BITCOIN MOVEMENT PREDICTION USING SENTIMENTAL ANALYSIS

Manasa Nagilla*, Saili Kadam*2, Deepak Jha*3

*1,2,3Student, Department Of Information Technology, Vasantdada Patil Pratishthan’s College Of Engineering Visual & Arts, Mumbai, Maharashtra, India.

ABSTRACT

Bitcoin has recently sparked interest in a variety of fields, including economics, computer science, and a variety of others, due to its unique combination of encryption technology and monetary units. Nowadays, social media is an excellent representation of public attitude and opinion regarding current events. Twitter, in particular, has gotten a lot of attention from analysts who are interested in researching public mood. Bitcoin forecasting based on general public opinions stated on Twitter has proven to be a fascinating subject of study. This study tries to see how strongly the ups and downs in Bitcoin values are associated with public sentiment exhibited in tweets. Through the use of a Long Short-Term Memory (LSTM) network, the task is accomplished with variable degrees of success. As a comparison to deep learning models, the prominent ARIMA model for time series forecasting is used. The non-linear deep learning approaches, as expected, surpass the ARIMA forecast, which has a dismal performance.

Keywords: Bitcoin, LSTM, ARIMA, Twitter.

I. INTRODUCTION

Bitcoin is the most valuable cryptocurrency in the world, and it is traded on more than 40 exchanges across the world, accepting more than 30 other currencies. According to the blockchain, it has a current market valuation of $9 billion USD and over 250,000 transactions every day. Due to its relative youth and resulting volatility, which is significantly greater than that of conventional currencies, Bitcoin as a currency presents a fresh possibility for price prediction. In comparison to typical fiat currencies, it is also unique in terms of its open nature; no complete data on cash transactions or money in circulation exists for fiat currencies. Bitcoin is a challenge of time series prediction. Traditional time series prediction approaches, such as Holt-Winters exponential smoothing models, are based on linear assumptions and require data that can be divided down into trend, seasonality, and noise. This methodology is better suited to tasks including seasonal effects, such as forecasting sales. These strategies are ineffective for this purpose due to the lack of seasonality in the Bitcoin market and its excessive volatility. Given the task’s complexity, deep learning appears to be a promising technological answer based on its success in other domains. Due to the temporal structure of Bitcoin data, the Recurrent Neural Network (RNN) and the Long Short-Term Memory (LSTM) are preferred over the standard multilayer perceptron (MLP). The goal of this experiment is to see how accurate deep learning models of LSTM (Long Short-Term Memory) and RNN can forecast the price of Bitcoin (Recurrent Neural Network).

II. LITERATURE SURVEY


From this paper we came to know the effect of Bayesian neural networks (BNNs) by analyzing the time series of Bitcoin process. Here the most relevant features from Blockchain information that is deeply involved in Bitcoin's supply and demand are selected and use them to train models to improve the predictive performance of the latest Bitcoin pricing process.


The aim of this work is to derive the accuracy of Bitcoin prediction using different machine learning algorithm i.e decision tree, linear regression and compare their accuracy.


In this paper, they have considered an approach for building regression predictive model for bitcoin price using expert correction by adding a correction term. It is assumed that an experienced expert can make model correction relying on his or her experience.
Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview

The blockchain technology is a relatively new approach in the field of information technologies. As one of its first implementations, bitcoin as a cryptocurrency has gained a lot of attention. Together with Ethereum, blockchain implementation with focus on smart contracts, they represent the very core of modern cryptocurrency development. This paper is meant to give a brief introduction to these topics.


In this paper, a comparative study of the various parameters affecting bitcoin price prediction is done based on Root Mean Square Error (RMSE) using various deep learning models like Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU).

Need of New System

1. The BNN algorithm is based on the existing system’s backpropagation algorithm. It’s a unique situation. It’s a time-consuming procedure that’s difficult to employ in a large network.
2. The Decision Tree technique is unsuitable for regression and forecasting. The data is used to predict continuous values as well as in the real world because linear regression is rarely linearly separable, it is not appropriate for prediction.
3. In the existing system, where the word "expert correction" is employed, as a result, the comparative values of combined models are fairly high, providing us with insufficient precision.
4. We learned from the previous system that LSTM is very efficient and effective. Although this model was more useful than others, it had one drawback: there was no way to test it.

The forecast and live dataset input affecting bitcoin price were exhibited. Only a graph on paper will suffice.

Overall procedure

In the market, the price of bitcoin changes every minute, which means it will fluctuate, go up for a while and then come crashing down. Such a roller coaster ride in the price of bitcoin does not provide a clear indication of whether it will climb or decline in the future. As a result, our product supplies us with a prediction of the bitcoin market’s movement based on public opinion on the social media platform Twitter.

1. To get the current Bitcoin price, we use the Bitstamp API.
2. Create a dataset of Twitter sentiment values: We're utilizing Python packages to get the public sentiment. Bitcoin Price Opinion: Tweepy Library is number one.
3. Bringing the Bitcoin and Sentiment datasets together: On the bitcoin dataset, we'll combine it with the sentiment dataset based on minute-by-minute time and day To integrate the dataset, we're utilizing Python's Pandas package.
4. Decide whether to buy or sell: We are using a model with the highest accuracy, precision, and recall. That model will be used to make buy and sell decisions.
5. Notifying the user of the decision: We'll utilize Twilio to communicate the buying or selling decision. Users are notified in real-time via SMS.

Advantages

1. People investing in bitcoin could keep a track of the fluctuating prices by visiting once and not searching multiple websites using our prediction system.
2. With the decision of buying and selling, the risk of loss in trading bitcoin is reduced to minimum with our accurate prediction system.

Disadvantages

1. This system is domain specific only applicable for prediction domain.
2. It supports only English language.
3. The system functions were implemented using free wares like Python.
III. MODELING AND ANALYSIS

ARIMA Model Implementation:
The steps for implementing ARIMA in Python are as follows:

i) Plot the data and examine the trend; if the trend is non-stationary, use differencing to make it stationary.

ii) Then, when the autocorrelation is positive and above a certain threshold value, such as 0.5, plot it and find the lag order.

iii) Using the statsmodels library, an ARIMA model can be generated as follows:

iv) Call ARIMA() and pass in the p, d, and q parameters to define the model.

v) The fit() function is used to prepare the model on the training data.

vi) You can make predictions by using the predict() function and supplying the index of the time or times you want to anticipate.

viii) In the event that we applied any differencing (d>0 when configuring the model), we would prefer the projected values to be in the original scale. The value 'levels' can be given by setting the typ parameter to that value: typ='levels'.

ix) We can use the forecast() function to skip all of these conditions for predict().

x) Given the AR model's reliance on data from previous time steps for differencing, a rolling forecast is necessary. Re-creating the ARIMA model after each new observation is a primitive technique to perform this rolling forecast. Each iteration, we manually maintain track of all observations in a list called history, which is seeded with the training data and to which new observations are added.
LSTM Model Implementation:

So, we're using the LSTM method, and here's an overview of the model.

As a result, a feature vector is just a vector that comprises data characterizing an object’s key qualities. Softmax is used to create a probability vector that will help us determine the final output. The Encoder-Decoder in LSTM is a recurrent neural network designed to solve sequence-to-sequence problems, and the Encoder-Decoder in LSTM is a recurrent neural network designed to solve sequence-to-sequence problems. Then, using the following formula, we'll calculate the model’s Root Mean Square Error. For the model, we utilized the look-back approach in LSTM, which informs us how many prior states the present state is dependent on.

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{n}}
\]

<table>
<thead>
<tr>
<th>Test Cases</th>
<th>Objective</th>
<th>Description</th>
<th>Input</th>
<th>Expected Output</th>
<th>Actual Output</th>
<th>Result</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC1</td>
<td>Collecting Bitcoin and Twitter data.</td>
<td>Data is collected using API.</td>
<td>live_data.csv</td>
<td>Bitcoin Price and Stamp.</td>
<td>Bitcoin Price and Stamp.</td>
<td>Successful</td>
<td>-</td>
</tr>
<tr>
<td>TC2</td>
<td>Import dataset.</td>
<td>Data in CSV file to be imported in Python code.</td>
<td>Merged_data.csv</td>
<td>Python should read and display review.</td>
<td>I/O errors. Pandas not able to parse data.</td>
<td>Failed</td>
<td>Data must be in .CSV or .TXT</td>
</tr>
<tr>
<td>TC3</td>
<td>Pre-Processing of data.</td>
<td>Perform basic cleaning of data.</td>
<td>Unprocessed data</td>
<td>Clean values to be displayed on prompt.</td>
<td>Clean values were displayed.</td>
<td>Successful</td>
<td>-</td>
</tr>
<tr>
<td>TC4</td>
<td>Implementation of model.</td>
<td>Predicting Bitcoin price and Calculating the errors in %</td>
<td>Processed data.</td>
<td>RMSE</td>
<td>RMSE value was displayed.</td>
<td>Successful</td>
<td>-</td>
</tr>
<tr>
<td>TC5</td>
<td>Connection to Storing</td>
<td>Processed data.</td>
<td>Predicted</td>
<td>Predicted</td>
<td>Successful</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
### IV. RESULTS AND DISCUSSION

The website consists of 5 fields that will enable the user to see the predicted bitcoin value, true bitcoin value, Sentiment value, Date and time, and Decision. The interface is intended on demonstrating the purpose of the proposed system. This website will be for the live bitcoin price prediction where internet users can see the bitcoin price prediction and we can then send the buying/selling decisions per minute. Also per minute SMS notification will be provided to the user on visiting the website.

<table>
<thead>
<tr>
<th>SQL DB</th>
<th>predicted values in SQL DB</th>
<th>values to be store.</th>
<th>values were store.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC6</td>
<td>SMS Notification</td>
<td>Sending real-time notification to user using SMS</td>
<td>Processed data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1: Website**

**Figure 2: SMS Notification**

To safeguard your property, do not disclose your verification code to others!
In total, two experiments were carried out using this method. In the first trial, we discovered that the ARIMA model had a root mean squared error of 16.939 percent. Then we realised that this was too high, so we deployed the LSTM model, which yielded a root mean squared error of 7.818.

![Figure 3: ARIMA Model Output](image)

In the second experiment, the major goal was to learn more about Lookbacks in LSTM. Lookback is similar to a Markov chain in that it shows us how many prior states the current state is based on. We'll use the knowledge from previous time steps to forecast the next step because we're dealing with time-series data. The following tests were conducted for this aim, and the results are tabulated.

<table>
<thead>
<tr>
<th>LSTM input Sequence Length</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookback = 1</td>
<td>8.198</td>
</tr>
<tr>
<td>Lookback = 2</td>
<td>7.818</td>
</tr>
<tr>
<td>Lookback = 3</td>
<td>10.519</td>
</tr>
<tr>
<td>Lookback = 4</td>
<td>12.493</td>
</tr>
<tr>
<td>Lookback = 5</td>
<td>10.655</td>
</tr>
<tr>
<td>Lookback = 6</td>
<td>11.675</td>
</tr>
<tr>
<td>Lookback = 7</td>
<td>12.508</td>
</tr>
<tr>
<td>Lookback = 2 + Sentiment</td>
<td>7.303</td>
</tr>
</tbody>
</table>

![Figure 4: LSTM Model Output](image)
V. CONCLUSION

RNN and LSTM deep learning models are clearly useful for Bitcoin prediction, with the LSTM being better at recognizing longer-term dependencies. However, with such a high variance challenge, translating this into remarkable validation findings is tough. As a result, the task remains challenging. There's a thin line between overfitting a model and preventing it from learning enough. Dropout is a useful feature that can help with this. Despite utilising Bayesian optimization to optimise dropout selection, good validation results could not be guaranteed. Despite the indicators indicate good performance, the ARIMA forecast based on error performed much worse than the neural network models, as the RMSE value in the ARIMA model is 16.939, while the RMSE value in the LSTM model is 7.303. Based on the findings of this experiment, we can infer that AI models that include sentiment analysis forecast values that are considerably closer to genuine value than values that do not include sentiment. Furthermore, we can conclude from our studies that deep learning approaches like the LSTM model are more accurate for forecasting time series than our typical machine learning approach ARIMA.

ACKNOWLEDGEMENTS

We are sincerely thankful to our guide Prof. Smruti Patil for her exemplary guidance, monitoring, and constant encouragement throughout the course of this project. We owe a special acknowledgment to her for giving us a lot of time during the period of preparing this project. We could never have done it without her support, technical advice, and constant suggestions to improve our work. We convey our special acknowledgments and gratitude to our review committee whose invaluable suggestions helped us in improving our project. We are also grateful to Mr. Rahul Khokle, the HOD of the Information Technology department for guiding and supporting us. We would also like to thank our honorable principal Dr. Alam Shaikh, for encouraging us to go ahead with this project.

VI. REFERENCES


