 2021, doi: 10.1109/ACCESS.2021.3133937.

[33] S. Mohapatra, N. Ahmed and P. Alencar, "KryptoOracle: A Real-Time Cryptocurrency Price Prediction Platform Using Twitter Sentiments," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 5544-5551, doi: 10.1109/BigData47090.2019.9006554.

[34] J. Sun, Y. Zhou and J. Lin, "Using machine learning for cryptocurrency trading," 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), 2019, pp. 647-652, doi: 10.1109/ICPHYS.2019.8780358.

[35] S. Rahman, J. N. Hemel, S. Junayed Ahmed Anta, H. A. Muhee and J. Uddin, "Sentiment Analysis Using R: An Approach to Correlate Cryptocurrency Price Fluctuations with Change in User Sentiment Using Machine Learning," 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), 2018, pp. 492-497, doi: 10.1109/ICIEV.2018.8641075.

[36] H. Baek, J. Oh, C. Y. Kim and K. Lee, "A Model for Detecting Cryptocurrency Transactions with Discernible Purpose," 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), 2019, pp. 713-717, doi: 10.1109/ICUFN.2019.8806126.

[37] J. Kim, S. Kim, H. Wimmer and H. Liu, "A Cryptocurrency Prediction Model Using LSTM and GRU Algorithms," 2021 IEEE/ACIS 6th International Conference on Big Data, Cloud Computing, and Data Science (BCD), 2021, pp. 37-44, doi: 10.1109/BCD51206.2021.9581397.