2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) | 978-1-7281-8337-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICACCCN51052.2020.9362735

Bitcoin Price Prediction Using Deep Learning and Real Time Deployment

E. Mahendra
Department of Electronics and Communication
Jaypee Institute of Information Technology,
Sector-128, Noida-201304, India
Email – eshamahendra@gmail.com

H. Madan
Department of Electronics and Communication
Jaypee Institute of Information Technology,
Sector-128, Noida-201304, India
Email – harshitamadan270@gmail.com

S. Gupta
Department of Electronics and Communication
Jaypee Institute of Information Technology,
Sector-128, Noida-201304, India
Email – shivangigupta615@gmail.com

S. V. Singh
Department of Electronics and Communication
Jaypee Institute of Information Technology,
Sector-128, Noida-201304, India
Email – sajaivir@rediffmail.com

Abstract—**The rapid development of digital currency especially crypto currency during the last decade is the most controversial development in the modern global economy. Because of the high volatile market and fluctuations in Bitcoin price, it has led a lot of confusion among the investors. This paper proposes the novel method of the construction of prediction model using deep learning approach. The proposed approach was found to be more accurate than the Machine learning models used for prediction as the deep learning model consider the non-linear nature of price. The results verify the applicability of model and give a direction to investors on how deep learning techniques can be used in decision making.**

Keywords:—Bitcoin; blockchain; deep learning; Flask.

I. INTRODUCTION

Digital currency is a method for trade which is web based and utilizes cryptographical capacities to perform money related exchanges [1]. The principle highlight of cryptographic money is that it isn't constrained by any fundamental power: the circulated pith of the blockchain makes digital forms of money hypothetically invulnerable to the old methods of government control and obstruction. Digital currencies support blockchain innovation to pick up decentralization, permanence, and straightforwardness [2]. Cryptographic forms of currency can be posted straightforwardly between two gatherings through the means of the utilization of open and private keys. These exchanges can be done with least handling expenses, allowing clients to keep away from the precarious expenses charged by conventional money related foundations [3]. Bitcoin is an advanced installment that uses cryptographic money and distributed (P2P) mechanization to produce and oversee financial exchanges instead of a focal power. Time arrangement estimating or expectation is a notable issue [4].

Much research has been accomplished for anticipating markets, for example, the stock market. Digital forms of money can be viewed as a type of virtual cash proposed to fill in as a mode of trade and presents a fascinating point since it tends to be treated as a period arrangement expectation issue [5]. This difficult despite everything stays in incipient stages. Thusly, there is high unpredictability in the market, and this offers open doors for additional exploration on the forecast of digital currency cost [6]. In addition, digital currencies, for example, the bitcoin are progressively received over the world. Due to the open idea of the digital money, it works on a decentralized, distributed, and trustless framework in which all exchanges are passed to an open record known as the blockchain. Such straightforwardness is obscure in the realm of old style money related markets. We will utilize the strategy of profound learning at forecast of bitcoin costs of the state-wise wrongdoing dataset. The bitcoin information is separated from the official site Kaggle.com. It comprises of the data like timestamp, open value, close value, significant expense, low cost and volume. Before preparing of the model information preprocessing will be finished after this component determination and scaling will be done so precision get will be high[7-8]. The long short term memory (LSTM), gated recurrent unit (GRU) and convolution neural network (CNN) [9] calculations will be tried at forecast of bitcoin costs and one with better exactness will be utilized for preparing. When the calculations were handled, exactness of various calculations was estimated and the calculation with the most precision is utilized for the forecast

II. METHODOLOGY

A. The Prediction Table

The proposed approach for building the deep learning model to predict bitcoin coin prices involved various steps as shown in Fig. 1 and discussed below:

1. **GATHERING DATA:** The dataset requirement entirely depends on the project requirement. The dataset can be collected from various sources such as file, database or even a sensor. The dataset of the bitcoin prices used in

ISBN: 978-1-7281-8337-4/20/$31.00 ©2020 IEEE

this work to build deep learning model was collected from Kaggle.com and some of the rows and columns of the dataset are shown in Table 1.The complete dataset has 9 columns and 1098 rows. The CSV file of dataset has prices based upon various factors such as open, high price, closing market price, market volume and the weighted price [4].

2. **DATA PREPROCESSING:** This is the most important step in deep learning which helps in building the model more accurately to perform the analysis. In this step, data collected from the site is converted to the clean data set. Then, the data is split into testing set and the training set. For e.g. Data can be divided into 80% of training data while 20% of testing data. [6].
3. **DATA SCALING PHASE:** In this step the data is scaled according to model requirements. It reshapes data to make the model more suitable [7].
4. **MODEL-BUILDING PHASE:** The preprocessed data is used to build the best performing model. Using tensor flow as the backend engine, Keras is use to increase the accuracy of the prediction model [8].
5. **MODEL LEARNING PHASE:** After the training data is defined, it is configuring with the defined model to start the learning phase. This is achieved using the **compile ( ) method** of the Keras sequential model [9]. After the fully configured neural network is defined, the data is passed to the model for training. This is achieved by calling **fit ( ) method.**
6. **EVALUATION:** This is the integral part as it helps in finding the best model that represents data and how well the prediction is achieved. Input values are passed to the model and the output is the predicted values. The output is compared with the testing data to calculate accuracy and the RMSE values.

Table 1 : Dataset used has columns as shown above

S.No.	Timestamp	Open	High	Low	Close	Volume (B)	Volume (C)	Weighted Price
1.	11.11.2012	966.34	1005	960.53	997.75	6850.59	6764742	987.47
2.	12.11.2012	997.75	1032	990.01	1012.54	8167.38	8273577	1013
3.	13.11.2012	1011.44	1039	999.99	1035.24	9089.66	9276500	1020.56
4.	14.11.2012	1035.51	1139.89	1028.56	1114.92	21562.46	23469645	1088.45
5.	15.11.2012	1114.38	1136.72	885.41	1004.74	36018.86	36211400	1005.35
6.	16.11.2012	1004.73	1026.99	871	893.89	27916.7	25523261	914.26
7.	17.11.2012	894.02	907.05	812.28	906.2	20401.11	17624310	863.89
8.	18.11.2012	906.2	941.81	881.3	909.75	8937.49	8168170	913.92
9.	19.11.2012	909.8	912.87	875	896.23	8716.18	7780059	892.6
10.	20.11.2012	896.09	912.47	889.41	905.05	8535.52	7704271	902.61
11.	21.11.2012	905.24	918.4	755	778.62	35893.77	29459969	820.75
12.	22.11.2012	778.7	832.99	751.34	807.47	17400.14	13632251	783.46
13.	23.11.2012	807.51	831.4	775	825.86	11409.52	9224730	808.51
14.	24.11.2012	825.98	837.76	810	818.27	6614.72	5470215	826.98
15.	25.11.2012	819.52	823.45	808	821.86	4231.46	33455366	81659

B. Discussion of Software Tools Required for the Work

1. **ANNACONDA:** The most popular python distribution platform for data science and machine learning along with dealing the large data and its predictions is one and only Anaconda. It provides various environments such as Spyder, Jupyter notebook, Jupyter lab, Pycharm and these can be run and maintained separately without any interference. There are around 7500 packages available in its cloud repository. Instead of using the command-line commands one can use the Anaconda Navigator which is Desktop GUI automatically installed with the Anaconda individual edition [10].
2. **FLASK:** Flask is a web framework written in Python which allow user to use its tools and libraries to build the applications such as web-pages, blog or a wiki. It is the Unicode-based development server and debugger. This framework is used in the project for the deployment of the built model [11].
3. **KERAS:** Keras is the open-source Python library which is used to build deep learning models and do training by the data provided. It is the Tensor flow's high level API that uses Tensor flow or Thenao as its backend engine. It contains tools such as layers, activation function, optimizers and objective to build the neural network blocks [7]
4. **MATPLOTLIB:** Matplotlib is the visualization library for the Python programming language for 2D plots of the array. It has various plots such as line, histogram, scatter plot for visualizing data[12].
5. The **Pandas** and **Numpy** libraryin data science are used for the data cleaning, its manipulation and analysis.

Pandas has 2D table object called data frame to store data while Numpy provide objects for multidimensional arrays [13].

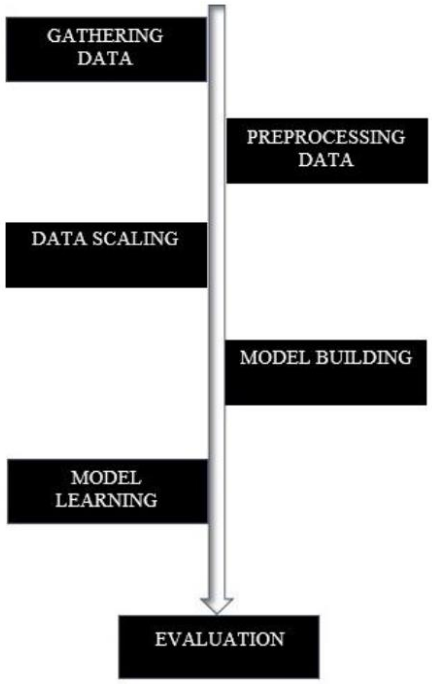


Fig. 1: Block Diagram of Process Flow

C. Implementation

The Time-Series data is the sequential data where the occurrence of the events is defined by time. If some reasonable output is to be achieved by using the data, a network is required which has access to prior knowledge to completely understand the data. This is achieved through Recurrent Neural Network. It uses its internal state (memory) for processing the sequential data. There are mainly 3 layers: The Input layer which receives the input, hidden layer which is characterized by its own weights and biases and activation function is applied here and the last is the output layer where the output is received. This type of neural network suffers the problem of vanishing gradient when the long term dependencies are present. The training of data also becomes difficult as the parameters are extremely large. To solve this problem, many variations have been developed. The LSTM and GRU solve the long term dependencies successfully.

(i) Long Short Term Memory (LSTM)

The long short term memory is the advancement of recurrent neural network architecture used in building deep learning models which deal with the problem of vanishing gradient of RNN. LSTM differs from RNN in module architecture. RNN has single neural network layer while LSTM has four layers interacting to produce output.

(ii) Gated Recurrent Unit (GRU)

Gated recurrent unit is the advance version of LSTM neural architecture that also solves long term dependencies problem. The LSTM and GRU neural networks produce same results at times but differ in their architecture. This network is capable to handle both the long-term and short-term dependencies at the same time. The three gates present the LSTM network is replaced by two gates in GRU thus reducing its complexity.

(iii) Convolution Neural Network

A convolution neural network (CNN) is another most important algorithm in deep learning which has also done much advancement in the field of computer vision. As the name suggest, this neural network is somewhat related to mathematical term convolution. The preprocessing of data is much lower as compared to other algorithms in deep learning. The architecture of CNN is similar to fully connected networks as that of neurons inside the brain. Single neuron of a layer is connected to the all neurons in the nearby layer. This fully connected networks cause the over fitting of the data. This network lacks the capability to remember thereby consider only the input and not the hidden state as that of in RNN.

(iv) Support Vector Machine

To compare our deep learning models, we came up with SVM as regressor algorithm of machine learning. It has proved to be a good prediction algorithm to predict the stock prices as compared to other back propagation algorithm. This algorithm can both be used in classification as well as regression problem but is mostly used in the classification problem. In SVM, each data value is plotted in n-dimensional space with each feature represented as the coordinate in the space. The two classes are differentiated by classification which forms the hyper-plane for differentiates the classes very well. The hyper plane is identified by thumb rule which states to choose the hyper plane which segregates the two classes in a better way.

D. Project Deployment

After the model implementation using various algorithms, the best suiting model is used to produce work in real time and then deploy it using Web API.

Application programming interface (API) is a kind of software which is used in building software applications. It is the set of protocols, routines and tools that helps the two software's to interact between them without knowing their individual implementation. API's are also used to program the graphical user interface (GUI) components. It basically provides all the building blocks for a program thus making it easier for a programmer to design new tools and the products. It provides a way to simplify design, increasing flexibility and usage of the web app. API's play a major role in collaboration of business and IT teams as they provide an easier way to connect the application to the cloud-native app development. There are mainly 4 types of APIs one can work with.

Our model is deployed to the website using Python and Flask. The dataset is trained for deep learning model and all these changes are passed on to framework then which uses UI

to get user data and finally the prediction is done as described in Fig 2. Following steps are followed for the deployment.

1. The file that contains code of LSTM model for the bitcoin price prediction is saved in the same location as the below files are created.

2. The other file is created which contains Flask APIs that receive data when the user input through GUI or API calls.

3. The request module is to be created inside the file which is used to call API from the file to display the returned value.

4. Lastly the HTML templates and CSS styling is done to develop the GUI of Flask and stored in the file of the same directory as that of the above files.

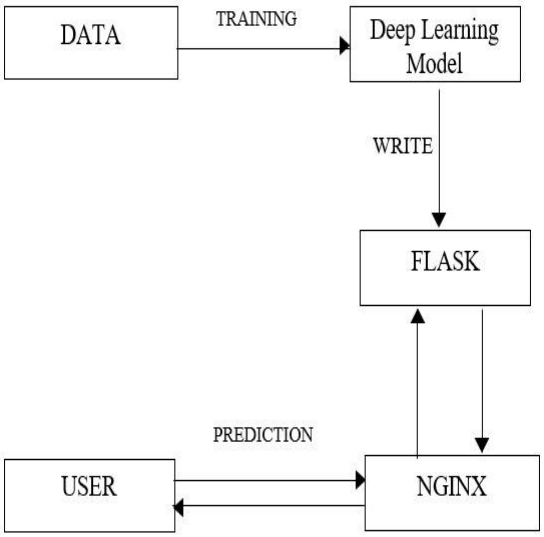


Fig. 2: Deployment work Flow

III. RESULTS AND DISCUSSION

The first algorithm implemented was LSTM. Its RMSE value came to be 0.022. The next algorithm implemented was GRU. LSTM had more number of parameters while GRU had less, so this makes GRU less complicated and more efficient. Its RMSE value came to be 0.014. Both the models had 3 hidden layers and 1 dense layer, 50 training epochs and batch size of 8 as shown in Fig. 3 and Fig. 4.

```
In [78]: runfile('C:/Users/user/Desktop/tryinglstm.py', wdir='C:/Users/user/I

Layer (type)                 Output Shape              Param #
=================================================================
lstm_48 (LSTM)               (None, None, 75)          23100
_________________________________________________________________
lstm_49 (LSTM)               (None, None, 30)          12720
_________________________________________________________________
lstm_50 (LSTM)               (None, 30)                7320
_________________________________________________________________
dense_72 (Dense)             (None, 1)                 31
=================================================================
Total params: 43,171
Trainable params: 43,171
Non-trainable params: 0
```

Fig.3: LSTM Model

```
Layer (type)                 Output Shape              Param #
=================================================================
gru_42 (GRU)                 (None, None, 75)          17325
_________________________________________________________________
gru_43 (GRU)                 (None, None, 30)          9540
_________________________________________________________________
gru_44 (GRU)                 (None, 30)                5490
_________________________________________________________________
dense_74 (Dense)             (None, 1)                 31
=================================================================
Total params: 32,386
Trainable params: 32,386
Non-trainable params: 0
_________________________________________________________________

Test RMSE:0.014
```

Fig.4: GRU Model

The following plots shown in Fig. 5 and Fig. 6 tells us the difference of Real and Predicted Bitcoin values for LSTM and GRU models respectivley.

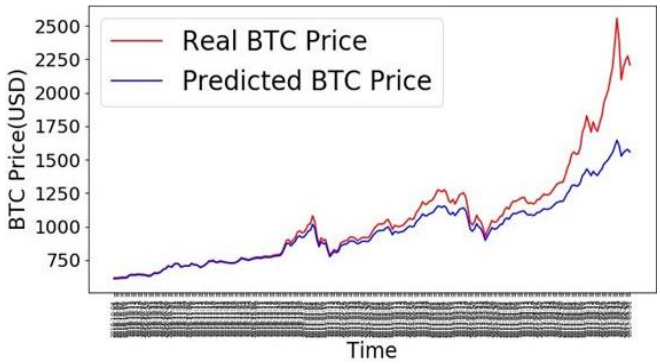


Fig.5: Plot obtained using LSTM Model

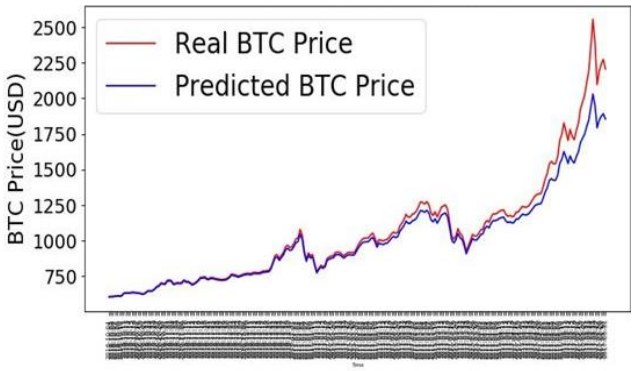


Fig. 6: Plot obtained using GRU Model

The Fig. 8 plot compares the actual and predicted prices of bitcoin using CNN Model. CNN model did the same neural networks as LSTM and GRU but with 1 convolution layer and many hidden layers. Comparing deep learning models with that of Machine learning, we used SVM and saw that it wasn't useful in our work as the accuracy came out to be negative as shown in Fig .7.

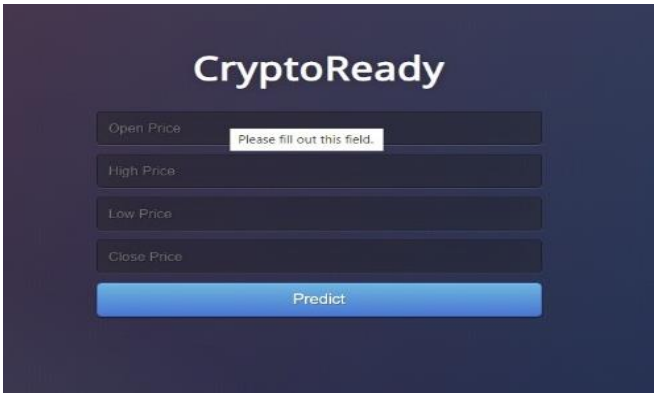
```
In [11]: svr_rbf_confidence = svr_rbf.score(x_test, y_test)
         print("svr_rbf accuracy: ", svr_rbf_confidence)

         svr_rbf accuracy:  -0.0028574795295994715
```

Fig. 7: SVM accuracy code block

Fig. 8: Plot comparing actual and predicted prices using CNN

The API deployment task was done using Flask and UI tools where an interface was developed to fetch values from user end and can be used to predict values with already trained model as shown below in Fig. 9. After providing the values by user, the API sends them to backend for prediction and the desired output in the form of Bitcoin price comes up on the screen as shown in Fig 10.



. **Fig. 9:** API design

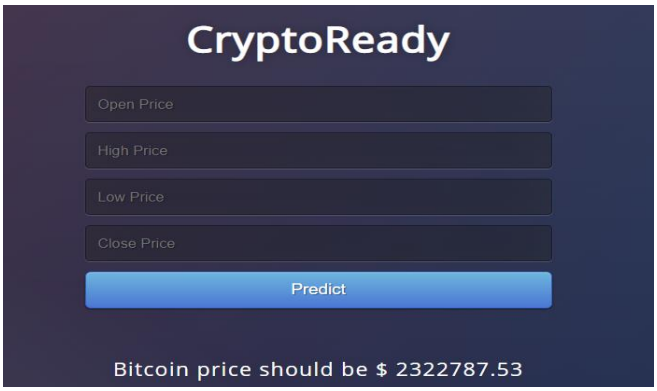


Fig. 10: API output window

IV. Conclusion and Future Scope

Prediction has always been an exciting and interesting topic for people around the globe. As the technology has grown so far, still predicting things before hand is in demand. Bitcoin has been of great importance is past few years. From blockchain to machine learning, it has occupied all markets. Predicting bitcoin prices was of great enthusiasm and knowledge. It was found that various deep learning algorithms were successful in prediction ranging from GRU, LSTM and CNN then comparing them with machine learning algorithm SVM. It can be inferred that deep learning models take long time to train. Prediction of bitcoin price was a complicated task as it is based on large dataset. Machine learning model or deep learning model without real time deployment are of no use. So API was designed for this purpose. This can be seen as our work in production. As we work, there is always a room for improvement. No. of neurons can be increased. Dataset of large columns will be of better results. Including more features or parameters is always for betterment. In coming future as per our needs, we can also develop an android app or website for same.

References

[1]. Y. Yao, J. Yi, S. Zhai, Y. Lin, T. K. G. Zhang and L. Y. Lee, "Predictive analysis of cryptocurrency price using deep learning", *International Journal of Engineering & Technology*, vol. 7, pp: 258-264, 2018.

[2]. S. Yogeshwaran, M. J. Kaur and P. Maheshwari, "Project based learning -predicting bitcoin prices using deep learning", *In Proc. IEEE Global Engineering Education Conference (EDUCON)*, Dubai (UAE), pp: 1449-1454, 8-11 April 2019.

[3]. S. Velankar, S. Valecha and S. Maji, "Bitcoin price prediction using Machine Learning", *International Conference on Advanced Communications Technology (ICACT)*, pp:144-147, 11-14 Feb 2018

[4]. V. Derbentsev, N. Datsenko, O. Stepanenko, and V. Bezkorovainyi, "Forecasting cryptocurrency prices time series using machine learning approach", *SHS Web of Conferences*, vol. 65, pp:1-7, 2019.

[5]. T. Phaladisailoed and T. Numnonda, "Machine learning models comparison for bitcoin price prediction", *10th Int'l Conference on Information Technology and Electrical Engineering (ICITEE),* pp: 506-511, January 2018.

[6]. S. McNally, S. Roche and J. Caton, "Predicting the price of bitcoin using machine learning" *26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP) Cambridge (UK),* pp:339–343, 21-23 March 2018.

[7]. F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies", *IEEE Communications Surveys & Tutorials,* vol.18 (3), pp: 2084-2123, 2016.

[8]. R.C. Phillips and D. Gorse, "Predicting cryptocurrency price bubbles using social media data and epidemic modeling", *IEEE Symposium Series on Computational Intelligence (SSCI),* pp: 1-7, 2017.

[9]. S. Mohapatra, N. Ahmed and P. Alencer, "KryptoOracle: A real-time crypto currency price prediction platform using twitter sentiments*", In Proc. IEEE International Conference on Big Data*, pp: 5544-5551, 2019.

[10]. Y. Deng, F. Bao, Y. Kong, Z. Ren and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading", *IEEE Transactions on neural networks and learning systems,* vol.28 (3), pp:653-664, March 2017.

[11]. U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu and R. Brooks, "A brief survey of crypto currency systems", In Proc. *IEEE 14th Annual Conference on Privacy, Security and Trust (PST)*, Auckland, New Zealand, pp:745-752, 12-14 Dec. 2016.

[12]. J. Sun, Y. Zhou and J. Lin, "Using machine learning for cryptocurrency trading," *In Proc.* IEEE *International Conference on Industrial Cyber Physical Systems (ICPS),* Taipei, Taiwan, pp. 647-652, 6-9 May 2019.

[13]. C. R. Phillips, and D. Gorse "Predicting crypto currency price bubbles using social media data and epidemic modelling." *In Proc. IEEE Symposium Series on Computational Intelligence (SSCI),* Honolulu, HI, USA, pp: 1-7, 27 Nov.-1 Dec. 2017.