Stock Price prediction using Big Data Visualization Technique

1st Nibedita Roy
Department of Computer Science
Kent State University
Kent, United States
nroy2@kent.edu

2nd Priyanka Vyas
Department of Computer Science
Kent State University
Kent, United States
pvyas2@kent.edu

***Abstract*—The abnormality of the financial market prevents simple models from accurately forecasting future asset values, so investing in a portfolio of assets has never been simple. The prevailing trend in research currently is machine learning, which involves training algorithms to execute tasks that would typically need human intelligence. In order to forecast future stock market prices, this study uses a Long-Short Term Memory (LSTM) model. The major goal of this study is to determine the degree of prediction precision that a machine learning algorithm can achieve as well as the degree to which epochs can enhance our model.**

***Index Terms*—Time series data analysis, machine learning, visualization, prediction.**

I. Introduction

A financial instrument known as a stock indicates a company's ownership interest. A share of stock is a fraction of a corporation, which an individual purchases. Stocks of companies are purchased by investors who anticipate an increase in value. The stock can then be sold for a profit if this happens because it increases the worth of the firm as well. Businesses can raise money to invest in their operations and grow by issuing shares. Stocks are a means for investors to increase their capital and outpace inflation over time.

The stock market is characterized as volatile, dynamic, and nonlinear. Earlier, investors used to adopt the traditional approach, which involved seeking assistance from stockbrokers to buy or sell a stock. Still, With the evolution of technology and more advanced and accurate prediction methods, stock market analysis has gained the attention of businessmen and traders to invest at a low-risk rate. The final outcomes from enormous data sets are predicted using machine learning algorithms. In this paper, the Long Short-Term memory model has been implemented, and the Time series prediction technique has been implemented using the LSTM model.

II. Project Description

A. Proposed Technique

As represented in the previous section getting historical data from the market is a mandatory step. Then there is a need to extract the feature which is required for data analysis, then divide it into testing and training data, training the algorithm to predict the price, and the final step is to visualize the data.

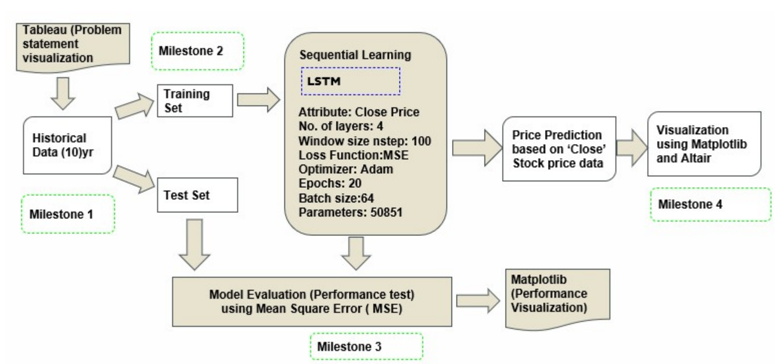

Figure I: Prediction Model Approach

Fig I shows the Prediction Model approach adopted in this study that has been implemented in four Milestone. Milestone 1 as data collection, Milestone 2 as data pre-processing, Milestone 3 as LSTM model design and implementation and Milestone 4 as Result Visualization.

In this study, initially to understand and analyze the problem statement through a dataset, initially Tableau is used to visualize the yearly price movement trend for selected individual stocks and a comparative study is visualized for Tesla's competitors. The data for this study is downloaded from Nasdaq and compiled separately for visualization purposes.

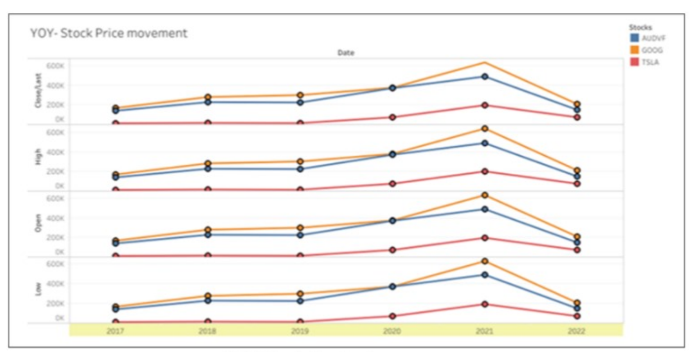

Fig1: Yearly Stock price movement

The individual stock selected for this study is Tesla stocks. Tesla stocks is an extensive index traded on the New York stock exchange and is a well-known stock. The plot above shows the yearly trend visualization of price movement for Stocks of Tesla, Audi, and Google. The dataset for the plot is a stock price history .csv file, which is gathered and compiled for the time period of 2017 to 2022. The dataset contains daily stock high, low, open, and closing prices. It can be observed in Fig 1. plot that the stock price reaches a new high and low every time as the stock market is sensitive to the political and macroeconomic environment. Some features cannot be included in the data and can be considered as noise which means an incomplete information gap between past stock trading price and volume with a future price. The graph shows a rebound in 2021 post covid and again a low in 2022.

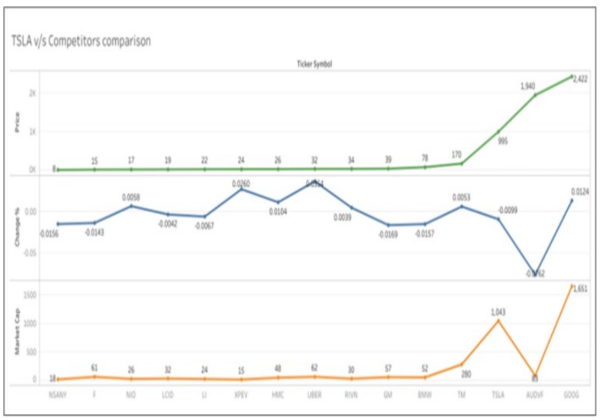

Fig 1.1: Tesla Competitors Comparative study of Stock price movement

Fig 1.1: Tesla Competitors Comparative study of Stock price movement

The graph above shows Fig 1.1. plot having stock data dated until 12/4/2022 It is a comparative study of the parameters using Tableau to study the stock price movement in line with companies Market Cap, percent change in stock price and closing stock price (i.e. Avg close).

While devising this comparison as the focus was mainly on individual stocks i.e.Tesla, the plot shows a comparative between direct competitors of Tesla i.e. from the automobile industry and Indirect competitors from IT services. The plot shows a rise in stock prices of Tesla, Audi and Google and compared to the market capital of Tesla and Audi, Tesla has more capital infusion compared to peers as Tesla works on the ability to generate recurring sales and potential to add to profits. Tesla is an indirect competitor of Uber and Google and the reason is because Uber and Google are also working on self-driving car projects. Through initial assessment using tableau graphs it was observed that accurate stock price prediction is challenging because of multiple factors that affect the stock price, like political, global economic conditions, unexpected events, a company's financial performance etc. but it also crucial to predict as the supply-demand balance is driven by market sentiment which in turn affects the stock price trend.

Through stock price prediction investors can increase the rate of investment and business opportunities in the stock market by devising an algorithm to predict the short term price of individual stock. In this study, a novel method is proposed to predict and visualize stock price for individual Tesla stocks using LSTM neural network. The stock analysis in this project is based on a 10-year stock trading record of Tesla Inc., where we load and read the data, clean and prepare it for further analysis, then visually portray the data analysis , trends , forecasting using various ways with the pandas and seaborn libraries.

III. Data Analysis and Prediction Method

Stock data is time series data whose prediction dependency lies on four types of components: a trend component, a seasonal component, a cyclical component, and an irregular component. The stock data study requires a specific way of analyzing a sequence of data points collected over an interval of time. In this study, the historical time series stock data tracks the movement of selective data points, like, security price, over a specified period of time (10 years) with data points recorded at regular intervals.

The first step in the Machine learning model is the Data preparation stage which includes sub stages i.e. Data collection and Data Pre-processing. The data is collected from the source and is pre-processed using transformation techniques where the raw data is modified into desired format. The second step in the process is the data training and learning stage which does the feature extraction. The final step is prediction analysis and data visualization where the trained model identifies the pattern that fits on the historical data and uses it to predict future value over a period of time. The process develops a model based on historical data and applies it to predict future value of stocks.

A. Data Collection

The financial data is mostly available in either structured or unstructured format and is accessible as historical data or real-time data in the public domain. The real-time data is not available from public APIs and requires to be purchased. Hence, structured historical data with attributes date, open price, high price, low price, closing price, Adj close price and volume is used in this study.

The code for data collection and download has been shared in the git repository for perusal. In this study the framework explored for coding are python, tensorflow using google collaboratory and the libraries like panda, panda-dataframe reader,numpy and yahoo finance(yfinance). The library pandas organizes data in data frames i.e. table structures and pandas-data reader is used to import the financial data from public domain into python data frames. YahooFinance ('yfinance') provides historical stock prices from yahoo finance and is one of the widely used API for stock data study as it provides the data from New York Stock Exchange (NYSE) and Nasdaq Stock Exchange (Nasdaq). Then, a ticker object for data (i.e.

TSLA-Tesla stock) that we want to get for is created to get the financial data information related to the stock. The function yf.ticker.history() is used to specify the argument period i.e. 10 years, for which the historical data is required in the study.

In this study, it is learned that using pip install ‘yfinance’ library installation is easy and python friendly also it provides data for free whereas Quandi and Nasdaq requires registration and provides data for a price through their premium and free options. The coding for this stage was explored using Nasdaqdatalink and Quandi also, but data fetching from the source failed repeatedly due to API, ticker or path issues and the historical stock price movement is visualized by plotting a line chart using Matplotlib plot method.

B. Features of Dataset

Date	Open	High	Low	Close	Adj Close	Volume
2010-06-29	19	25	17.540001	23.889999	23.889999	18766300
2010-06-30	25.790001	30.42	23.299999	23.83	23.83	17187100
2010-07-01	25	25.92	20.27	21.959999	21.959999	8218800
2010-07-02	23	23.1	18.709999	19.200001	19.200001	5139800
2010-07-06	20	20	15.83	16.110001	16.110001	6866900
2010-07-07	16.4	16.629999	14.98	15.8	15.8	6921700
2010-07-08	16.139999	17.52	15.57	17.459999	17.459999	7711400
2010-07-09	17.58	17.9	16.549999	17.4	17.4	4050600

The dataset features are stock prices with data points Open,High,Low, Close, Adj Close and Volume of stocks. The total sample size for this study is 2517 for a period of 10 years.

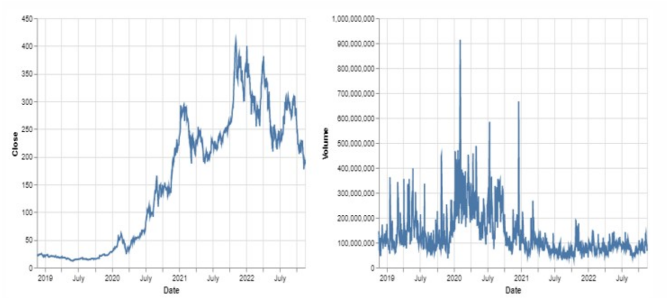

Fig 2.a: Visualization plot for close price and Stock volume versus Date

Fig: 2.b: Visualization plot for Tesla v/s Competitor close price and Avg close price versus Date.

The library Altair is used to visualize the above Fig 2a plot of Close price and Volume versus Date. Altair makes it simple to have multiple plots in one visualization. As we can see in the slide we can plot closing price movement and volume for TSLA stocks using a logical operator.The data is for time period 2018-2022 as Altair has a limitation to visualize upto 5000 records. Fig 2b is a comparative plot for companies General motor, Toyota motor and Tesla is also visualized for a comparative study in automobile industry.

In this study to design a prediction model, two Models i.e. Model 1 with training size 65 percent and Model 2 with training size 80 percent and remaining is the test size. As observed in the comparative plots earlier the feature ’Close Price’ shows the price of the stock at closing time, hence in this study close price is considered to study the price movement and price prediction.

C. Data Preprocessing and Training

Using Matplotlib plot method a line chart is created for the historical close prices of TSLA and the plot shows an upward trend in terms of the rise in the stock price. To build an LSTM model for price prediction, first split the dataset into training set and test set and then normalize the data using Scikit-Learn MinmaxScaler function to transform the values so that the values are ranged from 0 to 1 and then reshape the normalized data into two dimensional array.

Further split the dataset such that the training size of the dataset is 65 percent of closing prices extracted from total length of data frame and the test size is the difference between the total dataset length and training size as training size and test size resp. Convert this array of values into a dataset matrix by considering a window size n step starting from first sample of data and ending at n step’th sample from the end such that for each time step, LSTM will take n steps-1 samples for training and predict the last sample.

An empty list for a sequence of feature data(dataX) and sequence of label data (data Y) is created and for the window size of n step it is converted into a Numpy array which is an acceptable data format for Tensorflow to train a neural network model. The n step considered to train the model is a 100 days window of historical prices and reshape this data again into a three-dimensional array which is the acceptable input data for the LSTM model. The train (X train,y train) and test(X test,y test) data in a three dimensional array is ready as an input for the LSTM model.

D. Design LSTM model and Algorithm

All important and relevant libraries required to train the model are imported during Milestone 2. Tensorflow, which is an open source machine learning library, is imported to set up the LSTM network architecture.Long Short-Term Memory (LSTM) is one type of recurrent neural network which is used to learn order dependence in sequence prediction problems. Due to its capability of storing past information, LSTM is very useful in predicting stock prices. This is performed by feeding back the output of a neural network layer at time t to the input of the same network layer at time t + 1.

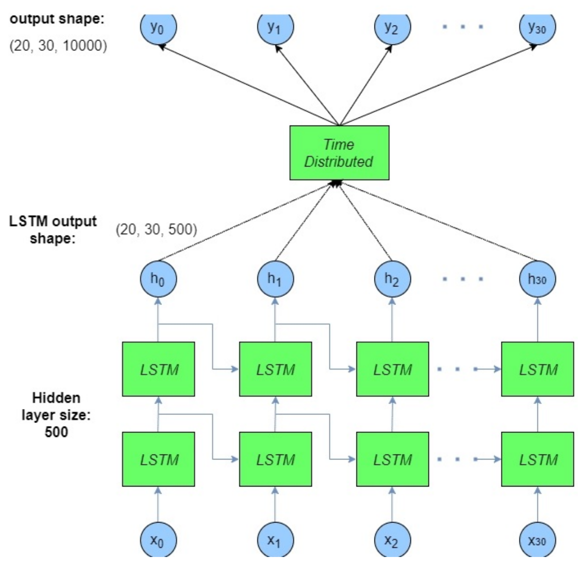

Fig. 3 LSTM Architecture.

This section will show the architecture of the LSTM network in Keras and describe what a complete LSTM architecture looks like. During back propagation, a recurrent neural networks suffer from the vanishing gradient problem. Gradients are values used to update a neural networks weights. The vanishing gradient problem is when the gradient shrinks as it back propagates through time. If a gradient value becomes extremely small, it doesn't contribute too much learning. Long Short-Term Memory (or) LSTMs are widely used for sequence prediction problems and have proven to be extremely effective. The reason they work so well is because LSTM is able to store past information that is important and forget the information that is not. LSTM has three types of gates: 1. The input gate: The input gate adds information to the cell state. 2. The forget gate: It removes the information that is no longer required by the model. 3. The output gate: Output Gate at LSTM selects the information to be shown as output.

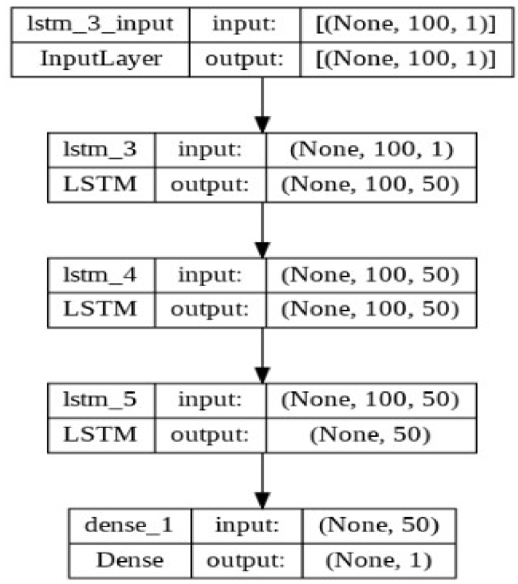

Fig. 3.1 Model 1 LSTM Architecture.

Fig 3.1 above is the Model 1 LSTM architecture with layerwise input and output shape. The three inputs control how quickly data enters and leaves the cell as the cell collects values over arbitrary time intervals. The LSTM's ability to learn context-specific temporal dependency is its key benefit. Without explicitly applying the activation function within the recurrent components, each LSTM unit gathers data for either a lengthy or brief duration. The LSTM cell's overhead door is in charge of both the loads and the ability to start the cell state. Consequently, loads can reach their ideal quality in a reasonable amount of time and data from a previous cell state can flow through a cell unchanged as opposed to swelling or contracting exponentially at each time-step or layer. Since the value stored in the memory cell is not iteratively modified, the inclination does not evaporate when prepared using back engendering, allowing LSTMs to handle the evaporating slope issue.

The data's input shape is as follows: (batch size, number of time steps, hidden size). In the Fig 3.1 above, the LSTM network is depicted as unrolled over all the time steps. The input data is then fed into the "stacked" layers of LSTM cells. These unrolled cells' output is still produced (batch size, number of time steps, hidden size). In this the time step for Model 1 is 100 and Model 2 is 60. The output is compared to the training y data for each batch, and Keras then propagates the error and gradient from that point. In this instance, the input x as showing in Fig 3 sample architecture have been advanced one time step to serve as the training y data; in other words, the model is attempting to predict the very next word in the sequence at each time step. The output layer, however, has the same number of time steps as the input layer because it performs this at every time step.

To build the model, first define the Sequential model that consists of the four layers. The first and second added for LSTM layer has 50 network units in the model and the return sequence is set to true so that the output of the layer will be another sequence of the same length. The third layer added is again an LSTM layer with 50 network units and lastly the densely connected layer that provides the output of one network unit. The summary of the LSTM model in Fig 4 is as below:

```
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

Figure 4: Model 1 Summary

As shown above the total trainable parameters are 50,851 and to train the model by fitting it with the training set,first compile the model and set adam optimiser and a loss function MSE i.e. Mean square error. Finally, the model is trained by fitting it with the training set and running it for 20 epochs and batch size 64 to calculate the validation loss and training loss. With limited parameters and the loss and val loss values it can be said the designed model is a good model.

```
Epoch 20/20
24/24 [==============================] - 5s 199ms/step - loss: 1.8721e-06 - val_loss: 0.0019
```

For the next milestone, the focus is to assess the performance of the trained model and visualize the results in line with the epochs and model loss.

E. Model Evaluation

Stock price analysis is an example of time series analysis and is one of the primary area in predictive analytics. The time series data contains measurements or observations attached to sequential time steps. In case of stock prices, the time stamps can seconds, minutes or days.

In this project we have used pandas for data manipulation and altair, matplotlib for data visualization. For visualization through altair we have used the data reader API of Pandas. We first import the dependencies necessary for the project and then used data reader to create Pandas dataframe that contain stock price data.

For performance evaluation, we have build two models using LSTM and added layers.

Model I

```
Layer (type)         Output Shape        Param #
================================================
lstm (LSTM)          (None, 100, 50)     10400

lstm_1 (LSTM)        (None, 100, 50)     20200

lstm_2 (LSTM)        (None, 50)          20200

dense (Dense)        (None, 1)           51

================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
```

Model II

```
Layer (type)         Output Shape        Param #
================================================
lstm (LSTM)          (None, 60, 100)     40800

lstm_1 (LSTM)        (None, 100)         80400

dense (Dense)        (None, 25)          2525

dense_1 (Dense)      (None, 1)           26

================================================
Total params: 123,751
Trainable params: 123,751
Non-trainable params: 0
```

Fig 5: Model 1 and Model 2 Layers and Parameter size

As shown in Fig 5 above, Model 1 has 3 LSTM layers with 50,50 and 50 neurons resp and 1 Dense layer with output dimension 1. Also, our training size is 65 percent and the test size is 35 percent and our total parameter is 50,851. Model 2 has 2 LSTM layers with 100, 100 neurons resp and 2 Dense layer with output dimension 25 and 1 respectively. So, it is observed when the time step is decreased and training size is increased to 80 percent data size then the total parameter is increases.

Further, Model 1 is updated with with a batch size of 64 and epochs 20, the plots shows that after a certain epochs the model manages to follow the trend and Model 2 is updated with with a batch size of 20 and epochs 3, the plots shows that the model manages to follow the trend from the first epoch run.

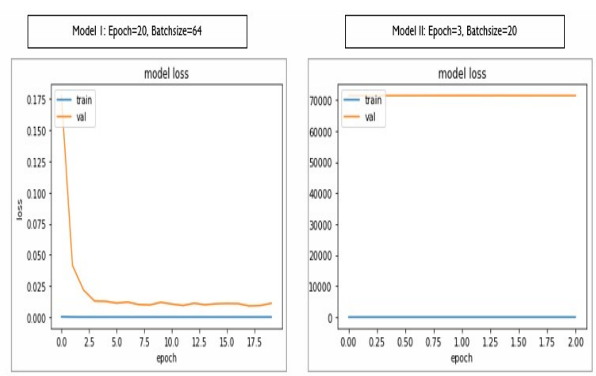

Fig 6. Epoch vs Batchsize plot for model 1 and Model 2.

The plots in the figure 6 above are the diagnostic plot that shows the model behaviour. All result and analysis plots has been made using Matplotlib library. The training loss v/s val loss. It is observed that the performance of the model is good on both the train and validation set.The plot shows the train and val loss decrease and stabilise around the same point. The plot of train and validation loss for model 2 in fig 6 is showing the characteristics of an underfit model. The performance of the model may be improved by increasing the capacity of the model, such as number of neurons in the layers or number of hidden layer(LSTM layer).

Fig 7. Model 1 and Model 2 evaluation using Loss Function RMSE.

We evaluated the model for predictions using the loss function root mean square error (Fig 7) and it is observed that for Model 1 = RMSE is 162 which is higher. Here, our test set value is 199, and the train set value is 17 which means the model badly overfits the data but it tests well in the sample and has little predictive value when tested out of the sample size, and Model 2= RMSE is 5.95 that shows the model is able to fit the dataset and is giving better predictions. RMSE score shows how far the prediction falls from the measured true values using Euclidean Distance. The RMSE statistic provides information about the short-term performance of a model by allowing a term-by-term comparison of the actual difference between the estimated and the measured value. The smaller the value, the better the model's performance.

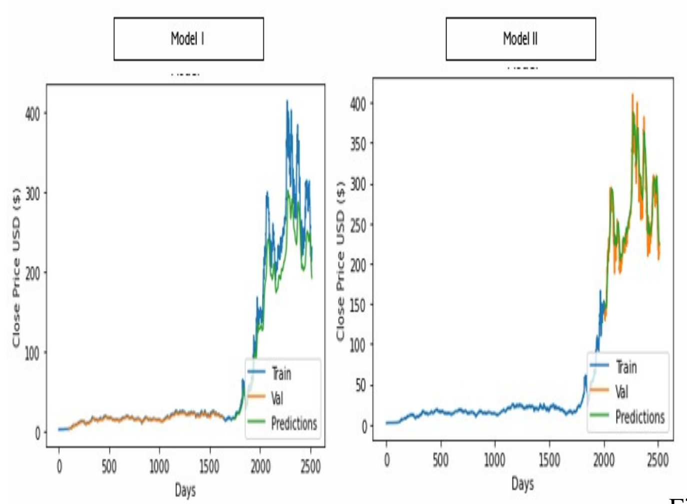

Figure 8: Model 1 and Model 2 prediction plot

From the above plot Fig 8, it can be summarized that the both model performed fairly good. Here, we can follow the unexpected jumps and drops. However, for the most recent data stamps we can see that the model predicted lower values compared to real values of the stock price.

To further evaluate the performance and efficiency of models,the training and val loss for Model 1 and Model 2 is measured for different epochs and batch size as shown in Fig 9 and Fig 10. While assessing the models, it is observed that at different epochs and batchsize Model 2 failed to perform as expected and the resulting plot is a fixed plot similar to plot obtained for epoch=3 and batchsize 20 in Fig (6).

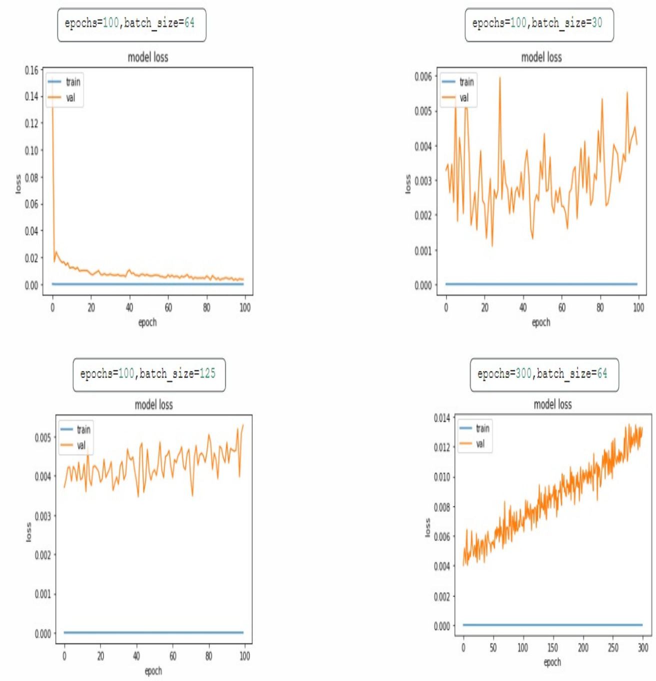

Fig 9. Plot for Epoch vs Loss for Model 1.

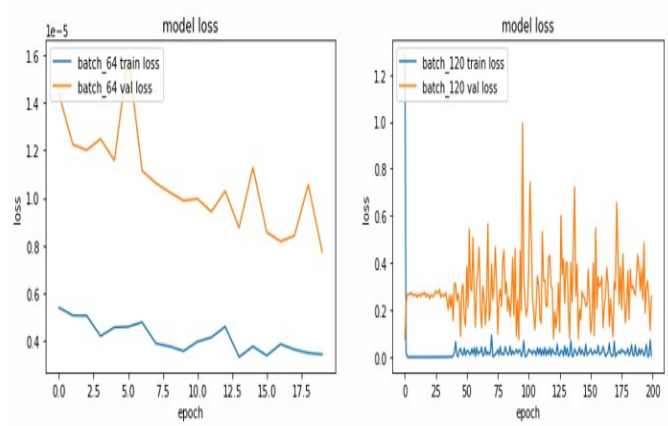

Fig 10. Plot for Batch vs Loss for Model 1.

Hence, Model 1 was further assessed as there variations in the training and val loss is observed for epoch size 100 and different batch sizes. As learned from the above plots as there is an increase in val loss as the batch size is increased or decreased with an increase in epoch size.

As the data size is small the model 1 shows optimal solution with batch size 64 and epochs 20 but with the increase in batchsize the model 1 or epochs size the model 1 shows under fitting. High levels of bias are seen in this underfit model, which may produce incorrect results for both the training set and the test set. Overfit models, on the other hand, have a high variance and provide accurate results for the training set but not for the test set. Less bias is produced with more model training, but variation may rise.

When a model can't generalize and instead fits too closely to the training dataset, it is known as overfitting. The same is observed in the first model when the training data set is (65 percent) too little and does not contain enough data samples to adequately represent all possible input data values or there may be large volumes of unimportant information, or noisy data, present in the training data that resulted into overfitting the model. Also its is observed that in Model 2, the model spends too much time training with just one sample set of data hence there is a possibility that due to the high model complexity, the training data's noise is learned with the increase in number of neurons in each layers respectively.

IV. Conclusion

This paper develops a model and program for stock prices prediction using data from Yahoo finance. Efficient and accurate prediction systems for stock prices help traders, investors, and analyst by providing supportive information like the future direction of the stock market. For future work, for LSTM model, we will use short term historical data to test the accuracy where we will use 2 years data compared with 10 years data to test the accuracy. Also, this method can further be improvised by using a larger data set and implementing transformer model for better prediction accuracy and good performance.

References

[1] K. Ullah and M. Qasim, "Google Stock Prices Prediction Using Deep Learning," 2020 IEEE 10th International Conference on System Engineering and Technology (ICSET), 2020, pp. 108-113, doi: 10.1109/ICSET51301.2020.9265146.

[2] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar,Stock Closing Price Prediction using Machine Learning Techniques, Procedia Computer Science,Volume 167,2020,Pages 599-606,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2020.03.326.

[3] Rouf, N.; Malik, M.B.; Arif, T.; Sharma, S.; Singh, S.; Aich, S.; Kim, H.-C. Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions. Electronics 2021, 10, 2717.

[4] Hu, Z.; Zhao, Y.; Khushi, M. A Survey of Forex and Stock Price Prediction Using Deep Learning. Appl. Syst. Innov. 2021, 4, 9. https://doi.org/10.3390/asi4010009

[5] Banerjee, S., Mukherjee, D. (2022). Short Term Stock Price Prediction in Indian Market: A Neural Network Perspective. Studies in Microeconomics, 10(1), 23–49. https://doi.org/10.1177/2321022220980537

[6] Chandola, D., Mehta, A., Singh, S. et al. Forecasting Directional Movement of Stock Prices using Deep Learning. Ann. Data. Sci. (2022). https://doi.org/10.1007/s40745-022-00432-6.

[7] Er, X., Sun, Y. (2021, July). Visualization Analysis of Stock Data and Intelligent Time Series Stock Price Prediction Based on Extreme Gradient Boosting. In 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE) (pp. 272-279). IEEE.

[8] K. C. Yang, C. Y. Chianglin, C. -H. Huang and I. H. Chen, "Knowledge Discovery and Data Visualization for Taiwan Stock Market: Using F-Score Analysis," 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2019, pp. 1512-1515, doi: 10.1109/IEEM44572.2019.8978817.

[9] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8),1735-1780.

[10] Aldhyani, T.H.H.; Alzahrani, A. Framework for Predicting and Modeling Stock Market Prices Based on Deep Learning Algorithms. Electronics 2022, 11, 3149. https://doi.org/10.3390/electronics11193149