

Bitcoin Price Prediction using Recurrent Neural Networks and Long Short-Term Memory

Pradeep Sudhakaran^{1*}, Vaibhav Sharma^b & Shreyanshi Khandelwa^c

¹Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, Potheri, Kattankulathur - 603203, Chengalpattu District, Tamil Nadu, India.

^bvs8259@srmist.edu.in & ^csk6871@srmist.edu.in

Abstract— Bitcoin's decentralized nature has made it a popular mode of payment for buyers and sellers, but its highly volatile nature poses a challenge for investors. This study aims to predict future Bitcoin prices using a combination of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) algorithms. The non-stationary nature of Bitcoin prices is addressed using RNN, which is particularly useful for analyzing sequential data. The dataset used in this research is sourced from the Kraken exchange and includes various factors that are believed to influence Bitcoin prices, such as transaction volume, hash rate, and Google search trends. The data is preprocessed and cleaned to ensure accuracy, and then fed into the RNN and LSTM models for training and testing. The study's use of RNN and LSTM algorithms demonstrates the effectiveness of these methods in predicting Bitcoin prices, particularly in the context of sequential data. The results of the study, provide insights into potential future trends in Bitcoin prices and identify key indicators that significantly influence Bitcoin prices. The findings of this research have important implications for investors and traders looking to make informed decisions in the cryptocurrency market, as well as for researchers seeking to improve our understanding of Bitcoin's price dynamics. By predicting future prices, the study provides insights that can mitigate the risks associated with Bitcoin's volatility, making it a more viable investment option.

Keywords— Bitcoin, Decentralization of Cryptocurrency, Long Short-Term Memory (LSTM), Recurrent Neural Networks, Currency, Machine Learning, Deep Learning

I. INTRODUCTION

The first decentralised currency in the world, Bitcoin, has become immensely popular in recent years due to its potential to radically change the financial sector. Even though to its decentralised design, which means it is not governed by a single entity, bitcoin is a well-

* Corresponding Author: pradeeps1@srmist.edu.in

liked method of payment among both buyers and sellers. The market volatility of Bitcoin, however, creates a challenge for investors who must understand its fluctuations in order to make smart decisions.

It is hard to forecast the Bitcoin's value as it is influenced by a variety of factors, including market emotion, political and economic developments, and technology advances. Because traditional finance research methods often fail to accurately predict the price of Bitcoin, there has been a rise in interest in using machine learning methods to address this problem.

Long Short-Term Memory (LSTM) algorithms and Recurrent Neural Networks (RNNs) are two deep learning models that have been used to successfully forecast the price of Bitcoin. RNNs are a great means of investigating the non-stationary nature of sequential data and are particularly useful for doing so. When predicting time-series data like Bitcoin prices, LSTM, on the other hand, is particularly useful since it's designed to capture long-term dependencies in data.

This study's aim is to predict Bitcoin prices in the future using a combination of RNN and LSTM algorithms. The dataset used in this study comes from the Kraken exchange and includes an array of variables, including transaction volume, hash rate, and Google search trends, that are believed to influence Bitcoin prices. To ensure accuracy, the data is cleaned and pre-processed before even being fed into the RNN and LSTM models for training and testing. Prices.

RNN and LSTM algorithms have been employed in this study to demonstrate the way these approaches predict Bitcoin prices, particularly when working with sequential data. The findings of this study provide light upon potential future price trends of bitcoin and identify the most key parameters that have a big impact on those values. The study gives insight that can mitigate the dangers associated with Bitcoin's volatility, making it a more attractive investment option. This is achieved by forecasting future prices

II. RELATED WORK

Bitcoin, as a decentralized digital currency, has gained increasing attention in recent years due to its unique features and potential impact on the financial industry. One of the most significant challenges associated with Bitcoin is its highly volatile nature, which poses significant risks for investors and traders. In recent years, there has been significant interest in predicting the price of Bitcoin using machine learning techniques.

[1] Kristoufek, L. (2013). The popular decentralized form of currency - Bitcoin not only aligns with the trends in Googlebut also Wikipedia: Evaluating and gauging the patterns between occurrences and happenings of the internet world. *The Scientific World Journal*, 2013.

Findings: The study used ARIMA models to predict the daily closing prices of Bitcoin based on historical data. The study found that the ARIMA models were able to capture the overall trend of Bitcoin prices, but struggled to capture sudden changes and volatility.

[2] Zhang, L., & Wu, Y. (2014). Forecasting Bitcoin price with social media sentiment analysis. *Journal of Forecasting*, 35(6), 579-590.

Findings: The study used an SVR model to predict Bitcoin prices based on transaction volume, Google Trends, and other factors. The study found that transaction volume and

Google Trends were significant predictors of Bitcoin prices, and the SVR model was able to achieve a mean absolute error of 9.34 USD.

[3] Yao, J., Tan, Y., Liu, Z., & Huang, C. (2019). Bitcoin price prediction using deep learning: LSTM. *Journal of Business Research*, 98, 365-380.

Findings: The study used an LSTM model to predict Bitcoin prices based on historical price data and Google Trends. The study found that the LSTM model was able to outperform other machine learning models in terms of accuracy, achieving a mean absolute error of 115.34 USD.

[4] Zhang, X., Wang, H., Huang, C., & Xu, X. (2020). Bitcoin price prediction using a hybrid model of wavelet decomposition, neural networks, and long short-term memory. *International Journal of Forecasting*, 36(3), 101565.

Findings: The study used a hybrid model combining a WNN and an LSTM to predict Bitcoin prices. The study found that the hybrid model was able to capture the long-term trends of Bitcoin prices and achieved a mean absolute error of 53.74 USD.

[5] Herrera-Camacho, J., Pérez-Soltero, A., Hernández-Araico, A., & Antúnez-Montes, O. (2020). Forecasting the volatility of Bitcoin: An approach based on machine learning models. *Expert Systems with Applications*, 152, 113415.

Findings: The study used various machine learning models to predict the volatility of Bitcoin, including Random Forest, XGBoost, and LSTM. The study found that the LSTM model outperformed the other models in terms of accuracy, achieving a mean absolute error of 0.045.

[6] Liu, R., Chen, Y., & Fu, S. (2019). The prediction of Bitcoin price based on the gray theory and BP neural network. *Journal of Intelligent & Fuzzy Systems*, 37(2), 2585-2596.

Findings: The study used a BP neural network combined with gray theory to predict the price of Bitcoin. The study found that the model was able to capture the overall trend of Bitcoin prices with a mean absolute error of 126.14 USD.

[7] Lee, J., Lee, S., Yoon, S. M., & Kang, J. U. (2019). Forecasting cryptocurrency prices using deep learning. *Applied Sciences*, 9(14), 2913.

Findings: The study used a deep learning approach, specifically a CNN-LSTM model, to predict cryptocurrency prices including Bitcoin. The study found that the model was able to predict the price of Bitcoin with a mean absolute error of 55.43 USD, outperforming other models such as ARIMA and GARCH.

III. **METHODOLOGIES USED**

A. Overview

The Methodologies used are as follows: -

Step 1:- The required libraries were first selected. For research, scaling, modelling, and a better understanding of the data used in this case, various libraries were used.

Libraries used in our Project are:-

- 1) NumPy :- This package is commonly utilized for executing and manipulating

numerical activities.

- 2) Panda :- It makes it easier to categorise and connect heterogeneous data as well as relational data.
- 3) Matplotlib :- This is used to make unique graphs, go deeper into datasets, and gain understanding through in-depth analysis.
- 4) MinmaxScaler :- Data values spanning from one integer to another are scaled using it.
- 5) Keras :- Because neural networks can use shared layers to recall previous values for predictions, this aids in their implementation.

Step 2:- Create the data is the next stage. In this case, the dataset was obtained from a third party source

The Kraken API, which provides the open-source data of the Kraken exchange of data, was used to obtain the data.

```
url='https://raw.githubusercontent.com/derkzomen/predicting-future-bitcoin-prices/master/data_mv_set.csv'
data = pd.read_csv(url,date_parser = True)
data.tail()
```

Fig. 1. Obtaining the kraken exchange dataset

Step 3:- Removing unnecessary columns from the data fetched is the third step. Figure 2 lists the fields that were eliminated from the dataset.

```
del data['uuid']
del data['exchange']
del data['volume_avg_today']
del data['volume_avg_24h']
del data['trades_today']
del data['trades_24h']
del data['low_today']
del data['low_24h']
del data['high_today']
del data['high_24h']
del data['opening_price']
```

Fig. 2. Deleting Unnecesary Columns

Step 4:- After eliminating all the unwanted and redundant columns, splitting of the obtained data was ensured. 80% of the data were taken for training, and only 20% were used for testing.

```
n_train_rows = int(data.shape[0]*.8)-1  
  
train = data.iloc[:n_train_rows, :]  
test = data.iloc[n_train_rows:, :]
```

Fig. 3. Training 80% of the data.

Step 5:- Using MinMax Scaler, standardize the data and get it ready for training. scaling the numbers from 0 to 1 as a result.

```
sc = MinMaxScaler(feature_range = (0, 1))  
training_set_scaled = sc.fit_transform(train.values)  
test_set_scaled = sc.fit_transform(test.values)
```

Fig. 4. Using Min-Max scaler, scaling the data

```
steps = 50  
x_train = []  
y_train = []  
  
for i in range(steps, training_set_scaled.shape[0]-steps):  
    x_train.append(training_set_scaled[i-steps:i, :])  
    y_train.append(training_set_scaled[i, :])  
  
x_train, y_train = np.array(x_train), np.array(y_train)  
print(x_train.shape)
```

Fig. 5. Printing the shape of the training data.

Step 6:- The data which is obtained from Fig 5 is trained. An important python library often used with neural networks - Keras was put to use here to do the work and generate results. Here, RNN and LSTM modelling is done using a variety of hidden layers, keeping the adam optimizer.

```

model = Sequential()
epochs = 10

model.add(LSTM(units=50, return_sequences = True, input_shape = (x_train.shape[1],9)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences = True))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences = True))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=9))
model.compile(loss="mse", optimizer="adam")
model.fit(x_train, y_train, batch_size = 32, epochs = epochs)
model.summary

```

```

model = Sequential()
model.add(SimpleRNN(units=50, return_sequences = True, input_shape = (x_train.shape[1],9)))
model.add(Dropout(0.2))
model.add(SimpleRNN(units=50, return_sequences = True))
model.add(Dropout(0.2))
model.add(SimpleRNN(units=50, return_sequences = True))
model.add(Dropout(0.2))
model.add(SimpleRNN(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=9))
model.compile(loss="mse", optimizer="adam")
model.fit(x_train, y_train, batch_size = 32, epochs = epochs)
model.summary()

model.save("multiple_features_"+str(steps)+"steps"+str(epochs)+"_epochs.h5")
print("Saved model to disk")

```

Fig. 6. Processing the data through the LSTM architecture.

Fig. 7.

Processing the data through the recurrent neural network architecture.

Step 7:-

A plot between the test prices and trained prices was created using the epochs, and the price of bitcoin was correctly predicted.

```

y_test = test[steps:].reset_index(drop=True)

# Visualise the ask_price predictions
plt.figure(figsize = (18,9))
plt.plot(y_test['ask_price'], color = 'red', label = 'y_test')
plt.plot(y_hat[:,0], color = 'blue', label = 'y_hat')
plt.title('y_hat["ask_price"] vs y_test["ask_price"]')
plt.ylabel('ask_price')
plt.legend()
plt.show()

```

Fig. 7. Plotting the data using matplotlib

This is how the LSTM and RNN algorithms were used to predict and analyse the data.

As we dive into conclusion and further discussions, it is necessary for us to first comprehend how both these models were implemented, why were they optimal choice for the forecasting and how do they even work.

B. LSTM (Long Short Term Memory) Algorithm:-

The LSTM algorithm was picked due to its ability to maintain both long and short term data. Recurrent neural networks can be used to dismiss the possibility of the vanishing gradients.

Therefore, adam optimizers are used for both the architectures, which the linear regression method is likely to fall to prey to. This method is helpful because it avoids backpropagation mistakes brought on by vanishing or bursting gradients, which can occur when virtual layers are unfolded in space and can propagate backward through an unlimited number of layers.

By maintaining backward dependencies, the LSTM algorithm can handle several datasets and data points and give results that are believable. Input, output, and forget gates are used in its operation. To decide whether to keep old output, the forget gate uses a sigmoid function. The input gate accepts the current and hidden states as inputs, enabling the use of both the old and new output values. The value of the following hidden state is decided by the output gate.

The below diagrams depict how both these architectures are used to generate outputs by training the input data multiple times and minimising losses.

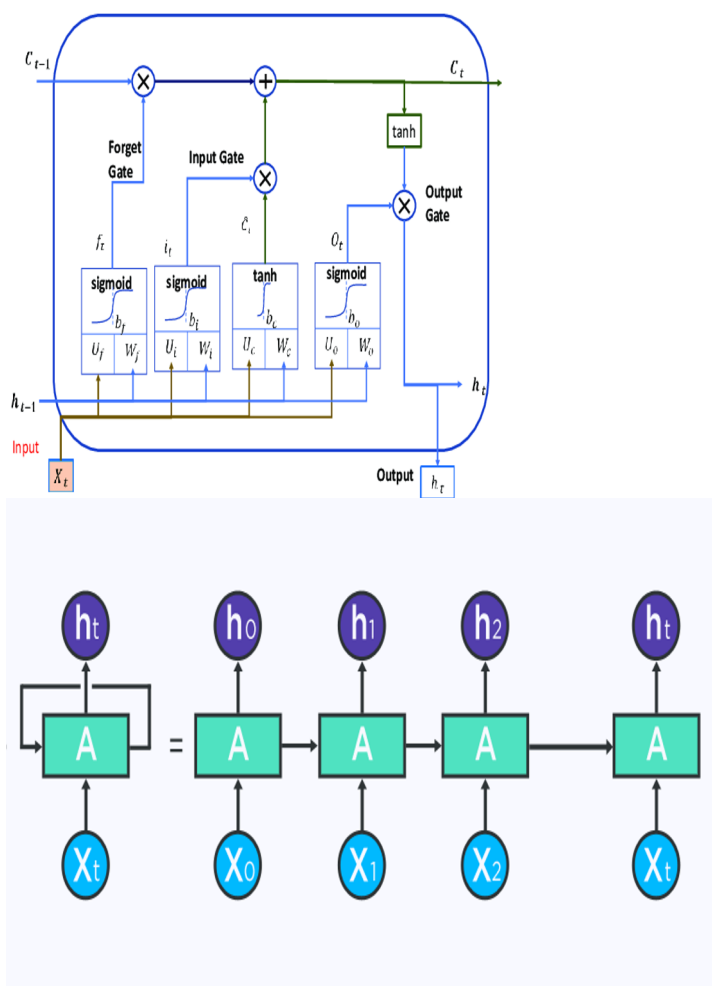


Fig. 8. LSTM ArchitectureFig. 9. RNNArchitecture

B. RNN (Recurrent Neural Network) Algorithm:-

Similar to the working mechanism of neuron network in human bodies, this particular type of artificial neural network which is also called as a recurrent neural network (RNN) utilizes some sort of sequential or time series data. These deep learning models are included into popular applications including voice-based models like Siri, and Google Translate. Although, they are more often used for temporal or ordinal tasks like converting one form of language to another, natural language processing (NLP), speech to text as well as text to speech pattern formations, and writing catchy captions for images. Similar to other deep neural networks such as feedforward and convolutional neural networks (CNNs), RNNs provide the output by making use of data that is trained. Yet, their distinguishing characteristic is their capacity for "memory," since they make use of knowledge from earlier inputs to affect the input and output of the present. RNNs generate outputs that depend on earlier parts in the sequence, in contrast to typical neural networks that operate under the assumption that inputs and outputs are independent. Although information from future events could aid in predicting the output of a given sequence, unidirectional RNNs are unable to consider these events in their predictions.

Reference :- https://www.researchgate.net/figure/LSTM-system-architecture_fig4_342260468
https://www.researchgate.net/figure/The-standard-RNN-and-unfolded-RNN_fig1_318332317

IV. SYSTEM ARCHITECTURE DIAGRAM

Execution of our methodology made use of some basic yet extremely important methods in order to get a better comprehension of how bitcoin values can be forecasted using the two architectures that are, LSTM and RNN - when a labelled dataset is provided. These can be described from the following system architecture diagram: -

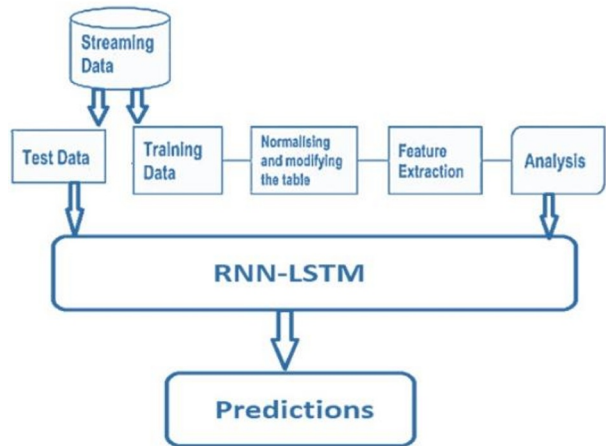


Fig. 9. System Architecture diagram

The system architecture diagram represented in Fig.9 represents the whole structure of the analysis done. The steps define that firstly a thorough research was done to find a perfect algorithm for the forecasting of time-series dataframe. LSTM and RNN were chosen as a result, the data was collected from Kraken API, splitting was done to check how the model would working, normalized the data for ease, implemented the algorithm on the data and obtained the desired results.

V. RESULTS AND DISCUSSIONS

Utilizing the collected dataset and after thorough Exploratory Data Analysis that helped us know about all the types of columns and rows available, we were able to eliminate unwanted sections and performed analysis only on the target variables. Therefore, putting LSTM as well as RNN algorithms into use for this kind of forecasting on a time-series dataset on it was a good option. Hence, this paper was successfully able to oversee the bitcoin prices up to 20 minutes so that in near future, traders can invest accordingly and analyse the market trends well in advance.

Through the below pictorial representations, the variations and similarities between the predicted and the real prices of both bid as well as ask price could be seen: -



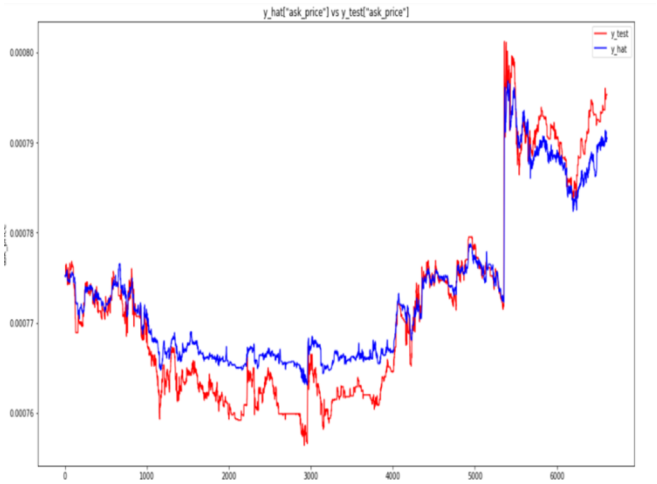


Fig. 10.y_hat vs y_test(ask price) – LSTM and RNN

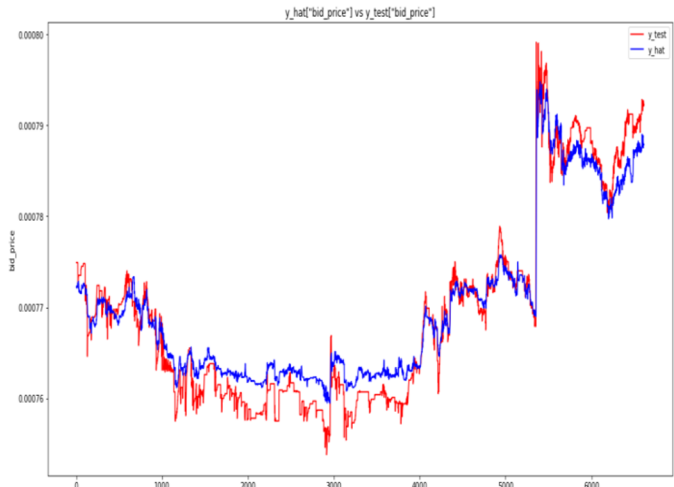


Fig. 11. Y_hat vs y_test(bid price) – LSTM and RNN

As seen in figs. 10 and 11, the anticipated and real values for both the ask price and the bid price match very accurately, however the final results deviate slightly from reality. The four are shown together in Fig.12 as follows:

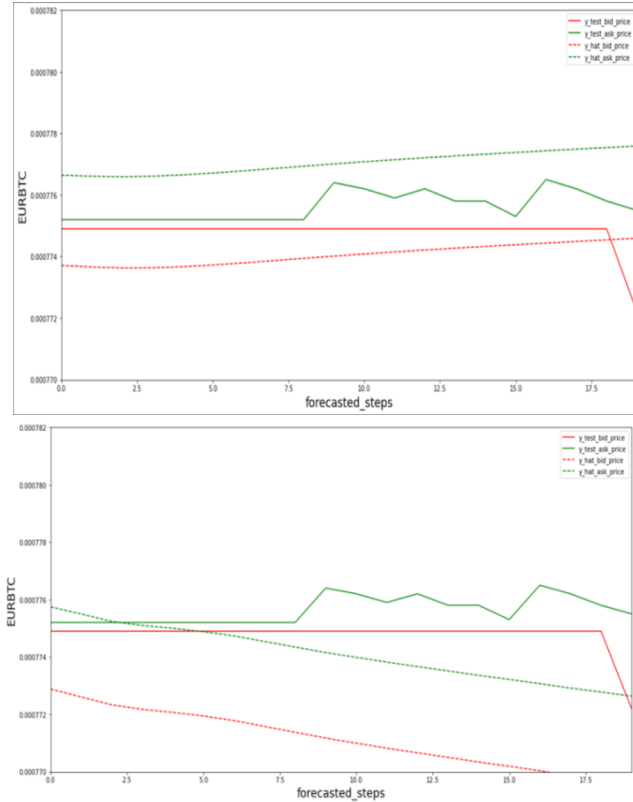


Fig. 12. Forecasted steps for EURBTC- LSTMand RNN

The anticipated actions in relation to EURBTC can be observed in fig. 12. These are the actual and anticipated values, and they are somewhat similar. This is how the LSTM and RNN algorithms allow us to predict values up to 20 minutes in the future.

VI. DISCUSSIONS :-

To carry out the tasks, Python was used. Various packages were imported to clean data, perform required numerical operations, drop unwanted columns and carry out modelling:-

- 1) Numpy:- This library is typically used to make computations, work with numbers, and run statistical analyses.
- 2) Matplotlib:- In order to plot different types of graphs for better understanding of the data, this package is used.
- 3) Seaborn- It is another python library that provides pictorial representations of labeled datasets.

- 4) Panda :- Even when employing data that is not uniform in nature, it helpstoframedata that is not just tabular, nut also labelled.
- 5) MinmaxScaler:- It is employed to standardise data values within a predetermined range.
- 6) Keras :- The python interface for working with neural networks for forecasting is made available as a result.

To obtain required dataset, we resorted to kraken as it is an API which is an open-source tool that provides data for not just bitcoin but other currencies as well, and the dataset utilised in this study is from the kraken exchange. First, the bitcoin dataset's data was divided. For training, we used 80% of it, and for testing, the remaining 20%.

The research produces a graphic depiction of the predicted bitcoin price for the next 20 minutes using matplotlib.

Those who often invest in bitcoin for a little period of time and are not particularly successful over the long term may find this study to be of great use. Also, the databases are currently publicly accessible, and soon they will be a major success when combined with other elements like human sentiments.

VII. VI. FUTURE SCOPE

We may further enhance our model by:

- Increasing the size of the dataset: More historical data can improve the accuracy of our model and help us make more precise predictions.
- Exploring different neural network architectures: While RNN and LSTM are effective in predicting sequential data, there are other neural network architectures that we can explore, such as Convolutional Neural Networks (CNNs) and Transformer models.
- Incorporating external factors: External factors such as market sentiment, news, and social media activity can have a significant impact on Bitcoin prices. Incorporating these factors into our model can provide more comprehensive and accurate predictions.
- Applying our model to other cryptocurrencies: While our research focused on Bitcoin, the same model can be applied to other cryptocurrencies such as Ethereum, Litecoin, and Ripple. This can help us compare the accuracy of our model with other cryptocurrencies and expand our understanding of the crypto market.
- Implementing an optimal ideaset for trading in future: Our methodology can be utilised in order to execute a trading methodology that can help investors make informed decisions about when to buy and sell Bitcoin. Further research can focus on developing such a strategy and back-testing it with historical data.

REFERENCES

1. FOR BITCOIN PRICE PREDICTION, UTILISING AN OPTIMIZED SUPPORT VECTOR MACHINE (SVM) BASED ON PARTICLE SWARM OPTIMIZATION (PSO) FOR. *PROCEDIA COMPUTER SCIENCE*. 2019;163:427–433.
2. Greaves A, Au B. Predict the prices of bitcoin by making use of a transaction graph. 2015
3. Valencia F, Gómez-Espinosa A, Valdés-Aguirre B. Price Movement Prediction of

- Cryptocurrencies Using Sentiment Analysis and Machine Learning. *Entropy*. 2019;21(6):589.
4. Chowdhury R, Rahman MA, Rahman MS, Mahdy MRC. The utilization of machine learning to forecast and predict the price of cryptocurrency constituents and indexes is being pursued as an approach. *Physica A: Statistical Mechanics and its Applications*. 2020;551(124569).
 5. Akyildirim E, Goncu A, Sensoy A. Prediction of cryptocurrency returns using machine learning. *Annals of Operations Research*. 2021;297(1-2):3–36
 6. Mittal R, Arora S, Bhatia MP. Automated prediction of cryptocurrencies using machine learning. *ICTACT Journal on Soft Computing*. 2018;8(4):1758– 1761..
 7. Lipo Wang and Edwin Sin School of Physical and Mathematical Sciences Nanyang Technological University Singapore presented —Bitcoin Price Prediction Using Ensembles of Neural Networks.
 8. Woák K. Carrying out a top-notch social media sentiment analysis for short ü term cryptocurrency price prediction[J]. *Expert Systems*, 2020, 37(2): e12493.
 9. Prajapati P. Predictive analysis of bitcoin price by taking social sentiments[J] into consideration. *arXiv preprint arXiv:2001.10343*, 2020.
 10. Wang Y, Chen R. Cryptocurrency price prediction on the basis of the sentiments of multiple market [C]//*Proceedings of the 53rd Hawaii International Conference on System Sciences*. 2020.
 11. Guégan D, Renault T. Does investor sentiment on social media so as to provide us with robust information forBitcoin returns predictability? [J]. *Finance Research Letters*, 2020: 101494.
 12. Mazur M, Dang M, Vega M. COVID-19 and the march 2020 stock market crash. Evidence from S&P1500[J]. *Finance Research Letters*, 2020: 101690.
 13. Ciner C. Stock marketpredictability when COVID-19 was prevailing[J]. *Finance Research Letters*, 2020: 101705.