Comparative Performance of LSTM and ARIMA for the Short-Term Prediction of Bitcoin Prices

Navmeen Latif[1], Joseph Durai Selvam[2], Manohar Kapse[3], Vinod Sharma[4] and Vaishali Mahajan[5]

Abstract

This research assesses the prediction of Bitcoin prices using the autoregressive integrated moving average (ARIMA) and long-short-term memory (LSTM) models. We forecast the price of Bitcoin for the following day using the static forecast method, with and without re-estimating the forecast model at each step. We take two different training and test samples into consideration for the cross-validation of forecast findings. In the first training sample, ARIMA outperforms LSTM, but in the second training sample, LSTM exceeds ARIMA. Additionally, in the two test-sample forecast periods, LSTM with model re-estimation at each step surpasses ARIMA. Comparing LSTM to ARIMA, the forecasts were much closer to the actual historical prices. As opposed to ARIMA, which could only track the trend of Bitcoin prices, the LSTM model was able to predict both the direction and the value during the specified time period. This research exhibits LSTM's persistent capacity for fluctuating Bitcoin price prediction despite the sophistication of ARIMA.

JEL Classification: C45, C53, O23, O33

Keywords: Bitcoin, ARIMA, LSTM, MAPE

[1] E-mail: navmeen1@gmail.com
[2] School of Business and Management, Christ (Deemed University) University. E-mail: joseph.selvam@christuniversity.in
[3] Symbiosis Center Management and Human Resource Development, Symbiosis International University, Pune, Maharashtra, India. E-mail: mk10oct@gmail.com
[4] Symbiosis Center Management and Human Resource Development, Symbiosis International University, Pune, Maharashtra, India. E-mail: sharmavins@gmail.com
[5] Oracle India Pvt LTD. E-mail: vaishali_mahajan@scmhrd.edu

1. INTRODUCTION

The cryptocurrency market has grown unpredictably and at a phenomenal rate throughout its brief existence. Since the public release of Bitcoin in January 2009, over 8000 cryptocurrencies have been developed as of December 2021, most of which have had limited success. While many of these cryptocurrencies have little to no demand or trading activity, a few have committed groups of backers and investors. Bitcoin, Ethereum, Dogecoin, XRP, and Ripple are popular cryptocurrencies. Except for "asset-backed stable coins" like Tether, cryptocurrencies are not backed by any assets. Cryptocurrencies, unlike financial assets, do not have balance sheets, are not issued by central banks, and are not guaranteed by governments. Cryptocurrency is a very risky investment because of all of these qualities. Cryptocurrencies are highly volatile, and their behavior is complicated and different. The value is determined by demand and supply. Therefore, investing in cryptocurrency necessitates a new set of skills. The cost of mining, competition (competing cryptocurrencies), legislative changes, media coverage/social content, and personage statements are all variables that influence cryptocurrency pricing. On the bright side, there is some evidence that crypto investments are beneficial. El Salvador is the only country globally that has made bitcoin legal tender. Then, for its next executive program on "Economics of Blockchain and Digital Assets," Wharton stated that it would accept tuition payments in cryptocurrencies such as Bitcoin, Ethereum, and USD Coin.

Due to constant advances in the digital environment and widespread promotions of cryptocurrency investment across media platforms, investment in cryptocurrencies has recently gained substantial appeal among Indians. Furthermore, the stock market's post-Covid outstanding performance has attracted a new generation of risk-tolerant investors prepared to play on the crypto market. India has around two crore customers and has nearly 15 local cryptocurrency exchange sites.

The term cryptocurrency has taken the financial world by storm, yet there has been an absence of official and open research on the data of digital assets. Predicting mature financial markets, such as the stock market, has been extensively explored (Kaastra and Boyd,1996). Bitcoin is an intriguing analogy since it is a time series prediction issue in a market that is still in its early stages. Traditional time series prediction approaches, such as the Holt-Winters exponential smoothing models, rely on linear assumptions and need data that may be classified as a trend, seasonal, or noise (Chatfield and Yar, 1988). This technique is better suited for tasks including seasonal impacts, such as sales forecasting. These techniques are ineffective due to the absence of seasonality in the Bitcoin market and its excessive volatility. Given the task's intricacy, deep learning is an intriguing technical option based on its performance in related fields (McNally et al., 2018). Therefore, this paper aims to help financial professionals, students, and cryptocurrency investors in choosing the best price prediction model for the Bitcoin (BTC) crypto coins which have the highest market capitalization as of December 03, 2021 (Weigend, 2018)

The goal of this article is to see how accurately machine learning models can forecast the price of Bitcoin. An ARIMA time series model is constructed for performance comparison purposes with the neural network models to simplify comparison to more traditional methodologies in financial forecasting. The closing price of Bitcoin in USD, as obtained from the Bloomberg Terminal, serves as the independent variable in this study. The study utilizes the root mean squared error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Mean Absolute Deviation (MAD) of the predicted closing price to evaluate model performance. The paper aims to help cryptocurrency investors take a calculated risk and reduce their overall risk exposure. The paper has four major sections – Section 1 gives a brief introduction of the cryptocurrency and time-series prediction, Section 2 discusses the type of cryptocurrency to be predicted, Section 3 represents the Machine Learning – Neural Networks algorithms to be used for forecasting – ARIMA and Deep learning (LSTM) and lastly, Section 4 explains the results and comparative analysis of forecasting models using RMSE (Root Mean Squared Error), MAPE (Mean Absolute Percentage Error), MAE and MAD.

2. LITERATURE REVIEW

2.1. Cryptocurrency & Bitcoin

Although the notion of electronic currency dates back to the late 1980s, Bitcoin, introduced in 2009 by a pseudonymous (and still mysterious) creator named Satoshi Nakamoto, is the first successful decentralized cryptocurrency (Satoshi Nakamoto., 2008). Bitcoin, as a currency, presents a distinct potential for price prediction due to its young age and accompanying volatility, which is significantly greater than that of traditional currencies (M. Briere et al., 2013). It is also distinct from typical fiat currencies in terms of its open nature; no comprehensive data on cash transactions or money in circulation for fiat currencies is available. In a nutshell, cryptocurrency is a virtual coinage system that works similarly to a regular currency, allowing users to offer virtual payment for products and services without needing a central trusted authority. Cryptocurrencies rely on digital data transfer, with cryptographic methods used to assure authentic, one-of-a-kind transactions. Bitcoin advanced the digital coin market by decentralizing money and liberating it from hierarchical power structures. Individuals and companies instead deal with the currency electronically through a peer-to-peer network. In 2011, Bitcoin drew much attention, including several altcoins — a catch-all term for all cryptocurrencies created after Bitcoin – sprung up. The market capitalization of the cryptocurrency business fluctuates drastically due to extreme volatility. Still, it is estimated to be slightly over $2.02 trillion at the time of this research, with Bitcoin accounting for around 45 percent of the market value.

Being the coin that heralded the cryptocurrency age, Bitcoin is still the coin that most people think of when discussing digital currency. The currency's mysterious creator, purportedly Satoshi Nakamoto, unveiled it in 2009, and it's been on a roller-coaster ride ever since. However, it wasn't until 2017 that bitcoin entered the public mind.

2.2. Time-Series Forecasting

Time series forecasting is a method for predicting future occurrences by evaluating previous patterns and assuming that future trends would be similar to previous trends. Forecasting is the process of predicting future values by fitting models to the last data. Time series forecasting is a data-driven method of effective and efficient planning used to solve prediction issues with a time component. Time series models have various applications, from sales forecasting to weather forecasting. Time series models are one of the most successful techniques for predicting situations when there is a degree of uncertainty about the future. Time series forecasting aims to anticipate a future value or categorization at a certain time. (Dotis-Georgiou, 2021)

2.3. ARIMA Model

ARIMA models have demonstrated their capacity to deliver accurate short-term projections. In short-term prediction, it consistently outperformed complicated structural mod- els (A. Meyler et al., 1988). The future value of a variable in an ARIMA model is a linear mixture of previous values and past errors.

In 1970, Box and Jenkins introduced the ARIMA model. It's also known as the Box- Jenkins approach, which consists of a sequence of steps for detecting, estimating, and diagnosing ARIMA models with time-series data. The model is one of the most widely used techniques in financial forecasting. (P. Pai and C. Lin, 2005) (N. Rangan and N. Titida, 2009) (Merhet et al., 2010) ARIMA models, or autoregressive integrated moving average models, are another time series forecasting technique. Autocorrelation is used in ARIMA models to provide predictions. When a time series has autocorrelation, there is a correlation between the time series and a lagged version of the time series. Auto regression is a time series model that predicts the value at the next step by using data from prior time steps as input to a regression equation. In an autoregressive model, the predictions are a linear mixture of the variable's historical values. Because

ARIMA models need stationary time series, differencing may be required before employing an ARIMA model for forecasting. (Dotis-Georgiou, 2021)

2.4. Deep Learning Models

Deep Learning is mainly used to achieve the most precise outcomes throughout several phases (Galleria et. al, 2014). When modules are placed on top of each other, the models described in this section apply a nonlinear function to the hidden units, allowing for a more lavish model capable of learning more complex pictures to build a deep network (Loutfi et al., 2014). Deep learning aims to create structures at the lower layers that segregate the various components in the input data and chain the representations at the higher layers. However, the disadvantage of training with many hidden layer units is that the error signal is back-propagated.

A long short-term memory network (LSTM) is a form of RNN that is particularly popular in time series analysis. It uses forget gates and feed-forward techniques to store knowledge, forget unnecessary inputs, and update the forecasting algorithm to model and forecast complicated time series issues. The Long Short-Term Memory (LSTM) network will be used in this study as a Deep Learning model. They are distinct from RNN Models in that, in addition to having a memory, they can pick which data to remember and which data to forget based on the weight and value of that feature. J. Schmidhuber et al. in 2001 constructed an LSTM for a time series prediction job and discovered that the LSTM performed as well as the RNN. This sort of model is also used here. The substantial computation involved in training LSTM is one constraint.

Lu et al. (2018) suggested a novel forecasting framework based on the LSTM model to anticipate the daily price of bitcoin using two different LSTM models (standard LSTM model and LSTM with AR (2) model). The suggested models' performance was examined using daily bitcoin price data from 2018/1/1 to 2018/7/28, totaling 208 records. The findings supported the suggested model's outstanding predicting accuracy with AR (2).

Karakoyun et al. in 2019 performed a study on the daily prices (2013-2018) of bitcoin to compare ARIMA and LSTM results and found that LSTM outperformed ARIMA in predicting the Bitcoin prices for the next 30 days. LSTM had a MAPE equal to 1.40% and ARIMA equal to 11.86%. The study results conducted by Gadosey et al. in 2019 reveal that the ARIMA model outperformed deep learning-based regression approaches. ARIMA produces the greatest results, with MAPE and RMSE of 2.76 percent and 302.53 percent, respectively. The Gated Recurrent Unit (GRU) outperformed the Long Short- term Memory (LSTM) with 3.97 percent MAPE and 381.34 RMSE, respectively. The data was again daily trading prices between 2014 to 2019.

3. METHODOLOGY

For the research, Bitcoin (BTC) historical Closing Prices for a range of nearly one year, beginning from 12/21/2020 0:00 to 12/21/2021 16:00 (MM/DD/YYYY HH/MM) with a frequency or period of 10 minutes is collected from Bloomberg Terminal. The data is pre-processed and checked for any missing values. The same dataset is used for both ARIMA and RNN-LSTM Models. The models are designed to predict Bitcoin Closing Prices for the next 261 periods in the future. Each period is equal to 10 minutes. The data points are exactly equivalent to 52631. The target variable in this research is only the Close Prices of Bitcoin. In this research. LSTM and ARIMA Models are built and evaluated based on prediction accuracy.

The main purpose of this study is to get more accurate predictions and compare the above-mentioned models. The whole dataset is divided into two datasets for the training of the models, i.e., the training dataset and the testing dataset. Since it is a time-series-based dataset, the samples are not chosen randomly, but a split percentage of 99.5% is used. Therefore, in simple terms, 99.5% of the dataset is used for training the model, and 0.5% is for the testing model, for which the models are made to predict. For the study, we had chosen a smaller test dataset because the goal was to build a short-term prediction model and check the accuracy of its predictions in the short term. The models were built to be useful for day traders, which is why we included nearly 200 future prediction points.

Training Data for both the models starts from 12/21/2020 0:00 to 12/19/2021 20:30 (using a 24-hour time format and date format is MM/DD/YYYY), and the testing data for both the model starts from 12/19/2021 20:40 to 12/21/2021 16:10.

The study evaluates the models for short-term prediction, which is nearly 20 hours, with predictions every 10 minutes.

TABLE 1
Testing and Training Dataset

TRAINING DATASET			**TESTING DATASET**		
From	To	Data points	From	To	Data points
12/21/2020 0:00	12/19/2021 20:30	52370	12/19/2021 20:40	12/21/2021 16:10	261

The framework of the proposed methodology is as follows:

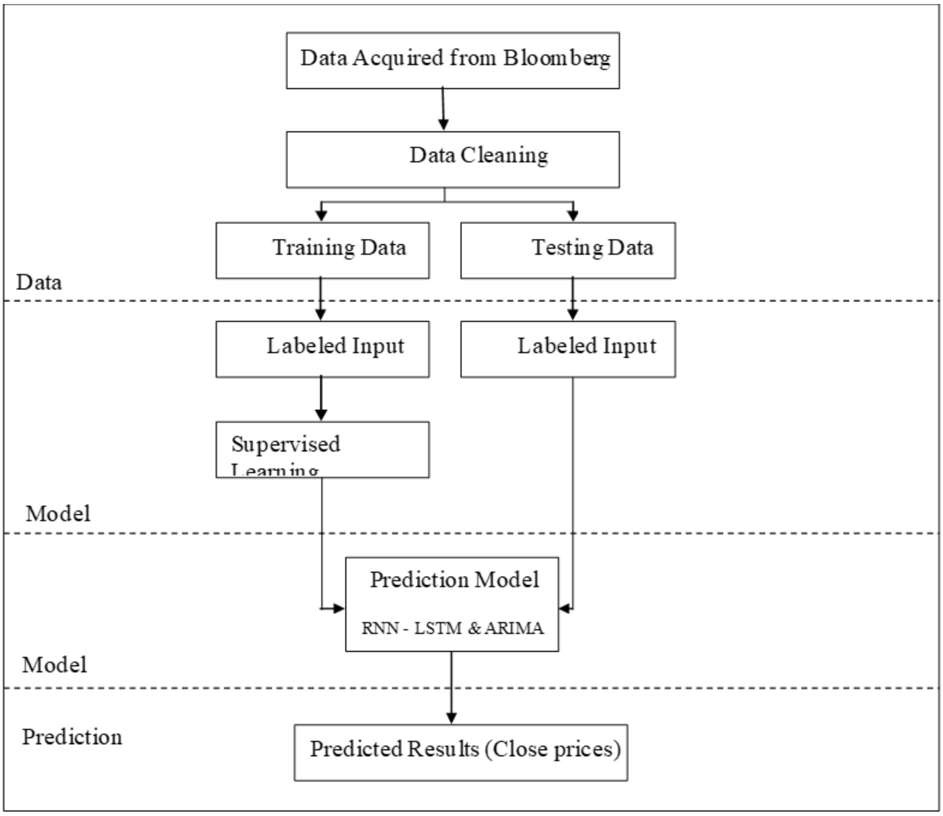

FIGURE 1: Proposed Methodology

3.1. Technology Used

The prediction models are coded in Python Programming Language. Python has been around since 1990 and is now the most popular programming language for Machine Learning research and development. It has a basic syntax that is easy to understand. Python is also a very efficient programming language. Developers can choose from hundreds of different libraries and frameworks. Some of the basic libraries used for this study are Keras, Tensor Flow, Scikit-learn, Numpy, Pandas, Seaborn, and Pmdarima.

The models were purely built on Google Colab Free version, a platform providing access to virtual RAM and Disk space for running huge codes. The jupyter notebook linked to the python kernel on Google Colab is connected to virtual machines that allow executing codes over it. The execution time required for building both models is nearly 4 hours which could vary as per the data size. Google Colab provided a virtual RAM space of 12.69 GB.

Prediction Models Used

3.2. ARIMA – Auto-Regressive Integrated Moving Average

The Auto-Regressive Integrated Moving Average (ARIMA) model is a well-known and commonly used time-series forecasting tool. In time-series data, ARIMA models may capture a variety of distinct typical temporal patterns.

- AR: Auto-Regressive denotes that the model relies on a dependent relationship be- tween an observation and a set of lagged data (sometimes called "time lag" or "lag").
- I: To make the time-series stationery, the model uses differencing of raw observations
- MA: indicates that the model uses the link between residual error and data.
- The standard ARIMA model is built on input parameters with 3 arguments, i.e., p,d,q.
- The number of lag observations is given by p.
- The letter d denotes the degree of differentiation.
- The moving average window's size/width is q.

The above parameters are chosen by certain methods. Examining the autocorrelation function and partial autocorrelation function is an effective and intuitive way to estimate the terms for autoregressive models, i.e., p and q. The value of d is determined by per- forming Augmented Dickey-Fuller Test or simply Adfuller Test and is also confirmed by running a python code on the entire data set. The hypothesis for the Adfuller test is,

H_{01}: A unit root is present in the time series
sample. H_{02}: The data series is not stationary.

After performing the above test and checking the p-value, if there is a need, then the data is the difference is applied on the data series either once or maximum twice to make it stationary as the basic assumption for ARIMA is that the data is stationary, which is not true in the case of Stock prices or Bitcoin prices. The framework for ARIMA Model is as follows:
For a time series analysis of future price predictions, Autoregressive integrated moving average (ARIMA) models are a common choice for forecasting over a short-term condition; they operate when data has a consistent or stable pattern (constant) across time with the fewest probable outliers. However, this does not always work in real-time, where data fluctuates dramatically and is very volatile. (C. Scheier and W. Tschacher , 1996)

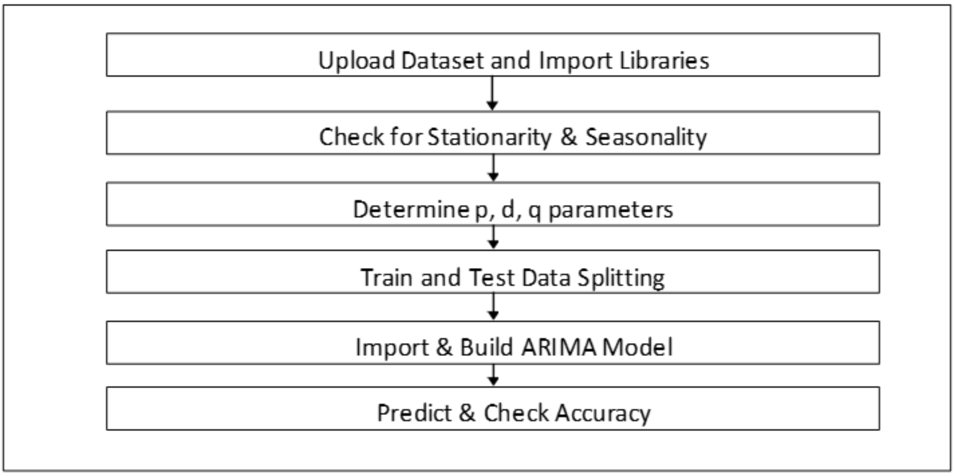

FIGURE 2: ARIMA forecasting framework

3.3. RNN - LSTM – Long Short-Term Memory

One of the modules available for Recurrent Neural networks is LSTM (Long Short Term Memory). Hochreiter and Schmidhuber (1997) invented LSTM, which was further re- fined and popularized by a number of researchers. Long short-term memory (LSTM) is a deep learning architecture that uses an artificial recurrent neural network (RNN). LSTM features feedback connections, unlike normal feed-forward neural networks. It can handle not just single data points but also complete data sequences. The LSTM network (LSTM network) consists of recurrently consistent modules, just like the RNN. The link between the hidden layers of RNN is different in LSTM, which is an upgraded form of RNN.

LSTMs are commonly employed for sequence prediction tasks and have shown to be quite successful. Sequence prediction is a problem that involves predicting the next value or values in a sequence using historical sequence information. They operate so effectively because LSTM can remember important information from the past while forgetting less critical information.

When dealing with Time-Series or Sequential Data, this feature is incredibly beneficial. When using an LSTM model, we have complete control over what information is saved and discarded. The "gates" are used to do this. There are three gates in the LSTM:

- The input gate is a device that contributes information to the state of a cell.
- The forget gate eliminates information from the model that is no longer needed.
- The LSTM output gate selects the information to be displayed as output.

Because LSTM is sensitive to data scale, the min-max scalar translates values from 0 to 1. Apart from that, the number of LSTM layers, the dropout value, and the number of epochs are all variables that may be changed in the LSTM model. The python libraries used for LSTM are Keras, Tensorflow, Pandas & Numpy, and Scikit – Learn.
The framework for RNN - LSTM Model is as follows:

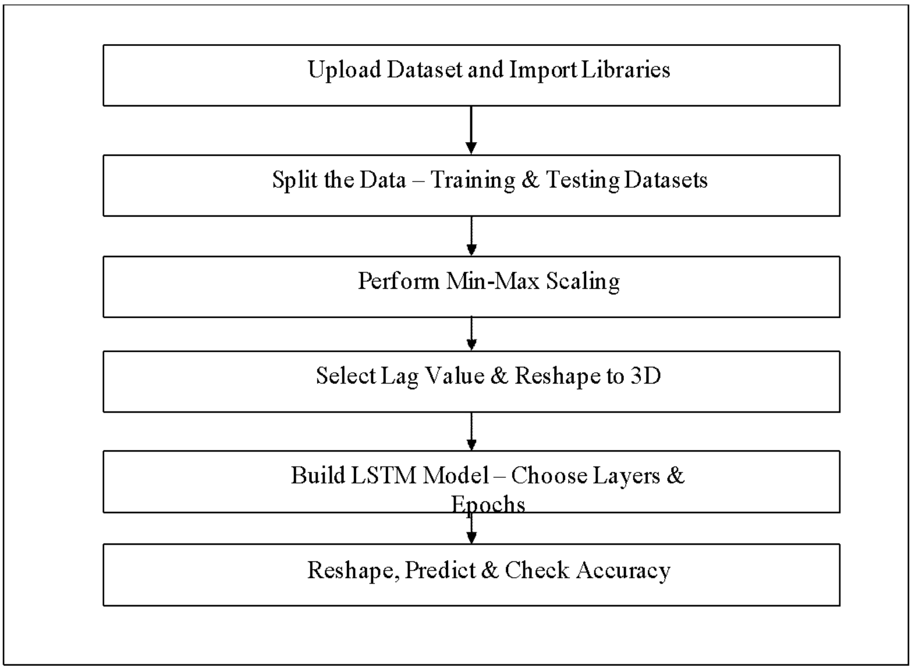

FIGURE 3: LSTM forecasting framework

3.4. Performance Measurement Methods

The results from both models are evaluated using the following metrics MAPE Forecast Accuracy (Mean Absolute Percentage Error), MAE Forecast Accuracy (Mean Absolute Error) or MAD (Mean Absolute Deviation) or WAPE (Weighted Absolute Percentage Deviation), and RMSE (Root Mean Squared Error)

4. IMPLEMENTATION & RESULTS

Before running the models, the first step is to upload the dataset on Google Colab and visualize the Bitcoin historical Close Prices. The following plot was generated using python code on the Colab, which displays the closing prices before the data split into testing and training data. It starts from 12/21/2020 0:00, and the last recorded closing price for this study is 12/21/2021 16:10. The y-axis of the graph represents the Close Prices of Bitcoin in USD ($), and the x-axis represents the DateTime. The only variable used in this study for both models is Historical Closing Prices, and the same is also being predicted.

ARIMA Results

The primary assumption of the ARIMA Model is that the time series should be station- ary, i.e., the properties should not depend on the time at which it is observed. Therefore, the time series need to be free from trends or seasonality. So, the first step is to perform the Augmented Dickey- Fuller Test (ADF Test), and in this case, the results are as follows:

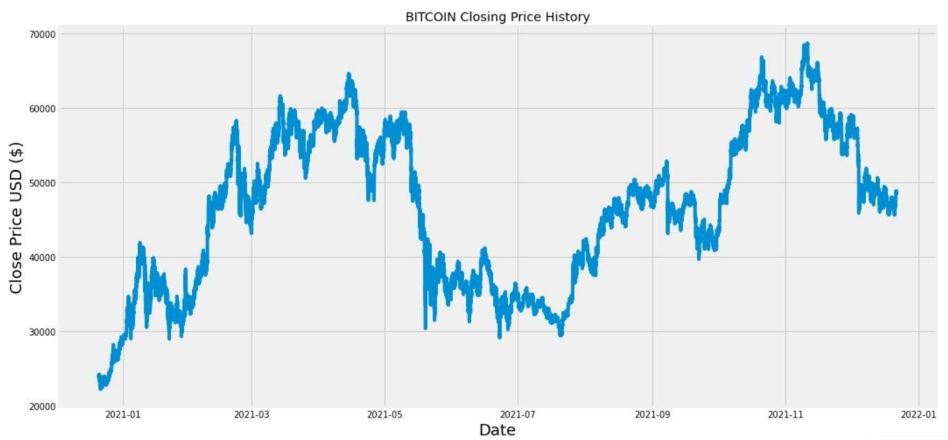

FIGURE 4: Historical Bitcoin Closing Prices ($)

```
from statsmodels.tsa.stattools import adfuller
result = adfuller(df.Close_.dropna())
print(f"ADF Statistic: {result[0]}")
print(f"p-value: {result[1]}")
```

```
ADF Statistic: -2.4157333739357947
p-value: 0.1373449305920295
```

FIGURE 5: ADFC Test Results

From the results shown above, it is clear that the price time series is not stationary as the p-value is higher than the significance value (α), equal to 5%. The results are quite realistic as most of the price time series are non-stationary. Otherwise, the investors would always gain by using buying low and selling high strategies.

$$0.137 > 0.05$$

Now, since the data is non-stationary, it needs to be differenced to make it stationary, eliminate any trends or seasonality, and make the ARIMA Model work. The returns are computed as they are usually randomly distributed

around a zero mean. In simple terms, the current value is subtracted from the previous value. If the time series is already stationary, d will be equal to 0, but for this data, d can either be 1 or 2.
Two methods are used to find the order of differencing. One is done manually by plotting ACF (Autocorrelation Function) plots. It tells us how many terms any required to remove any autocorrelation in the series.

The following graph shows the autocorrelation

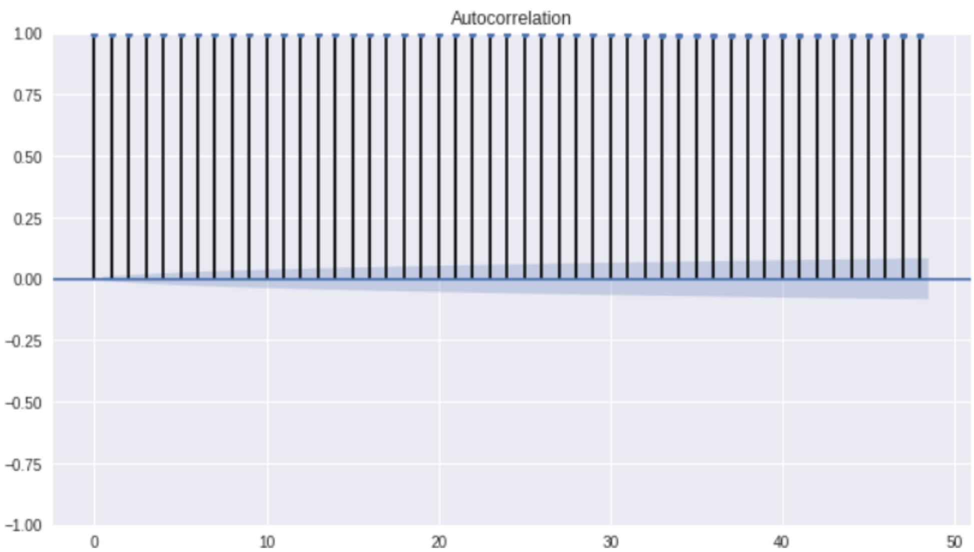

FIGURE 6: ACF Original Prices

As the time series is differenced once, we can see that the returns are randomly distributed around the zero mean, and the autocorrelation plot looks different.

To ensure the differencing value, the returns of returns are obtained and plotted. In simple terms, the data is differenced twice, and it can be observed that there is not much difference in the ACF plots, and both graphs look somewhat similar.

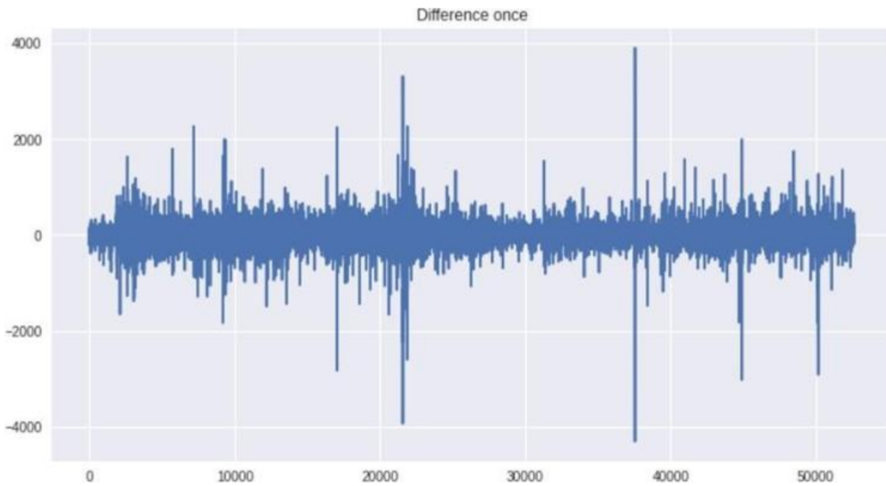

FIGURE 7: Differenced Once – Returns

From the above results, it is clear that the time series should be differenced once, but to further get clarity, the inbuilt ADF test is run in python, which gives the order of differencing by examining the whole dataset.

The next step in constructing the ARIMA Model is to find the number of lags or lag order (p). The AR terms are estimated by inspecting the Partial Autocorrelation (PACF).

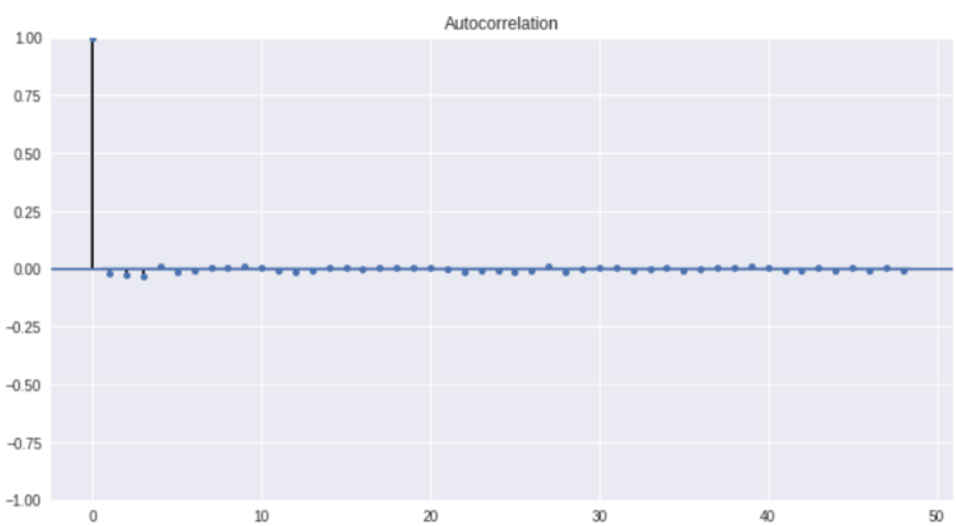

FIGURE 8: ACFF when differenced once

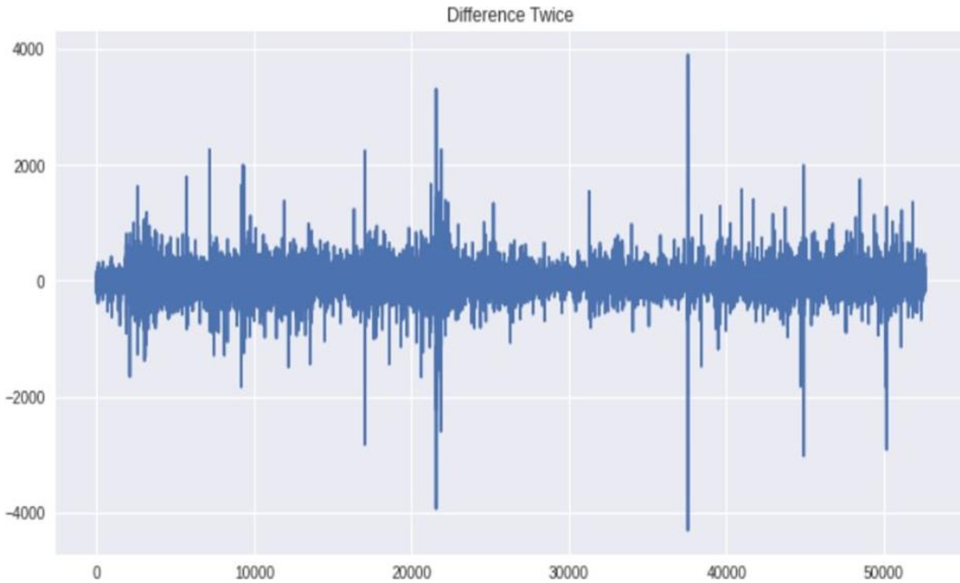

FIGURE 9: Differenced Twice - Returns of Returns

From the above PACF, it is clear that lag 3 stands out and is significant. Therefore, 3 lags are used as predictors. In simple terms, the p value for constructing the ARIMA Model equals 3.

The above graph shows that forecast error 3 can be suitable for the model as it stands out from the confidence interval indicated by a light blue shaded area. Therefore, the MA order for constructing the ARIMA Model is 3.

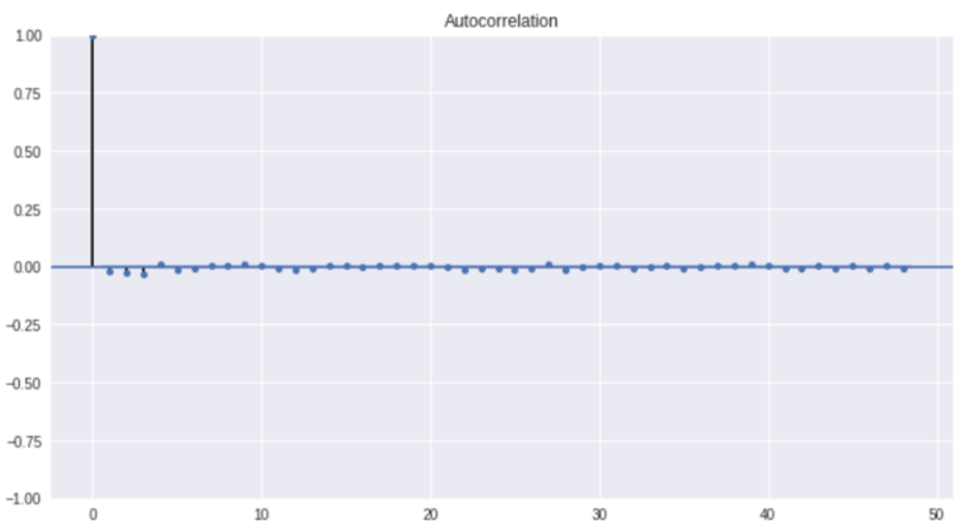

FIGURE 10: ACF when differenced twice

The ARIMA (3,1,3) model is built and trained on the training dataset based on the above-estimated parameters, and the summary of the same is printed. From the model summary, 3 AR terms and 3 MA terms, and one constant term exist. The coefficients for the linear regression are not very close to zero, so we are keeping all the terms presented above. Also, the associated p-values need to be very low to prove that these terms are important in the regression model, which is true as all the p-values are nearly equal to zero.

The below graph shows the predictions made by ARIMA (3,1,3) against the actual historical closing prices of Bitcoin. The projected graph seems to be holding up to the trend. However, it fails to predict the direction of the price but still is able to show the trend correctly.

4.1. LSTM RESULT

This model uses the artificial recurrent neural network to predict the BTC closing prices using the past 200 historical BTC prices. After data visualization and conversion into an array, the data is split into training and testing datasets, with a training dataset containing 99.5% of the total closing prices. The data is then scaled down before introducing it into the neural network. The data is scaled between a range of 0 to 1. The scaled training data is further divided into independent training datasets (variable) and dependent training datasets (variable). Using the past 200 historical prices, arrays are created. These arrays are then transformed into a NumPy array to feed them into the LSTM Model.

It is important to reshape the data that is being fed into the LSTM Model as it expects the data to be three-dimensional, in the form of the number of samples, number of time steps, and number of features. For this model,

- number of samples = 52370

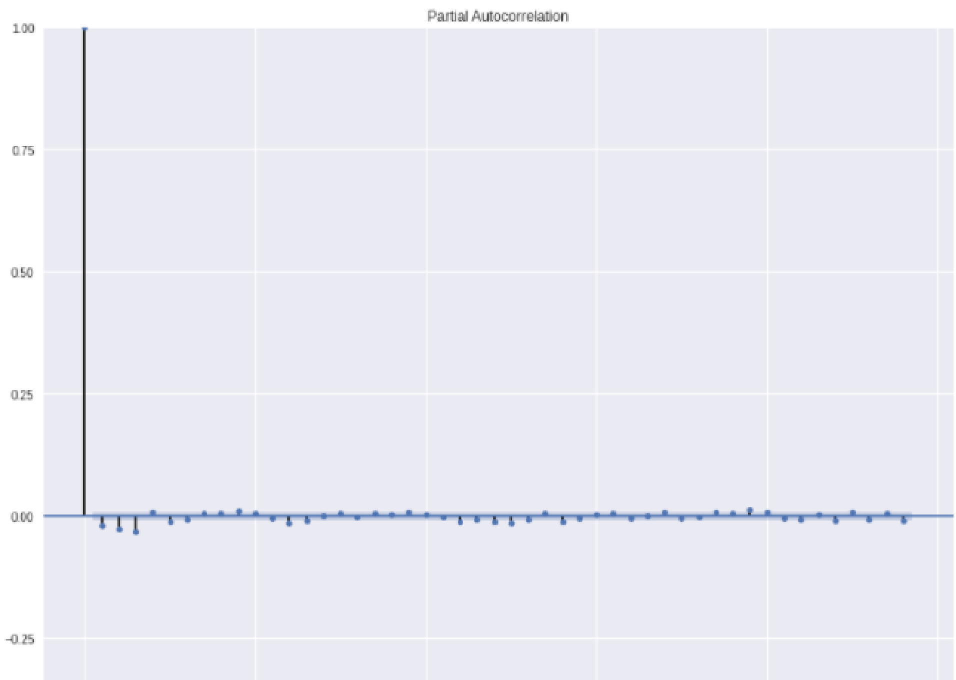

FIGURE 11: 8 : PACF Plot

- number of time steps =200
- number of features = 1 i.e.,closing prices

adamoss function as mean squared error. The optimizer improves upon the loss func- tion, and the loss function tells us how well the model did on training. The model is then trained with batch size equal to 1 and epoch set as 1. Epoch is the number of iterations.

The model is then tested using the scaled testing dataset, and the following results are achieved.

The above graph represents the training dataset used and the validating or testing dataset against the predictions made by LSTM. A closer view of the predictions is shown in the graph below:

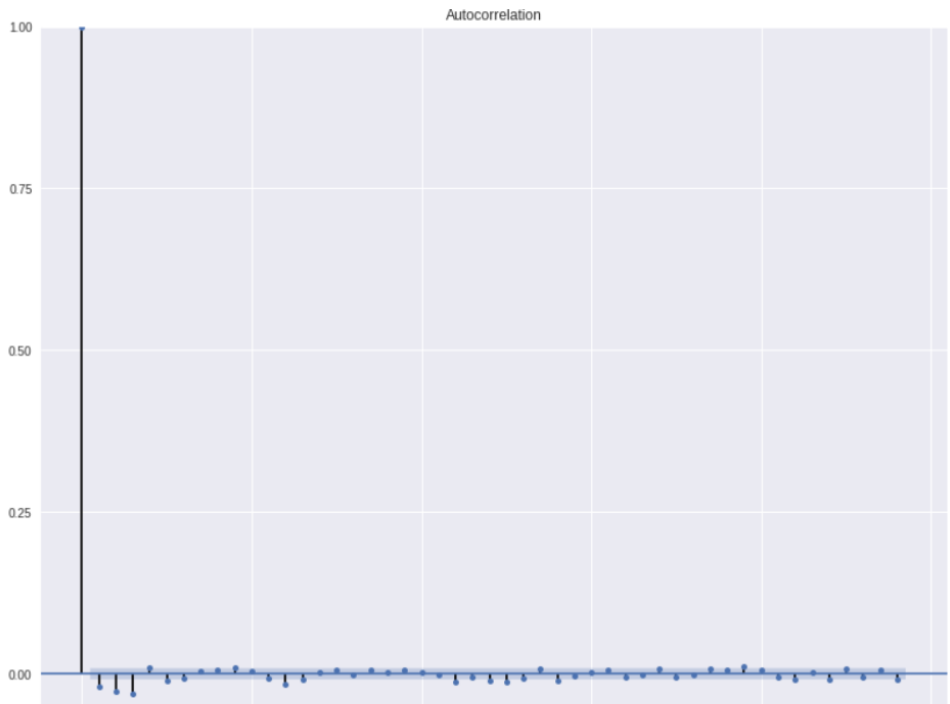

FIGURE 12

FIGURE 13

```
model.fit(x_train,y_train,batch_size=1 , epochs = 1)

52170/52170 [==============================] - 3321s 64ms/step - loss: 1.4531e-04
<keras.callbacks.History at 0x7f55f86d1110>
```

FIGURE 14

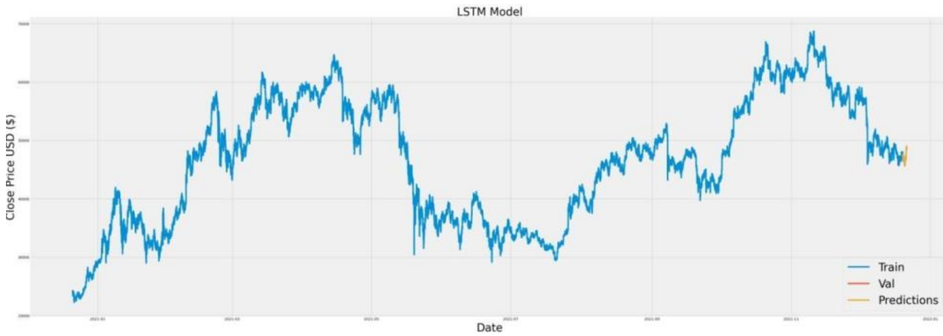

FIGURE 15:

LSTM - Actual Vs Predicted

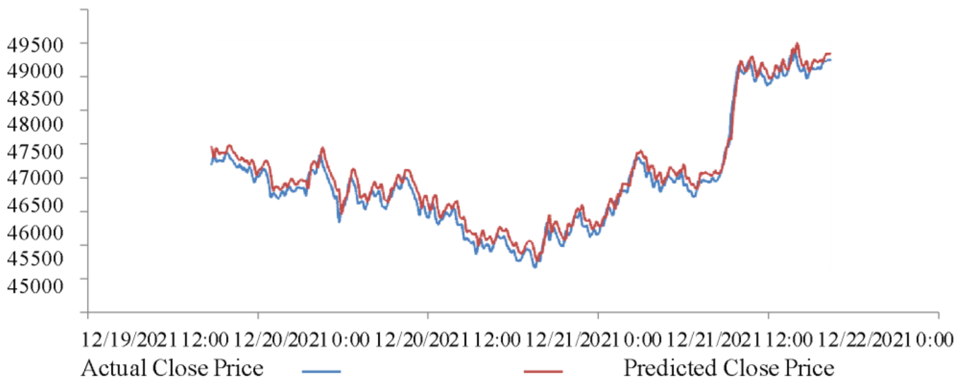

FIGURE 16: Actual vs. Predicted Prices of LSTM

It can be observed from the above graph that the LSTM model is quite accurate, not only predicting the trend but also following the direction of close prices.

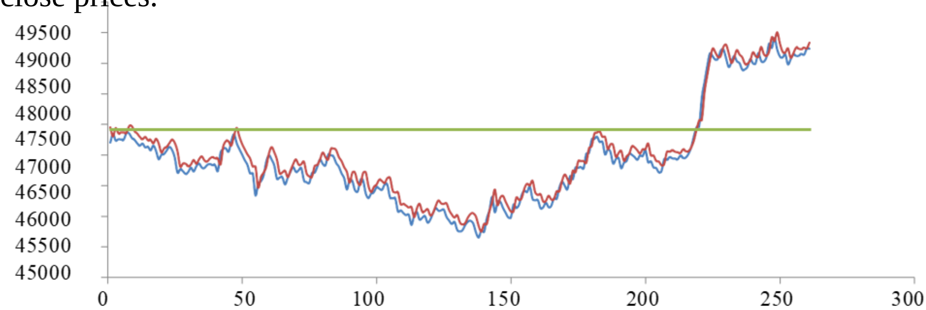

FIGURE 17: ARIMA vs. LSTM vs. Actual Prices of Bitcoin in USD EVALUATION

As can be seen from the graph below, LSTM can produce predictions close to the actual historical prices than ARIMA.LSTM model is able to predict the direction as well as the value in the given time period, while as ARIMA is only keeping up with the trend of the Bitcoin Prices only, it is not able to predict very closely to the actual prices. However, each model seems to have done a good job as they are evaluated using the above-discussed performance measurement metrics. The graph's Y-axis shows the Bitcoin prices in USD, and the x-axis represents the point in time. Both the models have predicted for 261 periods in the future with a gap of 10 minutes in each period. The models are suitable for making short-term predictions.

The table below shows the calculated performance measurement metrics of both models:

TABLE 2

Performance Measurement Metrics

	MAE	**MAPE**	**RMSE**	**Accuracy**
ARIMA (3,1,3)	837.77	1.79%	940.40	98.21%
RNN-LSTM	126.97	0.27%	151.95	99.73%

The above metrics were calculated both manually in excel as well as by executing the python code on the forecasted values. The above table shows that the LSTM model has performed better than ARIMA, but ARIMA is also giving good results with accuracy equal to 98.21%. With the ARIMA model, investors, on an average, can expect an error equal to $837.77 as per MAE and adjusting for the large rare errors, an error equal to $940.40 on average is expected from the ARIMA model.

On the other hand, from the forecasts of the LSTM Model, the error is minimal compared to the actual prices. An error equal to $126.97 is expected from the LSTM model on a more frequent basis, but an error equal to $151.95 is expected from this model, which could be infrequent in nature and is still very small. The overall prediction accuracy of the LSTM model is equal to 99.73%.

5. CONCLUSION

RNN and LSTM deep learning models are clearly useful for Bitcoin prediction, with the LSTM being better at recognizing longer-term relationships. Despite the precision measures indicating acceptable performance, the ARIMA forecast based on error performed much worse than the neural network model. The ARIMA model was able to forecast successfully because the Bitcoin trend is upward, which is what ARIMA learned; if the trend were downward, ARIMA would not be able to predict. The results demonstrate that LSTM outperformed other ARIMA with a 99.73 percent accuracy. It has been established that the quality of the training data and the size of the dataset population are critical for a good prediction. As a result, the LSTM is regarded as a credible cryptocurrency forecasting

model.
The challenge of this research lies in the computational power required, the high ram space, and efficient processors. The original dataset chosen consisted of nearly one lakh data points but had to be reduced later due to the unavailability of resources. The algorithm's accuracy rate for forecasted prices will be enhanced in the future. Furthermore, further work will be done to improve the LSTM utilizing minute-to-minute data to acquire the most accurate result per real Bitcoin value.

References

Briere, K., Oosterlinck, A. and Szafarz (2013), "Virtual currency, tangible ' return: Portfolio diversification with bitcoins", Tangible Return: Portfolio Diversification with Bitcoins.
https://doi.org/10.2139/ssrn.2324780

Chatfield, M. and Yar (1988), "Holt-winters forecasting: some practical issues", The Statistician, pp. 129-140.
https://doi.org/10.2307/2348687

Dotis-Georgiou, A. (2021), URL https://www.infoworld.com/article/3622246/an- introduction-to-time-series-forecasting.html.

Gers, F.A., Eck, D. and Schmidhuber, J. (2001). Info, B. (), URL https://blockchain.info/.

Kaastra, M. and Boyd (1996), "Designing a neural network for forecasting financial and economic time series", Neurocomputing, Vol. 10 No. 3, pp. 215-236.
https://doi.org/10.1016/0925-2312(95)00039-9

Karakoyun, E.S. and Cibikdiken, A.O. (2018), "Comparison of arima time series model and lstm deep learning algorithm for bitcoin price forecasting", The 13th multidisci- plinary academic conference in Prague, Vol. 2018, pp. 171-180.

Ã Karlsson, L., Loutfi, L. and A (2014), "A review of unsupervised feature learning and deep learning for time-series modeling", Pattern Recognition Letters, Vol. 42 No. C, pp. 11-24.
https://doi.org/10.1016/j.patrec.2014.01.008

Mcnally, S., Roche, J. and Caton, S. (2018), "Predicting the price of bitcoin using ma- chine learning", 26th euromicro international conference on parallel, distributed and network-based processing (PDP), pp. 339-343. https://doi.org/10.1109/PDP2018.2018.00060

Merh, N., Saxena, V.P. and Pardasani, K.R. (2010), "A Comparison Between Hybrid Approaches of ANN and ARIMA For Indian Stock Trend Forecasting"", Journal of Business Intelligence, Vol. 3 No. 2, pp. 23-43.

Meyler, G., Kenny, T. and Quinn (1998).

Nakamoto, S. (2008), URL https://bitcoin.org/bitcoin.pdf. Pai, P. and Lin, C. (2005).