Performance Evaluation of Machine Learning Algorithms for Bitcoin Price Prediction

Kavitha H
Associate Professor
Department of Information Science and Engineering
Siddaganga Institute of Technology, Tumkur, Karnataka
kavitha.halappa@gmail.com

Uttam Kumar Sinha
1SI15IS060
Department of Information Science and Engineering
Siddaganga Institute of Technology, Tumkur, Karnataka
uttamkumar.sinha95@gmail.com

Surbhi S Jain
1SI15IS057
Department of Information Science and Engineering
Siddaganga Institute of Technology, Tumkur, Karnataka
surbhisjain.9@gmail.com

Abstract—**Bitcoin is one of the most valuable cryptocurrency in the world with the prices as high as 19,783 United States Dollar(USD) in December of 2017. It made Bitcoin a very profitable market for investment but Bitcoin saw many ups and down as well. Today's Bitcoin price is 3913 USD but that doesn't mean that the prices always keep falling. The price of Bitcoin vary over time and is governed by various factors, like the market it is being traded in, scarcity, supply and demand. What has made Bitcoin valuable is that it can be used as a currency, we can pay a part or a fraction of Bitcoin to a person in exchange for something and the part is easily verifiable by blockchain. The small number of Bitcoin, roughly 16 million Bitcoins for the entire world has made it scare and above that its high utility makes it more prestigious.Trading of Bitcoin has proved to be very profitable to many people but the risk in trading is huge as the market of Bitcoin is very volatile. To decrease the risks, this project has been carried out to predict the price of Bitcoin using Recurrent Neural Network(RNN), Long Short Term Memory (LSTM) and Linear Regression(LR) to predict the price of Bitcoin. Evaluation of these algorithms is carried out to determine which among the two is better for the prediction of Bitcoin prices. The dataset used contains minute by minute prices of Bitcoin of over 5 years and contains almost 30,00,000 entries. Since, the dataset used is a big data, evaluating the performance of algorithms over a large dataset will give accurate results.**

Keywords- Bitcoin,* cryptocurrency, Recurrent neural network, Long short term memory, *Linear regression

I. INTRODUCTION

Bitcoin is one of the most valuable cryptocurrency in the world and the world's first decentralised cryptocurrency, in other terms it is a form of electronic cash[1]. A cryptocurrency in essence is a digital asset meaning it exists in a binary format and comes with the right to use and the data that do not possess that right are not considered assets[2], and it is designed to work as a method of exchange that uses robust cryptography to ensure reliable financial transactions, and substantiate the transfer of assets. Bitcoin is the new kind of money and an innovative payment network. It is an open-source and its design is public and nobody owns or administers it and everyone can take part to make assets[4]. After the release of Bitcoin in 2009, over 4000 alternative variants of Bitcoin which is referred to as altcoins have been created[3]. Over the past few months, cryptocurrency market has gone through an enormous volatility. The price of Bitcoin had reached its peak in December of 2017 by increasing to almost twenty thousand US dollars, and by January 2018 there was a decline in the value of Bitcoin. The thing which is even more surprising is that two years ago the value of Bitcoin was only one tenth of its current price. Therefore, what we observe is that not long ago, a year ago to be precise its value was half of what it is today. So now if we look at the Bitcoin price chart on a yearly basis, it is observed that the price is still high. When we look into the historical Bitcoin prices using a plot, we explore that there are many plots concerning the reasons behind this volatility and these are also used to assist the prediction reasoning of prices of cryptocurrency and in particular of Bitcoin[5]. It is evident that Bitcoin is probably one of the significant things of 2017, as Bitcoin grew by around eight hundred percent that year, which had the value around two fifty billion dollars, this definitely gained the interest of the world wide population in cryptocurrencies.

II. RELATED WORK

Artificial Neural Network(ANN) is one of the most powerful predicting mechanism for prediction in finance and medical field. Artificial neural network can perform a variety of tasks with great accuracies some of which can be pattern recognition, prediction or classification[1]. Artificial neural network are powerful universal approximator and can approximate any continuous function. These continuous functions can be approximated with any desired level of accuracy and in that case the prediction becomes very accurate. Compared to the traditional statistical model, Artificial neural network are a better alternative because of above stated reason [2]. In one study, a literature review of comparative studies between ANN and traditional statistical models has found that out of 96 studies, only 18% of the cases concluded that traditional methods have outperformed ANN while ANN models have either performed well or outperformed in 72% of the cases [3]. Hence, it is unsurprising that neural networks have been increasingly gaining more attention in the field of finance, especially for time series forecasting due to the highly non- linear and volatile nature of the financial market. Bitcoin is not a very old technology and hence only very less price prediction model is available. [4] deals with time series data and have created three time series data sets for 120, 60 and 30 minutes. Then they perform random forest classifier on their datasets to produce three linear models. To predict the price of Bitcoin from [4] these three models are linearly combined. From [5] a comparison is made between non linear autoregressive exogenous and multi layer perceptron. To evaluate and build the network they have used neural network toolbox of matlab. though non linear autoregressive exogenous performs better than the multi layer perceptron the authors have concluded the multi layer perceptron can be used for stock market prediction as multi layer perceptron can be easily modified for other parameters of Bitcoin. In [6], support vector machine and artificial neural network are used and that model gave 55% better results when compared to a regular artificial neural network. Several studies [4, 7] have carried multi layer perceptron but it anaylses only one observation at a time. A better version of recurrent neural network can be the long short term memory. Apart from having memory the long short term memory can choose which data to forget and which data to remember based on the importance and weight of that feature. [8] have implemented a Long Short Term Memory with Recurrent Neural network which is used for implementation and discussed in this paper as well. The Long Short Term Memory has a feedback network apart from feedforward neural network and hence it can be deployed even for general purpose computation. One minor drawback about long short term memory is that it requires a lot of computation and hence processing can be slow sometimes.[9] built a neural network and compared autoregressive integrated moving average, recurrent neural network and zhang's hybrid models on a large scale dataset and concluded that zhang's model got the best prediction results. In [10], authors have conducted an empirical study which compares the Bayesian Neural Network with other linear as well as non linear benchmark models based on the process of modeling and prediction of Bitcoin prices. Generalized linear model and bayesian regression are used for predicting Bitcoin price in [11-13] authors explored the relationship between the features of Bitcoin such as transaction volume and cost per transaction and the change of Bitcoin price the next day by using Genetic algorithm based selective neural network[.

III. Proposed Work

Before The dataset that we have used in this paper is bitstamp USD_1-min_data_2012- 01-01_to_2019 -03-13.csv and it contains minute by minute Bitcoin price from 2012 to 2019. The dataset has eight attributes, Timestamp, Open, High, Low, Close, Volume_BTC, Volume_Currency, Weighted_Price. The dataset was downloaded from Kaggle, a popluar website for machine learning. The Timestamp attribute of the dataset contains the date and timezone information so we have used pandas.Series.dt.date which returns numpy array of python datetime.date objects i.e, the date part of Timestamps without timezone information. Now we make an array and group-by the timestamp information without the timezone information which we obtained earlier. To calculate the real price of bitcoins we use the above array and take the mean of Weighted Price column(attribute). After obtaining and cleaning, the dataset is normalized, it can be done by using the Weka tool which is available online. This completes the first step of the system. To be able to create a program that trains on the historical BTC prices and predict tomorrow's BTC price, several tasks need to be completed which includes, importing the dataset and encoding the date and then split the data into training dataset and test dataset. Next, preprocess the data and use Keras to build recurrent neural network with long short term memory. Also use Keras to build linear regression model. this includes initialising the recurrent neural network and the linear regression. In case of recurrent neural network, add the input layer and lstm layer and then add the output layer. Once that is done, the network should be compiled. next fit recurrent neural network to the training set. Make the prediction of the Bitcoin prices using the trained model for test dataset. Similarly follow the same steps for Linear Regression model. And then evaluate the model and visualise the results. To compare the two models, the metrics used are Root Mean Square Error(RMSE), Mean Absolute Error (MAE) and Coefficient of Determination(R2).

Define Network, the first step of implementation of a Recurrent Neural Network(RNN) model is to define a network. In this project Keras library which is written in Python is used to build the RNN model. A neural network is defined as a sequence of layers. In keras these layers are in a class called sequential. An instance of sequential class is created. Now the next step is to add layers in the order it has to be connected. Recurrent neural network with long short term memory must comprise of Long Short Term Memory(LSTM) units. These LSTM units are the memory units for RNN. The layer which is used as output layer is called dense. This layer is used for predicting and producing the outputs. It is created followed by the creation of LSTM layers. The input layer which is the first layer in the network must be three dimensional in nature. And the input must consists of samples, timestep and features. sample includes the number of rows in the dataset. Features are the columns in the dataset. Timesteps include lag variable which are the past observation values for a particular feature. Since, the dataset used in this project is loaded in a NumPy array which is a two dimensional array, it is converted into three dimensional array using reshape function which is available in NumPy. The layers can be stacked on the model by adding it to the instance of sequential class. Sequential class can be thought as a pipeline in which the raw data is fed and in the end predictions come out in the order the input was fed. The activation function used in recurrent neural network is linear because for the regression problems the linear activation function should be used which has the same number of neurons as the number of outputs. The main task of activation function is to do non linear transformation of the input making it capable of learning and performing more complex task. For a binary classification sigmoid activation function is used which is also referred as logistic function. In case of the linear regression model, an activation function is not used. And that is the reason why a neural network without an activation function is just a Linear Regression model. For multiclass classification, softmax is used as the activation function. Thus in this project both types of models are chosen to check which among the two models perform better and predicts accurately.

Compile Network, the second step of implementation of a recurrent neural network model is compiling the network once it is defined. The concept of compiling the network is to precompute which is a necessary step after a model is defined. It is an efficiency related step. The task of compile network step is to transform the simple sequence of layers which is already defined in step one into a series of matrix which is highly efficient and it is in a format which can be used to execute the model on the CPU because that is how the project has configured Keras. There are parameters like optimization algorithm, loss function which has to be specified for compiling the network. The optimisation algorithm used in this project is adam which requires the tuning of learning rate. The role of optimisation algorithm in compilation step is that the loss function which is used to evaluate the network is minimised using this optimisation algorithm. The loss function used in this project is mean squared error because for regression problems that is loss function prescribed. For binary classification, the type of loss function that can be used is logarithmic loss which is also referred to as cross entropy or binary cross entropy. For a multiclass classification, multiclass logarithmic loss is used as a loss function, it is also called as categorical cross entropy. There are many optimisation algorithms like stochastic gradient descent, adam and RMSprop. Among these three, stochastic gradient descent is commonly used. It requires the tuning of a learning rate and momentum. In addition to optimisation algorithm and loss function as the parameters to the compile function, metrics can also be specified that has to be collected while fitting the model which is covered in the next section. And the most common metrics used for classification problems is accuracy.

Fit Network, the third step of implementation of a recurrent neural network model is to fit the network once it is compiled. By fit network it means that it can now adapt the weights on a training dataset. For fitting the network, the training dataset has to be specified, the inputs should be in a form of matrix X and the output should be an array of matching output patterns y. The recurrent neural network is trained using backpropagation algorithm. It is a method using which gradients are calculated that is needed in the calculation of the weights that is to be used in the neural network. The network is optimised according to the optimisation algorithm and loss function which is specified in the previous section. The network has to be trained for a specific number of epochs. In one epoch the entire dataset undergoes a forward pass and a backward pass in the network. The size of dataset is huge, all the data points in the dataset cannot be advertised to the entire network. Hence, the dataset is divided into batch size. The value of batch size determines the number of examples that will be advertised in each epoch or exposure. These batch are a set of input and output pattern pairs. This defines the number of patterns that will be exposed before the weights are updated in each epoch. The number of input output pairs chosen for each epoch also determines the optimisation efficiency, because care must be taken that not too many patterns are selected for each batch. The reason being that the efficiency will go down if too many patterns are loaded into the memory at a time. If the number of neurons is increased in the long short term memory layer then the capacity of the network also increases. With this it can be observed that the performance of the network also increases. Once the model is fit, the summary of the performance of the model during the training of the model has to be stored in an object called as history.

Evaluate Network and Make Predictions, the fourth step of implementation of recurrent neural network model is evaluating the network which is trained. The evaluation can be done using both the training dataset as well as test dataset. But if we perform evaluation of the network on the training dataset, it might give good results because the network has already seen all this data before at the time of training. And this performance evaluation result wouldn't be right to come to a conclusion regarding the performance of the model for predicting the Bitcoin price. We can then evaluate the network on a test dataset, which is not seen by the model yet. This will provide an estimate of how well the model is performing at predicting the price of Bitcoin for the dataset which is not seen earlier by the model. The model can also evaluate the loss and metrics like accuracy if it is specified when the model is compiled. To know the progress of the model, verbose should be on else it can be set off by setting it to zero as the verbose argument. If the model is performing well than the fit model can be used for making predictions on the new dataset. The array of input patterns is provided to the model to make predictions. The prediction made by the model is of the form which was provided by the output layer. This project is of type regression problem, so the predictions made are in the same format as that of the problem because linear activation function is used. In case of a binary classification problem, the output is not directly in the form of problem therefore the predictions which are in the array of probabilities are converted to one or zero rounding. After the model has made predictions, the performance evaluation of both the models is done using three most common metrics for regression problems. The mean absolute error, root mean square error and R squared value also called as coefficient of determination are the three metrics based on which the accuracy of the models is decided. The model is said to be accurate if it has low Root Mean Square Error value, also the value of mean absolute error must be low. Root mean square error tells how good the model is fit. The model is said to be of good fit if the value of coefficient of determination is close to unity. If the value of R squared is very less than the model is said to be worse. The metrics are determined for both recurrent neural network and linear regression models.

IV. Result Analysis

The actual prices of Bitcoin from the year 2013 till 2018 is plotted on a graph along with the predicted prices of Bitcoin using the Recurrent Neural Network(RNN) model. The figure 4.1 shows the graph of Bitcoin Price Prediction using RNN model. Recurrent neural network predicts the price of Bitcoin more accurately than the linear regression model which is implemented. It is also observed that the predicted values of Bitcoin differs every time the model is trained which implies that the model is learning. And with more learning it is able to predict the values of the Bitcoin prices more accurately. The

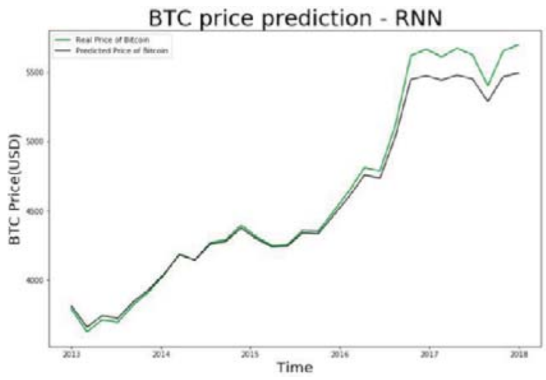

Figure 4.1 Bitcoin Price Prediction using Recurrent Neural Network mode

training of the model is stopped when a certain number of epochs are reached.

The actual prices of Bitcoin from the year 2013 till 2018 is plotted on a graph along with the predicted prices of Bitcoin using the linear regression model. The figure 4.2 shows the graph of Bitcoin price prediction using linear regression model. In this project, number of epochs considered is hundred which means the training of model stops when hundred epochs is reached. One epoch is not sufficient to train the model completely. Hence hundred number of epochs is chosen. In one epoch the entire dataset is passed to the entire neural network. Since the entire dataset is huge it is difficult to process or feed the entire dataset in one pass. Therefore, what is done is one epoch is divided into different number of batches. And each batch has a batch size. In this we have considered a batch size of five. It means that each batch has five training examples. The number of epochs often results in under-fitting of the curve. The three metrics used for the performance evaluation of machine learning models are Root Mean Square Error(RMSE), Mean Absolute Error(MAE) and Coefficient of Determination. Root Mean Square Error, Mean Absolute Error and R squared values are the most common metrics used to evaluate the accuracy of the models. It is important to understand these metrics are used to determine whether the models are accurate or misleading. *Mean Absolute Error* measures the average magnitude of the errors in a set of predictions, without considering their direction. Mean Absolute Error is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. Mean Absolute Error is

Figure 4.2 Bitcoin Price Prediction using Linear Regression model

the average vertical as well as horizontal distance between each point i among n data points and the identity line. If the value of MAE is more than the model is not said to be a good model. MAE is not sensitive to outliers unlike mean square error. MAE is the linear score, which means that all the individual differences are weighted equally in the average. This metric is widely used in finance. *Root Mean Square*

standard deviation of the differences between predicted values and observed values which is known as residuals. *Coefficient of Determination* is another common metric used for regression models. R squared value is used to determine how well the independent variable explain the variability in the dependent variable. This metric is closely related to mean square error but it is scale free. The term scale free indicates that the output values can range from very small to very large but it doesn't matter. The best value of Coefficient of Determination is unity. If the value of R squared is low than that model is poor. The value of it may vary from 0% to 100%. If the value is 100% it would imply that there is no variance at all meaning the two variables are correlated perfectly. The numerator part contains the average of the squares of the residues. It is the MSE of the model. The denominator part contains the variance in the Y values. It is the MSE of the baseline. Table 4.1 shows RMSE, MAE and Coefficient of Deter -mination values for RNN and Linear Regression. Recurrent Neural Network model with Long Short Term Memory is evidently effective for forecasting and prediction of Bitcoin prices than linear regression model because of its capability to recognize longer term dependencies. One limitation in training both the models is the significant computation required. If the size of the dataset is small then the recurrent neural network model does not train well and gives bad set of predictions. The recurrent neural network model has low Root Mean Square Error(RMSE) value which determines the square root of the variance of the residuals compared to linear regression model. The Recurrent Neural Network model has low Mean Absolute Error(MAE) value which determines the mean of the absolute errors compared to Linear Regression model. The R squared value of Recurrent Neural Network model is close to unity compared to linear regression model which determines that recurrent neural network model is better than linear regression model

Square Error	95.067	296.747
Mean Absolute	64.389	143.704
Error Coefficient of		
Determination	0.981	0.908

Table 4.1 Performance Comparison of RNN and Linear Regression models

Error or *Root Mean Square Deviation* is another common metrics used to evaluate the regression models. The model is said to be of good fit if it has lower root mean square error value than higher values. its value is always non negative. Root mean square error is defined as the sample

V. CONCLUSION

RNN model with LSTM is evidently effective for forecasting and prediction of Bitcoin prices than linear regression model because of its capability to recognize longer term dependencies. One limitation in training both the models is the significant computation required. If the size of the dataset is small then the RNN model does not train well and results in bad set of predictions. The RNN model has low RMSE value which determines the square root of the variance of the residuals compared to linear regression model.

REFERENCES

1. S. Gupta and L.P. Wang, "Stock Forecasting with Feedforward Neural Networks and Gradual Data Sub-Sampling," *Australian Journal of Intelligent Information Processing Systems, vol. 1*1, pp. 14-17, 2010.
2. T.X. Zheng, K. Fataliyev, and L. P. Wang, "Wavelet neural networks for stock trading and prediction," *Invited Oral presentation, SPIE Defense, Security, and Sensing, Bal-timo*re, USA 29 2013.
3. Khashei M. and Bijari M., "An artificial neural network (p,d,q) model for timeseries forecasting*," Expert Systems with Applications, vol. 37, pp. 479-489, January* 2010.
4. D. Shah and K. Zhang, "Bayesian regression and Bitcoin," *in 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton),* pp. 409-415, 2015.
5. M. Daniela and A. BUTOI, "Data mining on Romanian stock market using neural networks for price prediction". *informatica Economica, 17, vol. 3, 2013.*
6. A. Greaves and B. Au, "Using the bitcoin transaction graph to predict the price of bitcoin," 2015.
7. Y. Yoon and G. Swales, "Predicting stock price performance: A neural network approach," *in System Sciences, 1991. Proceedings of the Twenty-Fourth Annual Hawaii International Conference on, vol*. 4. IEEE, 1991, pp. 156–162.
8. F. A. Gers, D. Eck, and J. Schmidhuber, "Applying lstm to time series predictable through time-window approaches," *Neural Nets WIRN Vietri-01, pp. 193-200. Springer, London, 2002.*
9. N. K.Ahmed, A. F. Atiya, N. E. Gayar, & H El-Shishiny, "An empirical comparison of machine learning models for time series forecasting*", Econometric Reviews, Vol*. 29(5-6), pp.594-621, 2010.
10. H. Jang, & J.Lee, "An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on block chain information", *IEEE, Vol. 6*, pp. 5427-5437, 2017.
11. Bashar, Abul. "SURVEY ON EVOLVING DEEP LEARNING NEURAL NETWORK ARCHITECTURES." *Journal of Artificial Intelligence* 1, no. 02 (2019): 73-82.
12. Velankar Siddhi, Sakshi Valecha, and Shreya Maji. "Bitcoin price prediction using machine learning*," 20th International Conference on Advanced Communication Technology (ICACT), IEEE* pp. 144-147, 2018.
13. E. Sin, & L Wang, "Bitcoin price prediction using ensembles of neural networks", *13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), IEEE* pp. 666-671, 2017.