Crypto Hacks

PROJECT SYNOPSIS

OF MAJOR PROJECT

BACHELOR OF TECHNOLOGY
Computer Science and Engineering

SUBMITTED BY

Parth Mishra

Vaibhav Chaudhary

August 2022

KIET Group of Institutions, Delhi-NCR, Ghaziabad (UP)

Department of Computer Science and Engineering

Table of Contents

Content	Page no.
Introduction	3
Objectives	4
Literature Review	4
Methodology	6
Feasibility Study	10
Facilities required for proposed work	11
Expected outcomes	12
Conclusion Reference	14

INTRODUCTION

Predicting stock prices is a cumbersome task as it does not follow any specific pattern. Changes in stock prices are purely based on supply and demand during a period of time. In order to learn the particular characteristics of a stock price, we can use deep learning to identify these patterns through machine learning. One of the most well-known networks for series forecasting is LSTM (long short-term memory), a Recurrent Neural Network (RNN) that can remember information over a long period of time, thus making them extremely useful for predicting stock prices. RNNs are well-suited to time series data and they are able to process the data step-by-step, maintaining an internal state where they cache the information they have seen so far in a summarised version. Predicting a stock's future price could yield a significant profit.

Time series forecasting using TensorFlow LSTM (long short-term memory) neural networks.

OBJECTIVES

How accurate is machine learning at predicting stock prices? Analysis of data allows investors to make educated guesses. Their research will include reading the news and studying company history, industry trends, and other data. There is a widespread belief that stock prices are completely random and unpredictable, but this raises the question of why big firms like Morgan Stanley and Citigroup hire quantitative analysts. These days, rows of machine learning experts are likely to sit in front of computer screens in rows, rather than adrenaline-fueled men with loose ties running around yelling into a phone. Today, about 70% of all Wall Street orders are placed by software, we live in an algorithmic world.

This project utilizes Deep Learning models, Long-Short Term Memory (LSTM) Neural Network algorithm, to predict stock prices. For data with timeframes recurrent neural networks (RNNs) come in handy but recent research have shown that LSTM, networks are the most popular and useful variants of RNNs.

I have used Keras to build an LSTM to predict stock prices using historical closing prices and trading volume and visualize both the predicted price values over time and the optimal parameters for the model.

LITERATURE REVIEW

PROBLEM STATEMENT / HIGHLIGHTS

The challenge of this project is to accurately predict the future closing value of a given stock across a given period of time in the future. For this project, I have used Long Short Term Memory networks – usually just called "LSTMs" to predict the closing price of the S&P 500 using a dataset of past prices.

- ***ACHIEVEMENTS***
- **Built a model to accurately predict the future closing price of a given stock, using the Long Short Term Memory Neural net algorithm.**
- **Achieved Mean squared error rating of just 0.00003063.**

THINGS LEARNT IN THIS PROJECT.

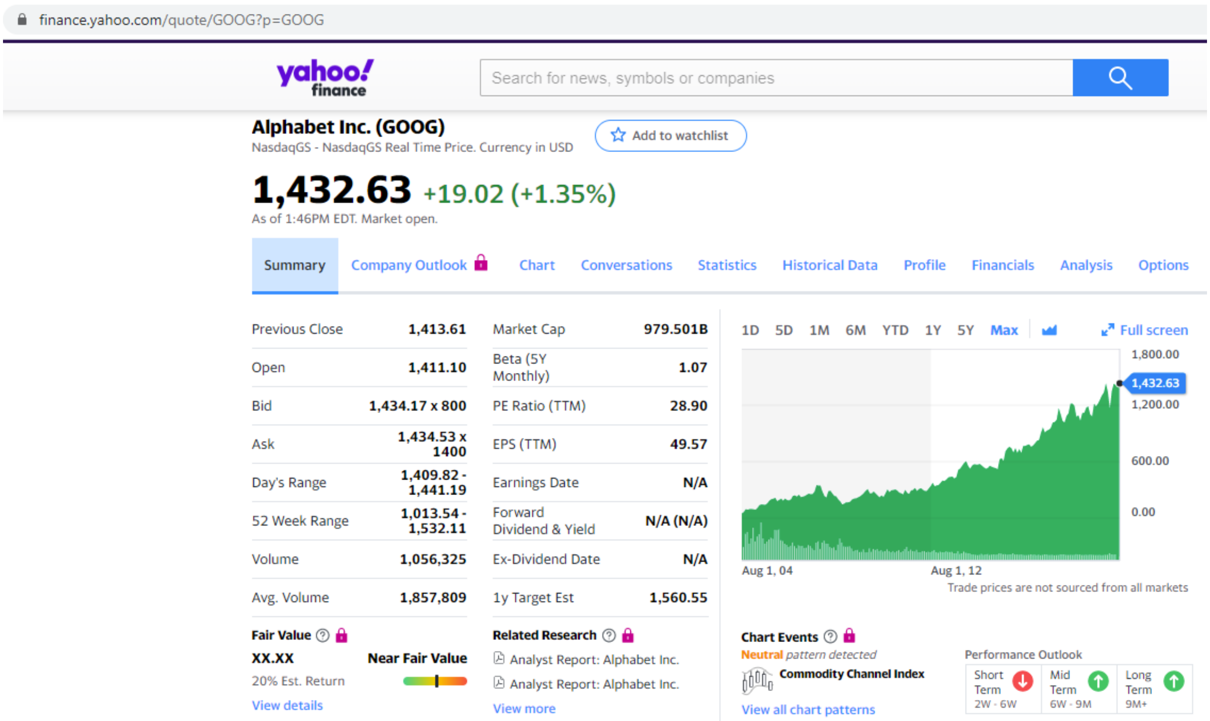
How to apply deep learning techniques: Long Short Term Memory Neural Network algorithms.
How to use Keras-TensorFlow library.
How to collect and preprocess given data.
How to analyze the model's performance.
How to optimize Long Short Term Memory Neural Network algorithms, to ensure an increase in positive results.

<u>METHODOLOGY</u>

IN ORDER TO RUN THIS MODEL FOR THE SUCCESSFUL COMPLETION OF THIS PROJECT ONE MUST KNOW ABOUT MACHINE LEARNING AND FULL STACK WEB DEVELOPMENT.

1. <u>STOCK MARKET</u>

The initial data we will use for this model is taken directly from the Yahoo Finance page which contains the latest market data on a specific stock price. To perform this operation easily using Python, we will use the finance library which has been built specifically for this, and that will allow us to download all the information we need on a given ticker symbol.

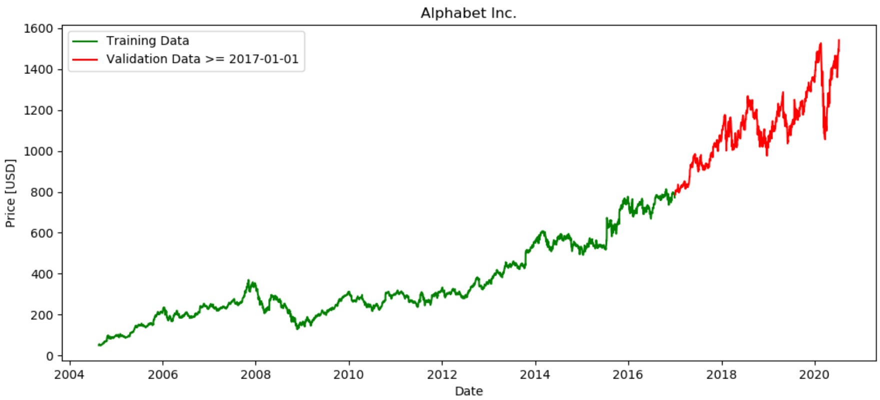


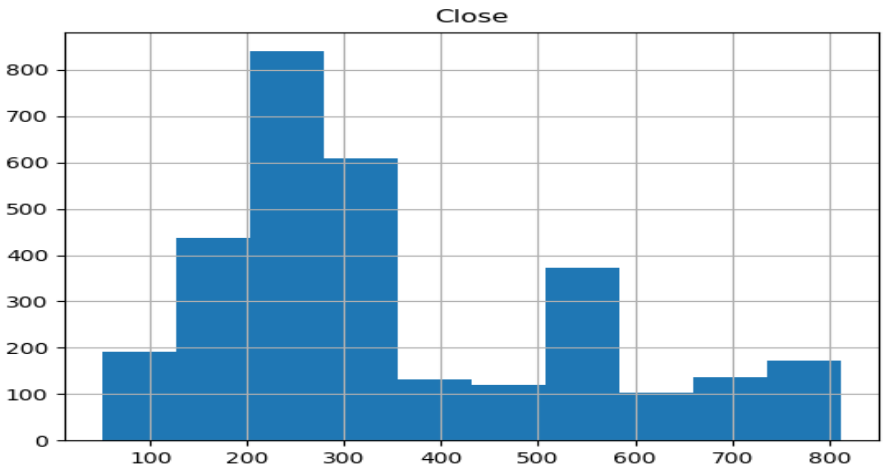
2. MARKET INFO DATA.

To download the data info, we will need the finance library installed and then we will only need to perform the following operation to download all the relevant information of a given stock using its ticker symbol.

3. DEEP LEARNING TRAINING MODELS.

Below you can find the chart with the division we will create between Training Data and Validation Data:



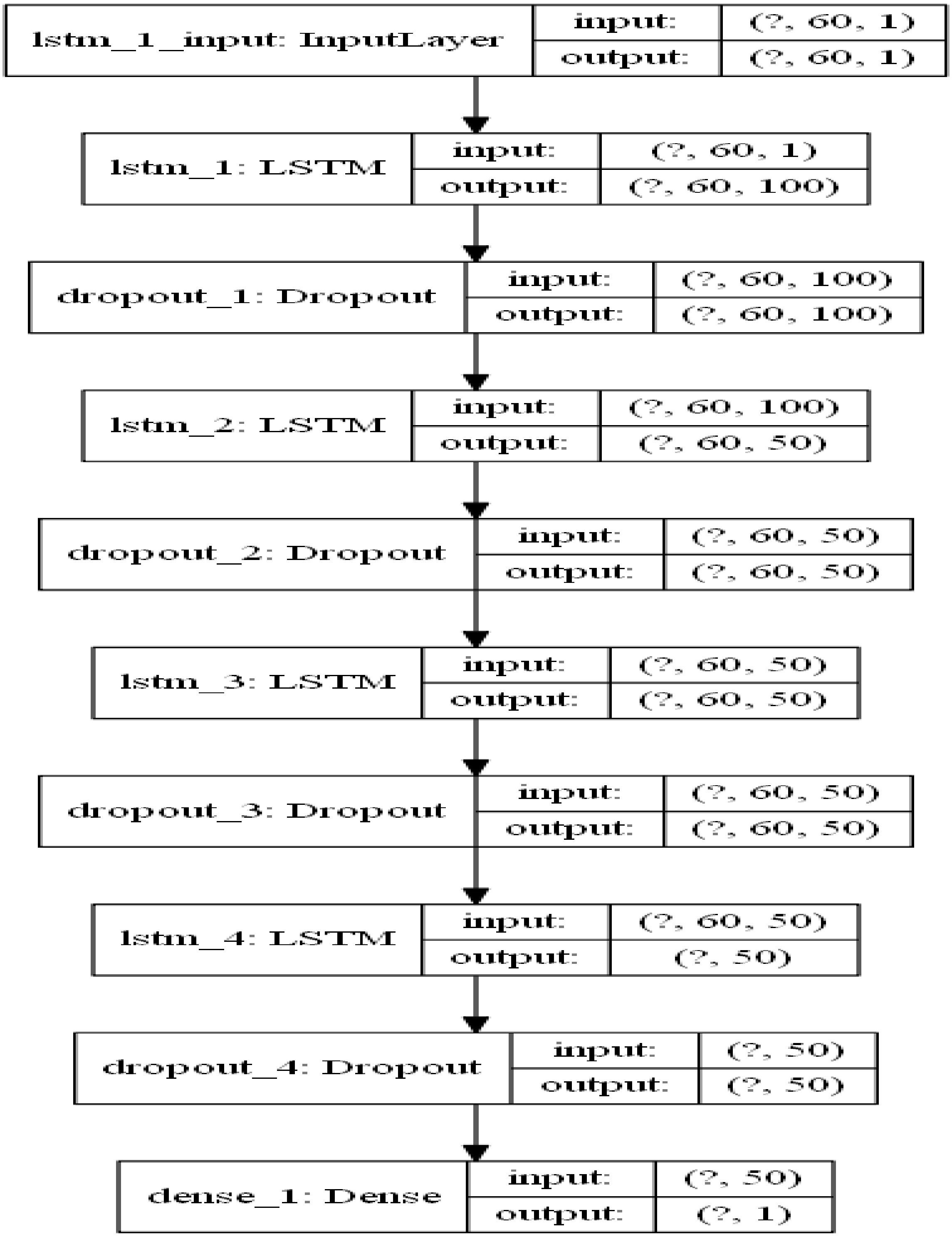


4. ADDING TIMESTEPS.

LSTM network needs the data imported as a 3D array. To translate this 2D array into a 3D one, we use a short [timestep] to loop through the data and create smaller partitions and feed them into the model. The final array is then reshaped into training samples, x number of timesteps, and 1 feature per step. The code below represents this concept:

```
time_steps = 3
for i in range(time_steps, train_scaled.shape[0]):
    x_train.append(train_scaled[i - time_steps:i])
    y_train.append(train_scaled[i, 0])
```

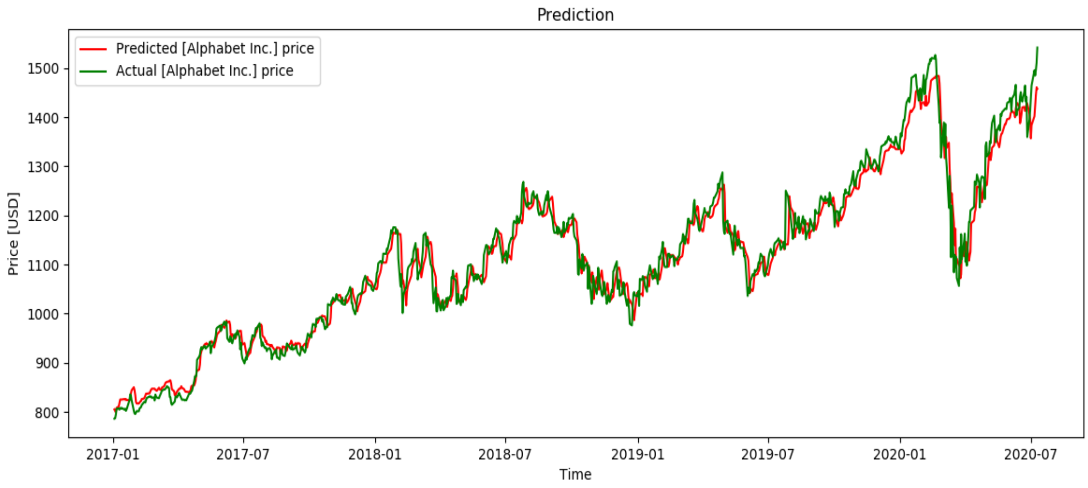
5. CREATION OF LSTM MODEL.



```python
def create_long_short_term_memory_model(x_train):
    model = Sequential()
    # 1st layer with Dropout regularisation
    # * units = add 100 neurons is the dimensionality of the output space
    # * return_sequences = True to stack LSTM layers so the next LSTM layer has a three-dimensional sequence
    # * input_shape => Shape of the training dataset
    model.add(LSTM(units=100, return_sequences=True, input_shape=(x_train.shape[1], 1)))
    # 20% of the layers will be dropped
    model.add(Dropout(0.2))
    # 2nd LSTM layer
    # * units = add 50 neurons is the dimensionality of the output space
    # * return_sequences = True to stack LSTM layers so the next LSTM layer has a three-dimensional sequence
    model.add(LSTM(units=50, return_sequences=True))
    # 20% of the layers will be dropped
    model.add(Dropout(0.2))
    # 3rd LSTM layer
    # * units = add 50 neurons is the dimensionality of the output space
    # * return_sequences = True to stack LSTM layers so the next LSTM layer has a three-dimensional sequence
    model.add(LSTM(units=50, return_sequences=True))
    # 50% of the layers will be dropped
    model.add(Dropout(0.5))
    # 4th LSTM layer
    # * units = add 50 neurons is the dimensionality of the output space
    model.add(LSTM(units=50))
    # 50% of the layers will be dropped
    model.add(Dropout(0.5))
    # Dense layer that specifies an output of one unit
    model.add(Dense(units=1))
    model.summary()
    tf.keras.utils.plot_model(model, to_file=os.path.join(project_folder, 'model_lstm.png'), show_shapes=True
                              show_layer_names=True)
    return model
```

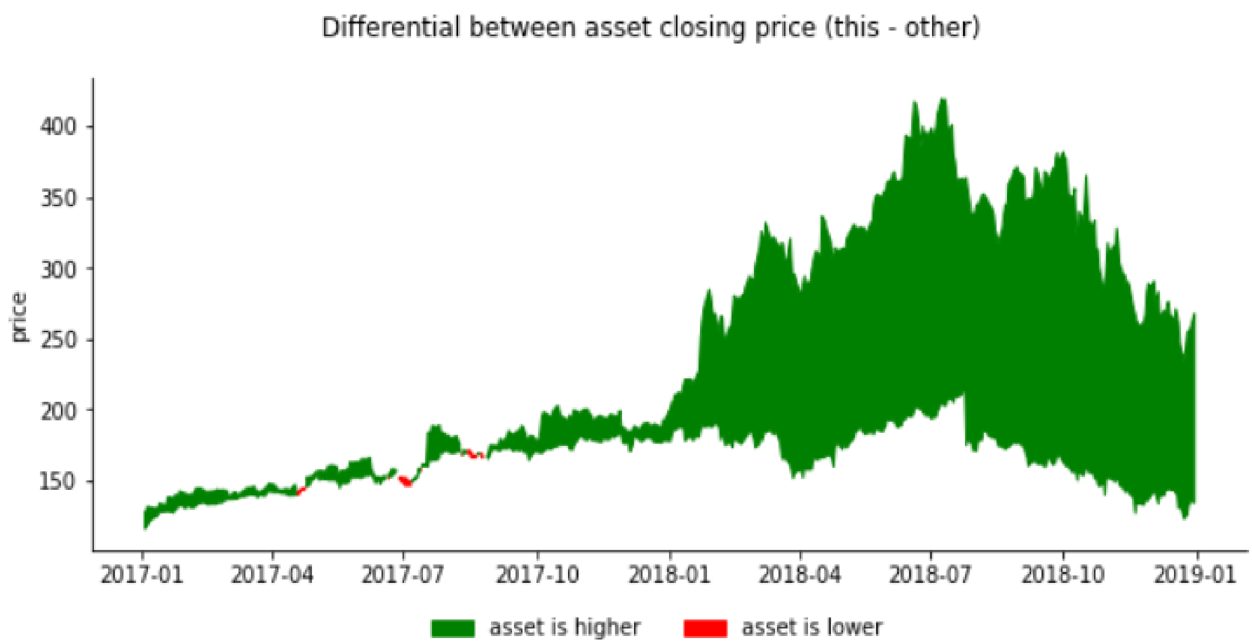
6. MAKING PREDICTIONS HAPPEN.

Now we can call the predict method which will allow us to generate the stock prediction based on the training done over the training data. As a result, we will generate a CSV file that contains the result of the prediction and also a chart that shows what's the real vs the estimation.



FEASIBILITY STUDY

Artificial intelligence plays an important role in the financial industry. AI and ML techniques are being used to solve real-world problems with high effectiveness, maximum accuracy, and less time. Nowadays, investment in the stock market is a lucrative option in which the power of ML is used to predict the moment of stock using sentiment analysis. Stock market prediction is a challenging task because it influences by various factors such as the sentiment of investors, the concert of stock, economical factors, and social media sentiments. This paper gives an overview of flavors of sentiment and stock price prediction along with developments and applications of AI in the Financial Sector. Some methods are used for stock market prediction like Linear Regression, k-Nearest Neighbors, and "Long Short Term Memory (LSTM) ARIMA model Recurrent Neural Network (RNN).



FACILITIES REQUIRED FOR PROPOSED WORK

This Project uses the following software and python libraries:

a. **PYTHON 3.9.7**
b. **NUMPY**
c. **PANDAS**
d. **KERAS.**
e. **TENSORFLOW.**
f. **GOOGLE COLLABS/JUPYTER NOTEBOOK.**

The model proposed in this project is

a. **LSTM MODEL.**
b. **ARIMA MODEL**
c. **YAHOO FINANCE MODEL.**

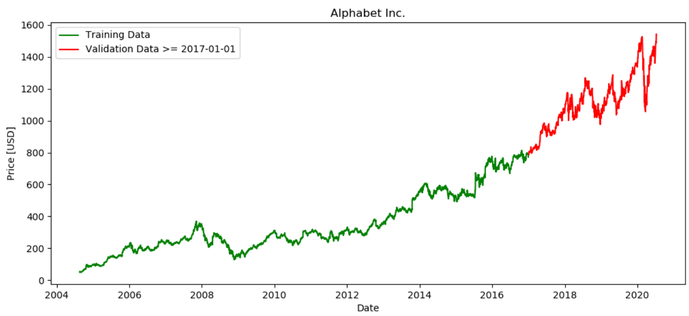
For the Web application, we have used:

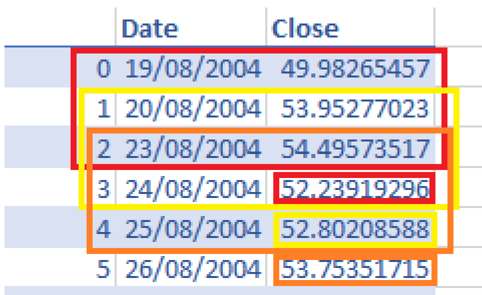
a. **HTML, CSS, and javascript.**
b. **NodeJS.**
c. **Flask, Django for Backend Development.**

Database proposed for this project:

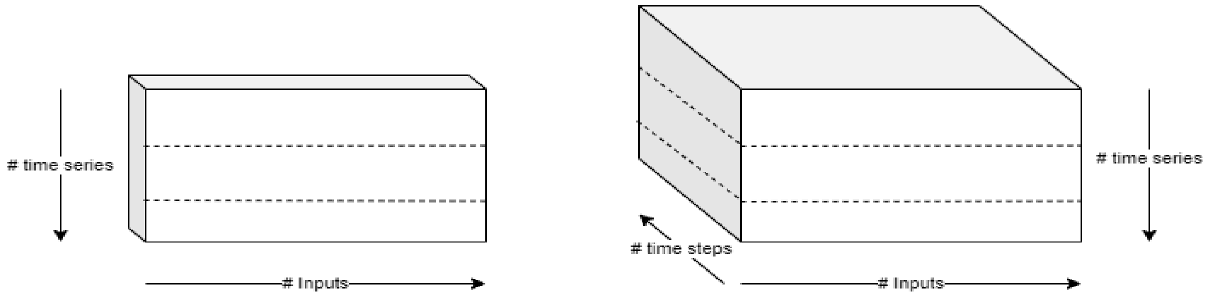
a. **MongoDB.**
b. **SQL(ORACLE).**

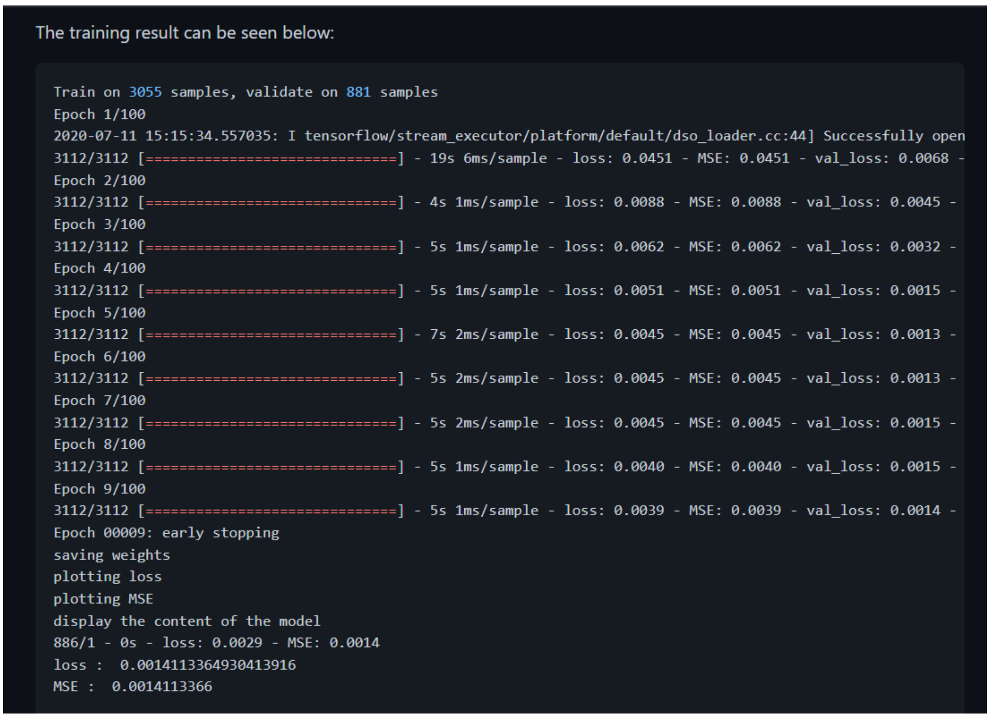
EXPECTED OUTCOMES(SCREENSHOTS)





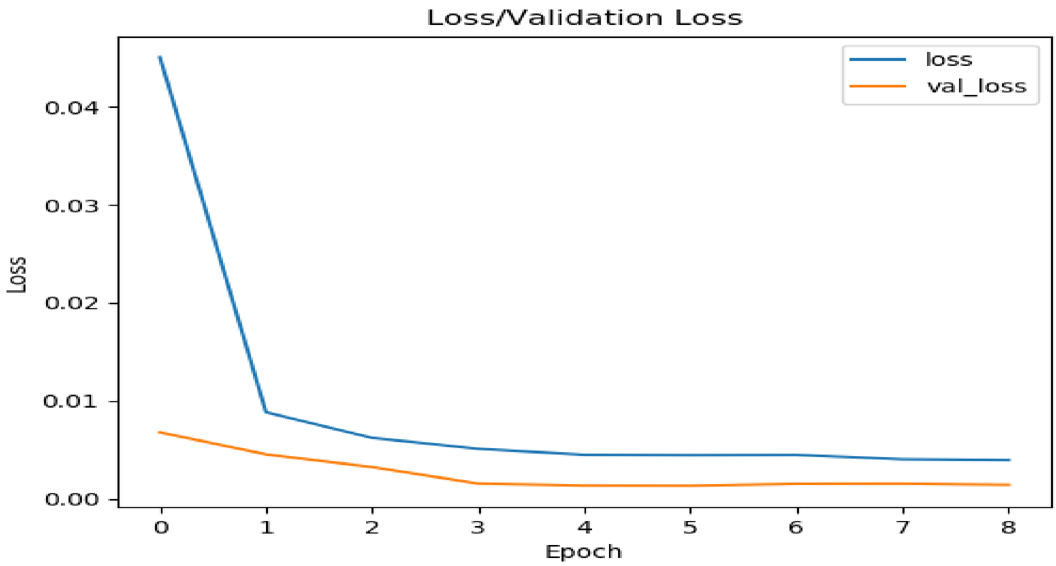
	Date	Close
0	19/08/2004	49.98265457
1	20/08/2004	53.95277023
2	23/08/2004	54.49573517
3	24/08/2004	52.23919296
4	25/08/2004	52.80208588
5	26/08/2004	53.75351715

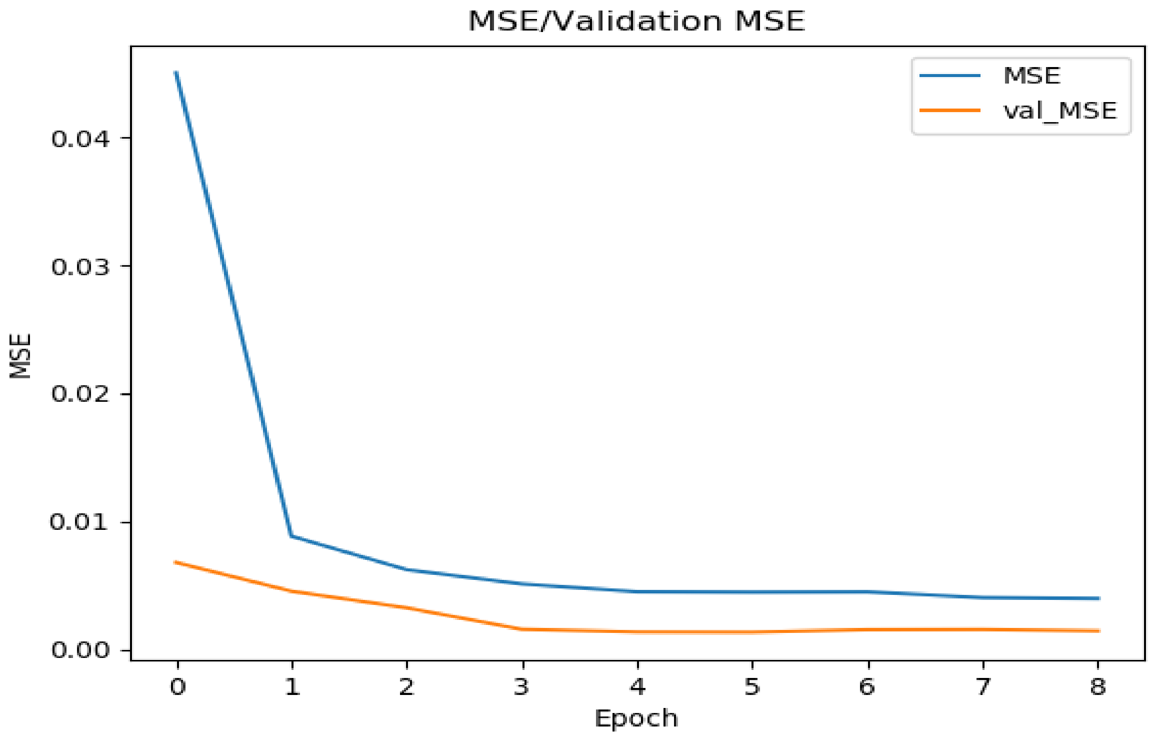




The training result can be seen below:

```
Train on 3055 samples, validate on 881 samples
Epoch 1/100
2020-07-11 15:15:34.557035: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully open
3112/3112 [==============================] - 19s 6ms/sample - loss: 0.0451 - MSE: 0.0451 - val_loss: 0.0068 -
Epoch 2/100
3112/3112 [==============================] - 4s 1ms/sample - loss: 0.0088 - MSE: 0.0088 - val_loss: 0.0045 -
Epoch 3/100
3112/3112 [==============================] - 5s 1ms/sample - loss: 0.0062 - MSE: 0.0062 - val_loss: 0.0032 -
Epoch 4/100
3112/3112 [==============================] - 5s 1ms/sample - loss: 0.0051 - MSE: 0.0051 - val_loss: 0.0015 -
Epoch 5/100
3112/3112 [==============================] - 7s 2ms/sample - loss: 0.0045 - MSE: 0.0045 - val_loss: 0.0013 -
Epoch 6/100
3112/3112 [==============================] - 5s 2ms/sample - loss: 0.0045 - MSE: 0.0045 - val_loss: 0.0013 -
Epoch 7/100
3112/3112 [==============================] - 5s 2ms/sample - loss: 0.0045 - MSE: 0.0045 - val_loss: 0.0015 -
Epoch 8/100
3112/3112 [==============================] - 5s 1ms/sample - loss: 0.0040 - MSE: 0.0040 - val_loss: 0.0015 -
Epoch 9/100
3112/3112 [==============================] - 5s 1ms/sample - loss: 0.0039 - MSE: 0.0039 - val_loss: 0.0014 -
Epoch 00009: early stopping
saving weights
plotting loss
plotting MSE
display the content of the model
886/1 - 0s - loss: 0.0029 - MSE: 0.0014
loss :  0.0014113364930413916
MSE :  0.0014113366
```


Loss/Validation Loss
loss
val_loss
Loss
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
Epoch


MSE/Validation MSE
MSE
val_MSE
MSE
0.04
0.03
0.02
0.01
0.00
0
1
2
3
4
5
6
7
8
Epoch

CONCLUSION

Artificial Intelligence plays an important role in the development of science and technology it is broadly used in each aspect of finance but there are corresponding challenges in applying artificial intelligence. Therefore, the financial system should understand artificial intelligence and make its system more perfect. In order to design a complete artificial intelligence, it is necessary to set principle rule which aims to guide the complete procedure of artificial intelligence in development, designing, management, and control. Research in the stock market has increased rapidly. the stock market prediction correctness can be improved to give a more accurate outcome by using sentiments, ANN, and machine learning.

REFERENCES.

- **RT Bayes. An essay towards solving a problem in the doctrine of chances. Resonance, 2003, 8: 80- 88.**
- **[2]. L Hodgkinson, E Walker. An expert system for credit evaluation and explanation. Consortium for Computing Sciences in Colleges, 2003, 19(1): 62- 72.**
- **[3]. LY Shue, CW Chen, W Shiue. The development of an ontology-based expert system for corporate financial rating Expert Systems With Applications, 2009, 36(2): 2130-2142.**