**moneBOT: A TRADE BOT AND STOCK PRICE PREDICTOR**

**A PROJECT REPORT**

*Submitted by*

**Ayushree Ghoshal 19BAI10022**

**Dev Singh 19BAI10093**

**Akshat Bharadwaj 19BAI10188**

**P. Yash Reddy 19BAI10190**

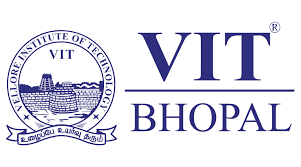
*In partial fulfillment for the award of the degree*

*Of*

**BACHELOR OF TECHNOLOGY**

*in*

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

****

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH- 466114**

OCTOBER 2020

**VIT BHOPAL UNIVERSITY KOTHRIKALAN,SEHORE**

**MADHYA PRADESH- 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**moneBOT- A Trade bot and Stock Price Predictor**” is the bonafide work of **“AYUSHREE GHOSHAL (19BAI10022), DEV SINGH (19BAI10093), AKSHAT BHARADWAJ (19BAI10188), P. YASH REDDY (19BAI10190)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported here does not form part of any other project/research work on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR PROJECT GUIDE**

DR. S SOUNTHARRAJAN, DR. NILAMADHAB MISHRA

Senior Assistant Professor,

School of AI and ML division, School of AI and ML division,

VIT BHOPAL UNIVERSITY. VIT BHOPAL UNIVERSITY.

The Project Exhibition I is held on 28th October 2020.

**ACKNOWLEDGEMENT**

First and foremost, We would like to thank the Almighty for His presence and immense blessing throughout the project work.

We wish to express my heartfelt gratitude to Dr. , Head of the Department, School of Computer Science for much of his valuable support and encouragement in carrying out this work.

We would like to thank my internal guide Dr NILAMADHAB MISHRA for continually guiding and actively participating in our project, giving valuable suggestions to complete the Project Work.

We would like to thank all the technical and teaching staff of the School of Computer Engineering, who extended directly or indirectly all the support.

Last, but not the least, We are deeply indebted to our Parents who have been the greatest support while we worked day and night on the project to make it a success.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Serial no** | **Title** | **Page no.** |
| 1 | Abstract | 05 |
| 2 | List of Abbreviation and List of Figures | 06 |
| 3 | Introduction | 07 |
| 4 | Literature Survey | 08 |
| 5 | Project Procedure | 09-10 |
| 6 | Work Done | 11-14 |
| 7 | Observation and Result | 15 |
| 8 | Conclusion and Future Work | 16 |
| 9 | References | 17 |
|  | | |

**ABSTRACT**

In this work we have discussed about the Stock Prediction model using LSTM and Trading bots that work on cryptocurrency. The prediction of Stocks is a tedious work and considering the uncertainty and unpredictable of the Stock market getting an ideas about the opening, closing, lowest highest prices of the Stocks really become very difficult. In this work we have tried the method that is known to be the most efficient method till date. We have used MACHINE LEARNING approach for our model. RNN and LSTM are used to make the trends accurate. Though designing a model that will give the exact predictions is really tough but even though having a rough idea of the price range can help the investors to at least be prepared for the upcoming uncertainty.

Trading bots are introduced in the market to reduce human workload. They can be mde with various variations according to the desired situation. Here we have just given a brief and simple ideas of designing a model. modifications are and will be always present and they can be well modified as per the choice of the company.

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATION** | **FULL FORM** |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| RNN | Recurrent Neural Network |
| LSTM | Long Short-Term Memory Approach |
| DSE | Dhaka Stock Exchange |
| DJIA | Dow Jones Industrial Average |
| ADAM | Adaptive Movement Estimation |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG NO** | **TITLE** | **PAGE NO** |
| 1 | Graphical Result of the CODE |  |

**INTRODUCTION**

Computer Science has many branches and Artificial Intelligence along with Machine Learning is the most versatile branch in this field. AI and ML together are creating wonders and they are also used in the trading and merchandise world technology. The simple concept behind Artificial Intelligence is providing the system a huge dataset and then training the system, means teaching it how to carry forward the assigned tasks so that it can work as an independent entity and also give the best possible outcomes.

Trading Bot and the Stock Price Predictor are just among the innumerable applications of Machine Learning. The curiosity of how does the machine calculate stocks all by itself and does trade is satisfied by Algorithms, which are the backbone in training a machine and making it work efficiently. Well known as Algorithmic Trading and Calculations, they are a blessing in disguise in the cooperate world. Trading and predicting have become just so easy with evolution of this technology. We all are aware of the unstability of Stocks and how tedious it is work and predict their values. These Smart machines have reduced effort made by human and thus given a new direction in prediction.

The most important Question is whether the Market is aware of the technology and they have accepted it or not. The answer is that the machine trading has now been widely used in the market overseas and thus do their trading in millions. Some companies known as Neotic, Trading Depth, HI Hedge have already set the benchmark in the AI featured trade. They have models that are a replica of the financial world to solve the real-world problems.

The correct prediction of stocks is a challenge itself. Due to unstable nature and variation of the stock market the difficulty level increases. With ML the complexity is reduced the use of RNN has made predictions much easier.

**LITERATURE SURVEY**

The concept of an Automatic Trading Bot was invented by Richard Dunchin in 1949. It consisted of prescribed rules for the purchase and selling of the funds. In the 1980’s well known traders John Henry began to adopt the trading based on rules. Apart from being famous trading bots were still not used by average investors as they were expensive.

There are two main cases for the Trading bots

* The bot can be used to make trading a simple and hassle-free process by the investors.it can look after the construction of Index and the diversification and rebalancing of portfolio.
* The bot will follow a very high-level advanced mechanism where the bot will compete with the market and make profit. However, this is not an easy task and requires much research.

The Stock price Predictions have many research works. To the Regression model the Support Vector machine were added, which were used to predict the stock market prices more accurately. Later LSTM approach was introduced and in addition to the Naive Bayessian method to be more accurate in their prediction. The Asian Stock Exchange inclusive the Chinese Stock Exchange used the back-propagation method and the neural network along with Adams Optimizer for an accurate prediction. LSTM time series learning model can be evaluated with the Emotional analysis model to get a accurate time series model for effective prediction of the stock. The final conclusion was that accurate value prediction can e achieved with this model.

Another method was the use of Artificial Neural Network along with adaptive neural fuzzy system was used to design a technology for the Dhaka Stock Exchange (DSE). Wavelet transforms, RNN and Artificial bee colony algorithm for the forecast of the stock market. The Dow Jones industrial average (DJIA), London FTSE, Tokyo Nikkei-225 index and the Taiwan Stock Exchange were all taken into consideration for evaluation and the results was the approach was fit for real time trading.

**PROJECT PROCEDURE**

To design a Stock price prediction model using the LSTM approach along with RNN have been successful in determining values. The fluctuations are very violent in the stock market. This technology helps experts to find the most effective models.

The first step would include the collection of Raw Data. Any Historical Stock Data from any well-known profit-making company is taken here we have considered the historical data of GOOGLE as the stocks of google being a huge industry are subjected to face a lot of variations and effects in the market are equally dominant. We will have to use this to make future predictions.

We need to use libraries such as numpy as we need to apply mathematical functions and operations, matplotlib foe visualization purpose of the comparison of the data, pandas a data analysis and manipulation tool that takes database and creates object, datetime for easy working with the dates.

Next step we need to preprocess the Data. this step includes the Data Discretization (reduce the part of the dataset by keeping only the particular parts) and Data transformation (normalizing the data) along with Data cleaning (filling the missing values) and Data integration. After we divide it into a clean dataset it needs to be divided into training and testing dataset for evaluation.

Following we have to do the Feature Extraction where we need to choose the features to be fed by the Neural Network. We need to import the Keras libraries and packages to build the RNN as it is a Tensor Flows high level app for building and training the deep learning models.

Now we need to train the Neural Network it is a very important step. The data is fed to the Neural Network and trained for predictions. We require to assign weights to the model followed by the Sequential layers of three LSTM layers and a dense layer with activation.

For the compilation of the RNN we use an Optimizer. We have used ADAM optimizer which combines the perks of ADAgrad and RMSprop.

ADAgrad essentially uses a different learning rate for every parameter on past guidance that have been computed for each parameter.

RMS prop considers fixing the diminishing learning rate by only using a certain number of previous guidance.

The ADAM (Adaptive Movement Estimation) computes adaptive learning from the parameter based on past gradients.

After regularization and dropout (new method to prevent overfitting). They make the neuron robust and hence predict the trends without focusing on any one.

Finally, we have the output generation step where we run our code and get the visualization of the data input in our training set. The output value has been generated by the output layer of the RNN and is compared with the target value. The error id minimized with the back-propagation method and also adjust the weights of the network.

This Entire procedure was done for the visualization of the Training dataset. The same procedure we need to require to for the Testing dataset. The same functions that we have used in the above-mentioned steps need to be called and thus we now require to test the model for giving us the desired results.

The testing phase is essential as we need to check whether our data is working properly on the machine. It will bring into the light the efficiency of work done by our machine and we will come to know whether our machine has been trained properly or not.

If the testing fails to give the desired results then we require to find the fault in the training section and repeat the entire process. We need to make corrections in the training algorithm that we have used and again run the testing dataset and the check until we get the accurate prediction of the dataset.

**WORK DONE**

Following the above-mentioned procedure, we now require to implement and code them. The step by step implementation and explanation is discussed below

**STEP 1: RAW DATA**

Historical stock data is collected and this is used for prediction of future stock prices.

***Importing the libraries***

**import numpy as np** – numpy so that we can apply mathematical function and operations to our multidimensional arrays.

**import matplotlib.pyplot as plt** -for visualisation purposes.

**import pandas as pd** – a data analysis and manipulation tool.

**import datetime** – libraray to work with data as data objects.

***Reading the dataset using.***

**dataset = pd.read\_csv('/content/Google\_Stock\_Price\_Train.csv',index\_col="Date",parse\_dates=True)**

***Display the dataset***. The head function gives us the top 5 rows present in the set. The tail gives us the last 5 rows present in the dataset.

**dataset.head()**

**dataset.isna().any()**

this function is used to check if any data is applicable or not. It detects any missing values or NA values.

**dataset.info()**

this basically prints the basic info. The 5 columns (head or tail assigned above) , number of non-null values and data type and memory usage.

**dataset["Volume"] = dataset["Volume"].str.replace(',', '').astype(floa)**

this function has been written to homogenise the datatype of the column types as we see the data type is object while the others are float.

***The next function involves taking out the 7 DAY ROLLING MEAN.*** For every single stock prediction, we require to look 7 days back collect all transactions that fall in that range and find out the average.

**dataset. rolling(7). mean().head()**

***The next step would involve comparisons of the previous graphs with the rolling mean.***

***Lastly in this step we are creating our training dataset and reading using pandas.***

**STEP 2 : DATA PREPROCESSING**

It involves data discretization (reduce a part by keeping particular parts), Data transformation(normalise), Data cleaning (fill in missing values) and data integration. After it is divided into a clean data set it is divided into a training and testing sets to evaluate.

**from sklearn.preprocessing import MinMaxScaler**

**sc = MinMaxScaler(feature\_range = (0, 1))**

**training\_set\_scaled = sc.fit\_transform(training\_set)**

here we need to sacle the data using MinMax Scaler form sklearn ( a python library )in range 0 to 1.

***Next we require to create a data structure with 60 timesteps*** we are taking data from day 1 to day 60 and make predictiobs on 61st day and then follow by taking data from day 2 to day 61 and predict the 62nd day.

**X\_train = []**

**y\_train = []**

a loop from i=60 to the end and the we append the x train[]- starts from i-60 so i=61 then its 1 thus it starts from the 1st day and ends at 61. and y train [] – gives the prediction on the I th day (61st)

**for i in range(60, 1258):**

**X\_train.append(training\_set\_scaled[i-60:i, 0])**

**y\_train.append(training\_set\_scaled[i, 0])**

**X\_train, y\_train = np.array(X\_train), np.array(y\_train)**

***The next thing to be done is to reshape the data.***

**X\_train = np.reshape(X\_train, (X\_train.shape[0], X\_train.shape[1], 1))**

**STEP 3: FEATURE EXTRACTION**

In this we are going to extract the features to be fed by the Neural Network from date, high, low, close and volume.

***We have to import Keras library and package to build the RNN. It is actually a Tensor Flow high level APP for building and training deep earning models.***

**from keras.models import Sequential**- linear stack of layers through which we create sequential models.

**from keras.layers import Dense**- this function is a regular deeply connected neural network layer commonly and frequently used to change dimensions of the output.

**from keras.layers import LSTM**

**from keras.layers import Dropout**

now ***We require to initialize the RNN*** for a time series problem we are using a regression model for that the first step is to read in the data which is a sequential data and assign it to regressor.

**regressor = Sequential()**

**STEP 4: TRAINING THE NEURAL NETWORK**

The data is fed to the Neural Network and trained for predictions. We are going to assign weights to the model. It is followed by a Sequential Input Layer followed by three LSTM layers and a dense layer with activation and a dense output layer with linear activation function.

We require to use dropout which is regularisation technique for reducing overfitting in NN. It drops units in NN as we require only one output so **units =1**

**regressor.add(LSTM(units = 50, return\_sequences = True, input\_shape = (X\_train.shape[1], 1)))**

**regressor.add(Dropout(0.2))**

# Adding a second LSTM layer and some Dropout regularisation

**regressor.add(LSTM(units = 50, return\_sequences = True))**

**regressor.add(Dropout(0.2))**

# Adding a third LSTM layer and some Dropout regularisation

**regressor.add(LSTM(units = 50, return\_sequences = True))**

**regressor.add(Dropout(0.2))**

# Adding a fourth LSTM layer and some Dropout regularisation

**regressor.add(LSTM(units = 50))**

**regressor.add(Dropout(0.2))**

# Adding the output layer

**regressor.add(Dense(units = 1))**

the next task would be to compile the RNN and use **optimizer(**one of the two arguments required to compile the Keras model). The type of optimizer can greatly affect how fast the algorithm converges the minimum value.

We have **used ADAM optimizer which combines the perks of ADAgrad and RMSprop.**

# Compiling the RNN

**regressor.compile(optimizer = 'adam', loss = 'mean\_squared\_error')**

while training we must be sure that the weights don’t get too large(focusing on one datapoint hence overfitting. We use Dropouts a new method for preventing overfitting.

**OUTPUT GENERATION**

On running and fitting the training set we have assigned value to **epoch** (a frame of time in ML) **batchsize** (refers the number of training examples)

# Fitting the RNN to the Training set

**regressor.fit(X\_train, y\_train, epochs = 100, batch\_size = 32)**

**STEP 5: Repeat the same for the test data and thus visualise the results.**

 Visualising the results

**plt.plot(real\_stock\_price, color = 'red', label = 'Real Google Stock Price')**

**plt.plot(predicted\_stock\_price, color = 'blue', label = 'Predicted Google Stock Price')**

**plt.title('Google Stock Price Prediction')**

**plt.xlabel('Time')**

**plt.ylabel('Google Stock Price')**

**plt.legend()**

**plt.show()**

**RESULT and OBSERVATION**

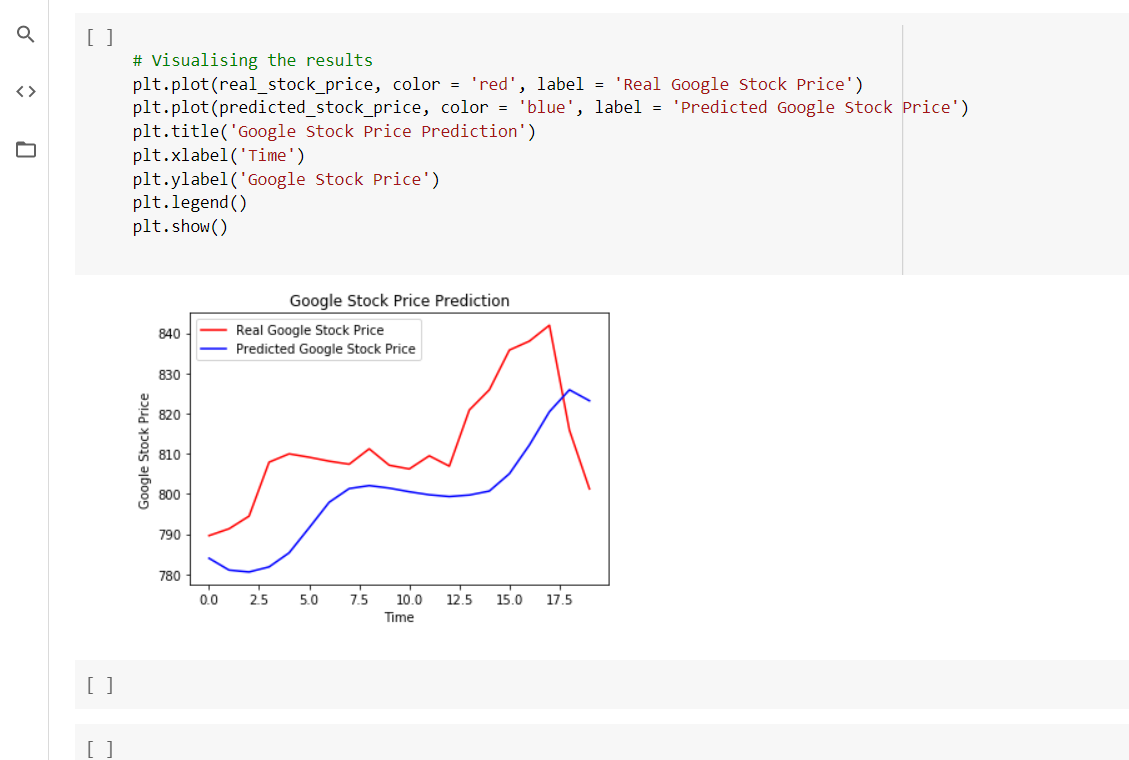


FIG 1: the result graph of the code

The figure above is the Snapshot of the Visualization result of the code. We see that the PREDICTED STOCK PRICE depicted in blue has somewhat the same pattern as to what our REAL STOCK PRICE is. The trend and variation of the graph is similar to the original trend however the accurate prediction is still not achievable as the uncertainty of the stock market is always a factor. but the most important factor satisfies by the machine is that the prediction of whether the stock price will be low or it will go down or what is the pattern and what will be the range of the prices all these ideas are very well predicted from the model. Thus the LSTM model is used for giving the accurate predictions.

**CONCLUSION and FUTURE RECOMMENDATIONS**

For the predictions of the Stocks Machine Learning technique based on RNN is found out to give the best yield. The Long Short-Term Memory (LSTM) is very much into popularity for its simplicity and for giving the best output till date. The pattern of the Graph obtained in the Stock price prediction is very close to the actual pattern. However, as the market and the movement of the stock values are highly violent it is quite possible that sometimes the machine might also fail to yield the closest result. The machinery used gives us the closest result but not the actual result. Our future aim would be to design such a model that can give the flawless report. The value of the market should be exact to the original. Specialists are working in the specifies areas for improvements and development of a perfect predicting system.

For our Trading Bot we can have many variations. The Existing Trading bot works only on Cryptocurrency. Our future aim would be to design such bots and introduce them that can work on paper currency as well especially for the Indian currency. There is no such machinery used in India and thus for our people welfare we can hope and work for the introduction of the bot that will deal with the Indian Standards and help the Indian traders and merchants.

**REFERENCES**