Predicting bitcoin price using LSTM and compare its predictability with ARIMA model

1Anshul Saxena, 2T.R. Sukumar

1Assistant Professor, Department of Management, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India
2Associate Professor, Department of Mechanical Engineering, Kumaraguru College of Technology, Saravanampatty, Coimbatore: 49, Tamil Nadu, India.

1anshulsaxena@kctbs.ac.in

Abstract: Due to the difficulty in assessing the exact nature of a time series, it is often considerably challenging to generate appropriate forecasts. Over the years, various forecasting models have been developed in the literature, but they have produced minimum accuracy in forecasting the bitcoin price. The study involves the time series forecasting of the bitcoin prices with improved efficiency using long short-term memory techniques (LSTM) and compares its predictability with the traditional method (ARIMA). The RMSE of ARIMA Model is 700.69 whereas for the LSTM is 456.78 which proves that tradition (ARIMA) model outperforms the machine learning algorithms in our case LSTM model.

Keywords: Time-series forecasting, Bitcoin, LSTM, ARIMA, Prediction, RMSE, Keras, Tensorflow etc.

1. INTRODUCTION

Bitcoin is a digital currency and an advanced kind of payment network. Bitcoin applies peer-to-peer technology to work with no central authority or banks, handling transactions and the issuing of bitcoin are carried out collectively by the same. Bitcoin is open-source; its design is public, nobody owns or controls Bitcoin and everyone can hold a part of it. Through many of its special and different properties, Bitcoin allows exciting uses that could not be covered by any other payment system.1

Bitcoin market is a highly volatile market working 24/7. It is the leading cryptocurrency because of its open nature and high adaptability. Time series forecasting helps to develop a mathematical model that can forecast future observations on the basis of available data. Due to the difficulty in assessing the exact nature of a time series,

1https://bitcoin.org/en/
it is often considerably challenging to generate appropriate forecasts. Over the years, various forecasting models have been developed in the literature, but they have produced minimum accuracy in forecasting the bitcoin price. The study involves the time series forecasting of the bitcoin prices with improved efficiency using long short-term memory techniques (LSTM) and compares its predictability with the traditional model (ARIMA).

Objective of the study involves

- To forecast the bitcoin price with improved efficiency using long short-term memory techniques (LSTM).
- To compare it predictability with the ARIMA model

2. RELATED WORK

Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010)² conducted a large-scale comparative study on various machine learning models for time series forecasting. The forecasting results are compared with those of ARIMA, ANN, and Zhang’s hybrid models which comparatively achieved the best forecasting accuracies. Greaves, A., & Au, B. (2015)³ have investigated the predictive power of blockchain network-based features on the future price of bitcoin. As a result of blockchain network-based feature engineering and machine learning optimization, they obtained Bitcoin price movement classification accuracy of roughly 55%. Khashei, M., & Bijari, M. (2010)⁴ also proposed a hybrid model of artificial neural networks is proposed using auto-regressive integrated moving average (ARIMA) models in order to yield a more accurate forecasting model than artificial neural networks. Bakar, N. A., & Rosbi, S.⁵ worked on a study to forecast the Bitcoin exchange rate using the weighted moving average method. Then, the validity of the forecasting model is validated using mean absolute percentage error (MAPE) calculation. Results indicated mean absolute percentage error is 0.72%.

3. RESEARCH METHODOLOGY

3.1 CRISP-DM

The project follows Cross Industry Standard Process for Data Mining (CRISP-DM) methodology\(^6\). This is an industry-proven technique for the data mining effort. The life cycle model has six phases with arrows indicating the most important and frequent dependencies between phases. The sequence of the phases is considered to be easy. In fact, most projects move back and forth between phases as needed.

The data mining life cycle\(^7\)

![Data Mining Life Cycle Diagram](https://www.ibm.com/support/knowledgecenter/SS3RA7_18.1.1/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html)

In such a situation, the modeling, evaluation, and deployment phases might be less related to the data understanding and preparation phases. However, it is still important to consider some of the queries raised during these later phases for long-term planning and future data mining goals.

The CRISP-DM model provides an opportunity to modify the approach easily that makes it more adaptable. For instance, if your association intends to discover illegal tax avoidance, it is inclined that you will filter through various channels for guaranteeing data without a specific exhibiting objective. Rather than displaying, your work will focus on data examination and perception that will reveal suspicious cases over monetary data. CRISP-DM permits you to make an information mining model that fits your specific necessities.

3.2 RNN

Recurrent Neural Networks developed by Elman have been recently acquiring popularity in the networks designs and increased computational power from graphical processing units. They are mainly useful with sequential data (in our case the time series data of bitcoins) because each neuron or unit can access its internal memory to maintain information about the previous input.

Simple RNN Structure\(^8\)


\(^7\)https://www.ibm.com/support/knowledgecenter/SS3RA7_18.1.1/modeler_crispdm_ddita/clementine/crisp_help/crisp_overview.html

\(^8\)http://colah.github.io/posts/2015-08-Understanding-LSTMs/
One limitation of RNN is that they are influenced by the vanishing gradient issue. This issue is that as layers and time steps of the network relate to each other they are susceptible to exploding or vanishing gradients. Vanishing gradients are considered to be an issue as they can turn out to be too little for the system to learn while inclinations can be constrained utilizing regularization. Furthermore, some examination has discovered that while RNN is capable of taking care of long-term dependencies, in practice they often fail to learn due to the difficulties between gradient succession and long-term dependencies.

### 3.3 LSTM

The Long Short-Term Memory network or LSTM address the vanishing gradient problem common in the recurring neural network. This is a type of recurrent neural network used in deep learning because very large architectures can be successfully trained. LSTM allow the network to continue to learn over many time steps by upholding a more constant error. This allows the network to learn long-term reliance. An LSTM cell contains forget and remember gates which allow the cell to decide what information to block or pass based on its strength and importance. As a result, weak signals can be blocked which prevents vanishing gradient. The performance of both the RNN and LSTM network are evaluated to find the efficacy of the model.

### 3.4 ARIMA

Irfan Ahmed Mohammed Saleem, Dr. S. Jaisankar (2018), ARIMA model is implemented to compare its predictability with the LSTM and figure out which is the most suitable method for time series data which has huge fluctuations. ARIMA (Auto regression integrated moving average) is a class model that captures a suite of different standard temporal structures in time series data which include trend, seasonality, cycles, errors and non-stationary data. This allows it to exhibit dynamic temporal behaviour in a time sequence. The data preparation phase is done similar to the LSTM model approach.

---

3. DATA SOURCE

Data can be collected from any of the Bitcoin exchanges performing with a higher volume such as Bitfinex, Binance, OKEx, Upbit, Bithumb, and GDAX. The data focused in this study is from Bitfinex exchange dated from Apr 28, 2013, to Feb 28, 2018.

4. IMPLEMENTATION

The study focuses mainly on the closing price of the bitcoin to develop the predictive model. The increase or decrease in the bitcoin price with the higher volatility makes it harder to predict however the machine learning models try to predict to some extent with a significant accuracy. Here the implementation is carried out using Recurring Neural Network with LSTM.

5. DATA NORMALISATION

In neural networks and in other data mining models we need to normalize the inputs, or else the network will be performing poorly. Normalization is done to have the same range of values for each of the inputs to the ANN model. This can guarantee stable convergence of weight and biases. Here normalization is done using the MinMaxScalar Package. Once normalization is done, the data is plotted using matplotlib libraries and the trend is viewed to check the fluctuations of the closing prices and volume of the bitcoin over the past 5 years (2013-2018).

![Fig 5.1](image-url)

The above chart depicts that there is higher fluctuation in the bitcoin price from April 2017 to Jan 2018.

Heatmap is displayed to find the correlation between open, high, close, low and percentage change using the below code.
5.3 SPLITTING OF DATA

In statistics and machine learning we split the data into training and testing data. The model is tried to fit on the training data, in order for us to make predictions. When we do this, there are chances for two things to happen, one is overfitting of the model and the other one is underfitting of the model. Overfitting means that the model is trained too well and the predictions are too close and when the model is underfitted it does not fit closely with the model. Here the Scikit Library is used for splitting the data. The data is split into 67% for training and 33% for testing. The total observation in train dataset is 1231 and in the test dataset is 607.

5.4 TRAINING THE MODEL

LSTMs help preserves the error that can be back-propagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps (over 200), thereby opening a channel to link causes and effects remotely.

S. Saravanakumar, V. Dinesh Kumar (2017) LSTMs contain information outside the normal flow of the recurrent network in a gated cell. Information can be stored in, written to, or read from a cell, much like data in a computer’s memory. The cell makes decisions about what to store, and when to allow reads, writes, and erasures, via gates that open and closes. The below function convert the series into supervised learning data.

There are two models available in Keras. One is a sequential model which is suitable for time series forecasting and the other one is model used with functional API. The dense layer is used output layer with the input shapes. The optimizer function used is ‘Adam’ which is having a learning rate of 0.01 with the mean absolute error as the loss function. The loss method used is mean absolute error.

5.5 ROOT MEAN SQUARED ERROR

Time series generally focus on the prediction of real values, called regression problems. Therefore the performance measures in this tutorial will focus on methods for evaluating real-valued predictions. The most commonly used mean squared error (MSE), mean absolute
error (MAE), root mean squared error (RMSE) etc.

5.6 ARIMA (Auto Regressive Integrated Moving Average) MODEL

ARIMA model is implemented to compare its predictability with the LSTM and figure out which is the most suitable method for time series data which has huge fluctuations. ARIMA (Auto regression integrated moving average) is a class model that captures a suite of different standard temporal structures in time series data which include trend, seasonality, cycles, errors and non-stationary data. This allows it to exhibit dynamic temporal behaviour in a time sequence. The data preparation phase is done similar to the LSTM model approach.

5.6.1 ARIMA MODELING

The underlying principle in ARIMA model is to estimate the trend and seasonality in the series and remove those from the series to get a stationary series. Then statistical forecasting techniques can be implemented in this series. The final step would be to convert the forecasted values into the original scale by applying trend and seasonality constraints back. **Trend** – varying mean over time. For eg, in this case, we saw that on average, the number of passengers was growing over time.

**Seasonality** – variations in specific time-frames. eg people might have a tendency to buy cars in a particular month because of pay increment or festivals.

**AIC: Akaike information criterion.** AIC is a model quality measure, developed by Hirotugu Akaike, that penalizes complex models to prevent overfitting. In this definition, $k$ is the number of estimated parameters, including initial states, and $SSE$ is the sum of the squared errors. The best order and seasonal order for the model is $(1,1,1)x(1,1,1,12)$ and the model is implemented with the same.

Static forecasting is carried out and the RMSE is calculated to compare with the results of the LSTM model.

6. RESULTS

From the results, it is evident that the machine learning models (LSTM) take much greater time to compile because of its complex calculations than the traditional models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Compilation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>61 MilliSeconds</td>
</tr>
<tr>
<td>ARIMA</td>
<td>4 MilliSeconds</td>
</tr>
</tbody>
</table>

The model compilation time of LSTM is 61 milliseconds and for ARIMA model
it is 4 milliseconds.

The lower the loss, the better a model (unless the model has over-fit to the training data). The model compilation time of LSTM is 61 milliseconds and for ARIMA model it is 4 milliseconds. From Fig 6.1, the loss is minimal for the LSTM model at the learning rate of 0.01. It is not the best fit as it is almost impossible to make both the train and test to meet at a point as there is a huge fluctuation in the time series data. The lower loss makes the model a better one for the prediction of the bitcoin prices.

RMSE of ARIMA Model is 700.69 whereas for the LSTM is 456.78 which proves that tradition (ARIMA) model outperforms the machine learning algorithms in our case LSTM model. Note: The Root Mean Square Error is minimal for the models since the data varies from 0 to 20000 USD which huge fluctuations in their closing price.

Since the model accuracy is minimal in LSTM, machine learning algorithm (Long Short term memory) is best suitable for bitcoin forecasting. In general, LSTM is the suitable for forecasting the time series data of higher fluctuations.

Prediction graph of LSTM

Fig 6.2

7. SUGGESTIONS & CONCLUSIONS

The study focuses only on the closing price of the bitcoin to develop the predictive model. It does not take into consideration the other economic factors such as news about bitcoins, government policies, and market sentiments into account which could be the future scope of the project to predict the price with much more accuracy. The prediction is limited to the past data. The ability to predict on streaming data would improve the performance and predictability of the model. The study involves only the comparison between ARIMA and LSTM. Comparing with more machine learning models would confirm the result.

The model developed using LSTM have more accuracy than the traditional
models which prove deep learning model, in our case LSTM (Long Short-Term Memory) is evidently effective learner on training data than ARIMA with the LSTM more capable for recognizing longer-term dependencies. The study is done using the daily price fluctuations of the bitcoin which triggers the study to further investigate in future the predictability of the model using hourly price fluctuations.

8. REFERENCES


10. https://anaconda.org/anaconda/python

11. https://keras.io/getting-started/sequential-model-guide/


15. https://keras.io/models/about-keras-models/

16. https://keras.io/metrics