STOCK MARKET PREDICTION USING LSTM NEURAL NETWORKS

[1]P. Akhila, [2]Dr.P.Krishna Subba Rao, [3]CH.Avinash
[1]Student,[2] Professor, [3]Asst. Professor
[1]Department of Computer Science and Engineering (CSE)
[1]Gayatri Vidya Parishad College of Engineering, Visakhapatnam, India

Abstract: Predictive analytics has been developed in significance along with the emergence of big data. In the past few decades, forecasting of stock market is gaining more attention as the profitability of investors in the stock market mainly depends on the predictability. To predict the stock prices there are many algorithms used. Recurrent neural nets are a specific type of ANNs (artificial neural networks) aimed to recognize patterns in sequences of data, such as text, genomes, handwriting, the spoken word, or time series data originating from sensors, stock markets and government agencies. Many recent works on Google stock prediction are implemented using GRU (Gated recurrent units). The limitation of the existing work using GRU is it has a high loss rate. So, for efficiency and accuracy, our proposed method uses LSTM (long short-term memory). LSTMs help conserve the error that can be back propagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps. The efficiency of the proposed method is proved by comparing the performance of it with the previous method using GRU.

Keywords: Predictive analysis, big data, recurrent neural nets, ANN, genomes, GRU, LSTM

I.INTRODUCTION

Forecasting of stock market is gaining awareness as the profitability of financiers in the stock market is rising day by day. Investors in stock market want to make the most of their profits; it would be advantageous for them if the instant can be predicted beforehand. Stock market prediction is the act of trying to find out the future value of a company stock or other budgeting appliance exchanged on a trade. The conquering forecast of a stock's prospect consequences could yield significant profit. The enlargement of the stock market is an essential both for the assertion of an incessant financially viable expansion and also for the proficient allowance of resources in financial system. In order to advocate the clients Most Investors accomplish accumulation deal, use technical, fundamental or time series scrutiny in trying to predict stock prices. However, these approaches do not usually assurance good returns because they guide on trends and not the most likely consequences. It is therefore mandatory to explore improved ways of prophecy.

Early research about stock forecasting was based on Efficient Market Hypothesis (EMH), and Random Walk Theory (RWT).

1.1. Efficient Market Hypothesis(EMH)

The EMH is a theory in fiscal finances that states that quality prices fully reflect all vacant information. A straight connotation is that it is impossible to "beat the market" reliability on a risk-adjusted basis since market prices should only react to new information. Deeming that news always go behinds a random walk pattern which defines prophecy, it is nearly unattainable to predict stock price with a precision of more than 50%.

1.2. Random Walk Theory(RWT)

The random walk theory states that market and securities consequences are random and not persuaded by past events. The scheme is also called as the "weak form efficient-market hypothesis. "The central scheme behind the random walk theory is that the randomness of stock consequences provides attempts to find worth patterns or take benefit of new information ineffectual. In particular, the theory maintains that day-to-day stock worth is self-governing of each other, meaning that momentum does not generally exist and calculations of past earnings growth does not forecast prospect intensification. The Random Walk Theory or the Random Walk Hypothesis is a mathematical model of the stock market. Advocates of the theory trust that the prices of securities in the stock market develop according to a random walk.

1.3. Artificial Neural Network(ANN)

Artificial neural network is a computational model lying on the constitution and functions of organic neural networks. Information that pours through the network distress the constitution of the ANN because a neural network amends or find out in a sense based on that input and output.

The motivation in the wake of RNNs is to make use of chronological information. In a traditional neural network, it is presumed that all inputs (and outputs) are sovereign of each other. But for many tasks that's a very bad idea. To envisage the next word in a sentence, better known which words came before it. RNNs are called recurrent because they carry out the same task for every factor of a sequence, with the output being depended on the preceding calculations. Another way to imagine about RNNs is that they have a “memory” which incarcerates information about what has been gauged so far. In theory RNNs can make use of information in arbitrarily long successions, but in practice they are limited to looking back only a few steps.

1.4. Recurrent Neural Network(RNN)

In theory, RNNs can devise use of the information in randomly long successions, but in practice they are inadequate to looking back only a few steps. The vanishing gradient problem averts typical RNNs from learning long term dependencies. Gated Recurrent units (GRU) proposed in 2014 to combat the vanishing gradient problem by altering the way in calculating the concealed units.

RNNs have revealed enormous accomplishment in numerous NLP tasks. The mainly widespread used type of RNNs is LSTMs (Long Short-term memory), which are a lot enhanced at confining long-term dependencies than vanilla RNNs. LSTMs are indispensable the same thing as the RNN, But, they just have a diverse way of computing the concealed states. LSTMs were specifically designed to get the vanishing/exploding gradient problem.

II.LITERATURE SURVEY

Recurrent neural networks are unique from feed-forward architectures were scrutinized for sculpting of nonlinear performance of fiscal markets. Recurrent neural networks could be configured with the right selection of parameters such as the number of neurons, the number of epochs, the amount of data and their association with the training data for predictions of economic markets. By investigating of learning and forecasting of the recurrent neural networks is scrutinized the same effect: better learning, which is often illustrated by the root mean square error, does not give assurance for a better prediction. There are such recurrent neural networks settings where the best consequences of nonlinear time series forecasting could be acquired. For the improved enhancement of RNN learning the EVOLINO algorithm would be preferred because it was very clearly shows training and validation of the recurrent neural network for nonlinear data inputs.

2.1. EVOLINO

Schmid Huber established a general formation of succession learning algorithm Evolution of recurrent systems with linear Outputs (EVOLINO). EVOLINO uses evolution to discover good RNN hidden node weights, while using the techniques such as linear regression or quadratic programming to compute finest linear mappings from the concealed state to the output.

When quadratic programming is used to exploit the margin, it is unattainable to acquire the first evolutionary recurrent support vector machines. EVOLINO-based Long Short-term Memory (LSTM) can solve tasks which cannot solve by Echo State nets. There was established a new class of recurrent, truly chronological SVM-like devices with internal adaptive states, trained by a novel technique called Evolution of systems with Kernel-based outputs (EVOKE), an example of the latest EVOLINO class of techniques. EVOKE evolves recurrent neural networks to sense and correspond to temporal dependencies while using quadratic programming/support vector regression and pseudo-inverse regression. EVOKE achieves recent state-of-the-art gradient-based Recurrent Neural Networks (RNNs) on different time series forecasting chores. RNN learning is used for context-sensitive languages recognition. RNN is difficult and frequently increasing difficulty for standard RNNs because it entails unlimited memory resources.

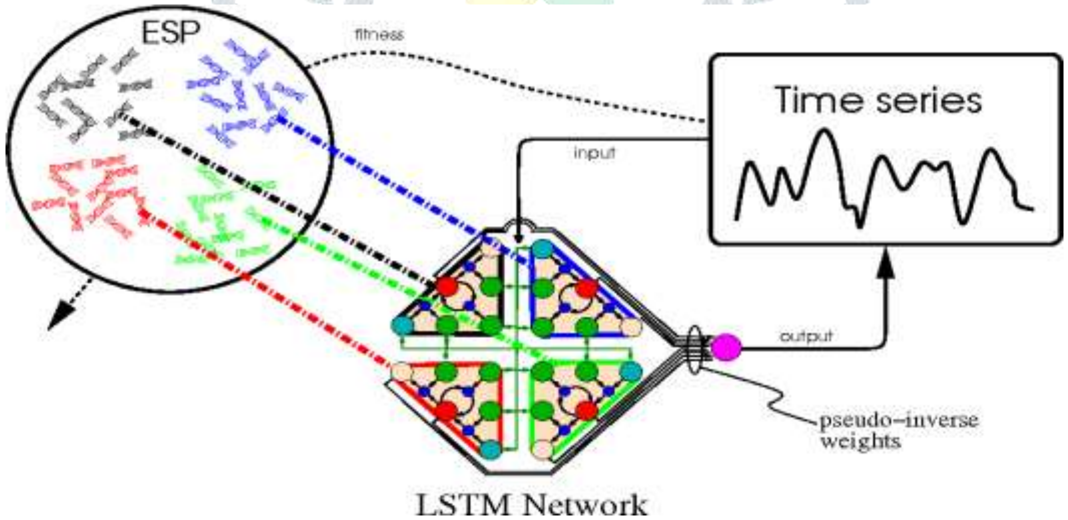

Figure 2.1: Block diagram of EVOLINO

2.2. Multi-Layer Perceptron Model:

Multi-Layer Perceptron (MLP) is also called as back propagation. It is a feed forward neural network which contains one or more layers between input and output layer. MLP plots sets of input data onto a set of suitable outputs. Feed forward means, the data is streaming in a single direction from input to output layer (forward). An MLP contains huge number of layers of nodes in a directed graph and each layer fully attached to the next layer. Except for the input nodes, each node is a neuron which consists of a nonlinear activation function. This network is trained with the help of back propagation learning algorithm. The main use functions of MLPs are pattern classification, recognition, prediction and approximation. Multi-Layer Perceptron can resolve problems which are not linearly separable. MLPs separate classes via hyper planes. MLPs utilize distributed learning. MLPs contain one or more hidden layers.MLP exploits a supervised learning technique called back propagation algorithm for training the data.

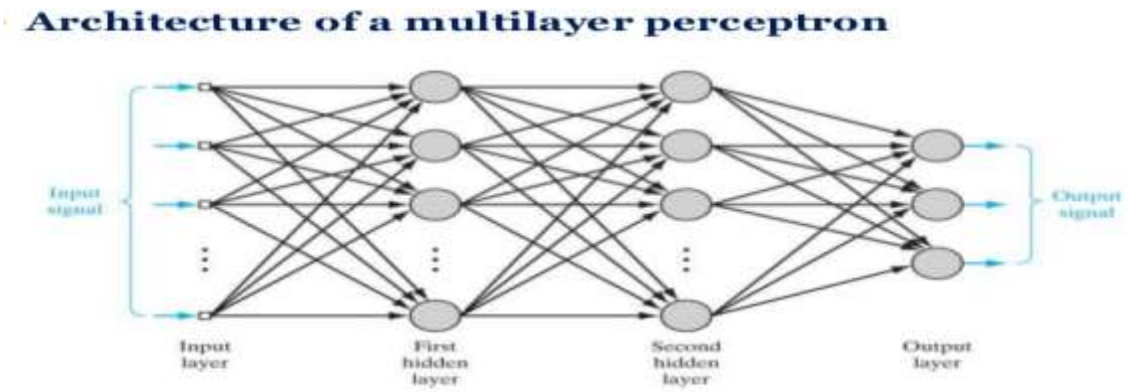

Figure 2.2: Architecture of a multilayer perceptron

2.3. Gated recurrent units:

Gated recurrent units (**GRU**s) are gating machinery in recurrent neural networks launched in 2014 by Kyunghyun Cho et al. Their concert on polyphonic music modelling and speech signal modelling was launch to be similar to that of long short-term memory However, GRUs have been shown to exhibit better performance on smaller datasets.

III.METHODOLOGY

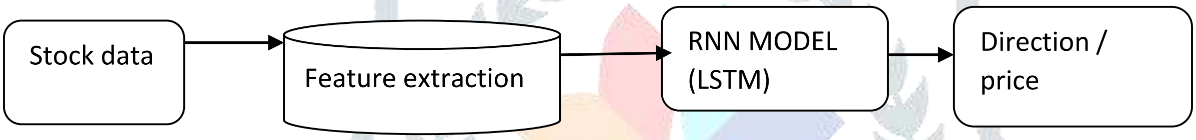

Figure 3: Proposed method

The main aim of our proposed method is to predict the Google Stock Price using LSTM (Long Short-Term Memory) which gives better results when compared to GRU (gated recurrent units) which is practically proved. For implementing the proposed method, first collect the stock data i.e., Google stock data from yahoo finance i.e. from 2012 to 2016 December. After that, feature scaling and feature extraction is done. And then, the proposed method needs to apply RNN model (LSTM) for predicting stock price of January 2017.

3.1. LSTM (LONG SHORT-TERM MEMORY):

LSTM is a type of recurrent network that has demonstrated very triumphant on a number of problems given its potential to discriminate between recent and early examples by giving diverse weights for each while fail to remember memory it thinks irrelevant to predict the next output. In this way, it is more proficient to switch long sequences of input when contrast to other recurrent neural networks that are only able to remember short sequences.

Long Short-Term Memory (LSTM) networks, are a deep and recurrent model of neural networks. Recurrent networks vary from the traditional feed-forward networks that mean they don't only have neural connections on a single direction, in other words, neurons can pass data to a preceding or the same layer. In this case data doesn't flow on a distinct way, and the realistic consequences for that is the survival of short term memory, in addition to long term memory that neural networks already have in significance of training. LSTM are aimed for an enhanced performance by tackling the vanishing gradient issue that recurrent networks would endure when dealing with lengthy data sequences. It keeps the error flow invariable through special units called" gates" which allows for weights adjustments as well as truncation of the gradient when its information is not necessary.

Historic price data from dissimilar stocks from Google Stock Exchange will be used as spring of information for the network. Along with this data, a huge number of technical indicators will also be generated to feed the network as features. Upon this dataset the model will be trained, estimated and will attempt to predict whether the price of a particular stock will go up or not in the next 15 minutes with a graph and also displays the loss rate. LSTM is used to forecast value associations using an input that is not based on text and it is not something that has been widely investigated.

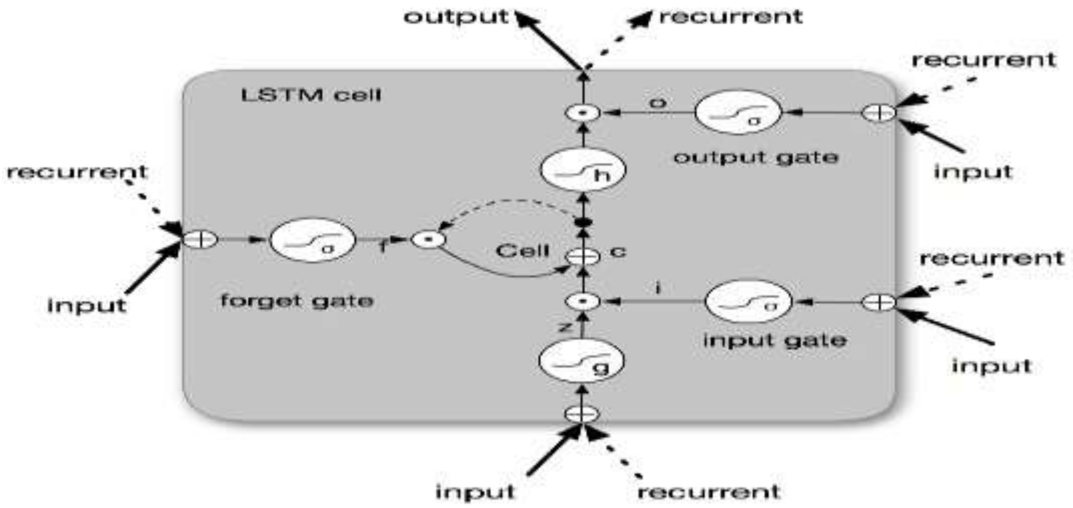

Figure 3.1: LSTM network

LSTM uses a wide range of technical indicators to do so, and the objective is to appraise the usage of such method that is something commonly used on speculation approaches. Additionally, test the hypothesis that the short-term memory capability can present better results compared to traditional feed forward networks.

The important tasks in this particular project are:

- A new worth faction forecast model for stock markets using deep learning-based technique.
- The validation of the model using real data from Google stock exchange.
- Assessment of the model by evaluating and analyzing it against with Gated Recurrent Neural Networks.

$$
\begin{aligned}
\mathbf{z}^t &= g(\mathbf{W}_z\mathbf{x}^t + \mathbf{R}_z\mathbf{y}^{t-1} + \mathbf{b}_z) && \textit{block input}\\
\mathbf{i}^t &= \sigma(\mathbf{W}_i\mathbf{x}^t + \mathbf{R}_i\mathbf{y}^{t-1} + \mathbf{p}_i \odot \mathbf{c}^{t-1} + \mathbf{b}_i) && \textit{input gate}\\
\mathbf{f}^t &= \sigma(\mathbf{W}_f\mathbf{x}^t + \mathbf{R}_f\mathbf{y}^{t-1} + \mathbf{p}_f \odot \mathbf{c}^{t-1} + \mathbf{b}_f) && \textit{forget gate}\\
\mathbf{c}^t &= \mathbf{i}^t \odot \mathbf{z}^t + \mathbf{f}^t \odot \mathbf{c}^{t-1} && \textit{cell state}\\
\mathbf{o}^t &= \sigma(\mathbf{W}_o\mathbf{x}^t + \mathbf{R}_o\mathbf{y}^{t-1} + \mathbf{p}_o \odot \mathbf{c}^{t} + \mathbf{b}_o) && \textit{output gate}\\
\mathbf{y}^t &= \mathbf{o}^t \odot h(\mathbf{c}^t) && \textit{block output}
\end{aligned}
$$

Figure 3.2: LSTM formulas

IV.IMPLEMENTATION

4.1. Data Setup

The data is collected from Yahoo Finance. The data is .csv file. LSTM is most powerful model and performs better than traditional feed forward neural networks. It is super robust with some high dimensionality with several layers. LSTM adds dropout regularization to avoid over fitting by using Keras libraries. To train the LSTM by using 5 years of data beginning of 2012 to ending of 2016. Based on the correlations identified or captured by the LSTM of the Google Stock Price, our proposed method is trying to predict the first month of 2017 i.e., January. For pre-processing the data, it needs some python packages like numpy, matplotlib and pandas.
The attributes for the given data set is DATE, CLOSE, OPEN, HIGH, LOW, VOLUME.

Table 4.1: Description of attributes

ATTRIBUTE NAME	**DESCRIPTION**
DATE	Time (which minute of the day)
CLOSE	Closing price (price at the end of the minute)
HIGH	High price (maximum price during the minute)
LOW	Low price (Minimum price during the minute)
OPEN	Opening price (price at the beginning of the minute)

VOLUME	How many contracts were offered to be bought/sold in the minute)

4.2. Training Process:

This prototype was designed to effort on a systematic frame manner. A new neural network was produced at the end on each trading day that means a new set of weights is defined using a new set of training and corroboration data. For training it is used the data which consists of trading prior to the current day, and the model performance is validated by using the data of the past week. On the following day, all the predictions will be done using the most recent model. The proposed method is working with a time series, the supervised learning algorithm that was preferred was the LSTM neural network (Long short-term memory), which is the current neural network proficient of categorizing input data taking into account the previous instances. Google's Tensor Flow was used to build the model, which consists of a LSTM input layer, that will take both technical indicators and pricing data as input and will feed an output layer using sigmoid activation

S(t)=1 /1+e−t

The input layers have a dimensionality of 180 features, that consists of the set of technical indicators and the price return data (open, close, minimum, maximum) and volume. It will have an output using the tanh function and that will be connected to the network's output layer through 20 connections. LSTM will also check the error rate of MAPE (mean absolute percentage error), RMSE (root mean square error), and MAE (mean absolute error) and compare the results with GRU.LSTM gives best results when compared to GRU with all error rates. LSTM improves the accuracy and gives perfect results for large datasets. LSTM is most powerful model and performs better than traditional feed forward neural networks. It is super robust with some high dimensionality with several layers. It also adds dropout regularization to avoid over fitting by using keras library. For implementing LSTM, the proposed method uses PYTHON language. In this SPYDER is used to build the code because the execution time is fast in this SPYDER. The proposed method compares all the errors i.e., MAPE, MAE, RMSE with GRU practically, and gives best result. To train the data, LSTM uses 5years of data beginning of 2012 to ending of 2016. Based on correlations identified or captured by the LSTM of google stock price, the proposed method is going to predict the first month of January 2017.For processing the data, the packages that are needed are numpy, matplotlib, pandas, keras and tensor flow. In these methods, building RNN is a regressor, Indeed the method is dealing with regressor because, it is trying to predict a continuous outcome (Google stock price). For regression, the way to evaluate the model performance is with a metric called RMSE (root mean squared error). It is calculated as the root of the mean of the squared differences between the predictions and the real values. However, for our specific stock price prediction problem, evaluating the model with the RMSE doesn't make much sense, since the method is not interested in the directions taken by the predictions, rather than the closeness of their values to the real stock price. The method wants to check if the predictions follow the same directions as the real stock price and also check whether our predictions are close to the real stock or not. The predictions could indeed be close but often taking the opposite direction from the real stock price.

4.3 Results

4.3.1. Performances

The prediction performances are measured in terms of 3 metrics to compare the results of the 3 methods. Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are two scale-dependent measures while Mean Absolute Percentage Error (MAPE) is a scale-independent measure. The three metrics are defined as follows:

4.3.1.1. Root Mean Square Error (RMSE): Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a gauge of how far from the regression line data points are; RMSE is a gauge of how spread out these residuals are. In other words, it tells you how deliberated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Figure 4.3.1.1:

4.3.1.2. Mean Absolute Error (MAE): MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Figure 4.3.1.2:

If the absolute value is not taken (the signs of the errors are not removed), the average error becomes the Mean Bias Error (MBE) and is usually intended to measure average model bias. MBE can convey useful information but should be interpreted cautiously because positive and negative errors will cancel out.

4.3.1.3. Mean Absolute Percentage Error (MAPE): The MAPE (Mean Absolute Percent Error) measures the size of the error percentage terms. It is calculated as the average of the unsigned percentage error, as shown in the example:

$$\left(\frac{1}{n}\sum\frac{|Actual - Forecast|}{|Actual|}\right)*100$$

Month	Actual	Forecast	Absolute Percent Error
1	112.3	124.7	11.0%
2	108.4	103.7	4.3%
3	148.9	116.6	21.7%
4	117.4	78.5	33.1%
MAPE			17.6%

Figure 4.3.1.3:

Many organizations focus primarily on the MAPE when assessing forecast accuracy. Most people are comfortable thinking in percentage terms, making the MAPE easy to interpret. The MAPE is scale sensitive and should not be used when working with low-volume data. Notice that because "Actual" is in the denominator of the equation, the MAPE is undefined when Actual demand is zero. Furthermore, when the Actual value is not zero, but quite small, the MAPE will often take on extreme values. This scale sensitivity renders the MAPE close to worthless as an error measure for low-volume data.

Table 4.2: comparison of LSTM and GRU loss rate for the errors i.e., RMSE, MAE, MAPE

S.NO	EPOCHS	ERRORS	LSTM LOSSRATE	GRU LOSSRATE
1	200	RMSE	2.4856e-04	2.6200e-04
2	200	MAE	0.0102	0.0103
3	200	MAPE	4884.8326	7803.3526

In the above table, the proposed method calculates the loss rate by applying different iterations (epochs) i.e., 200, 500, and 1000 for both LSTM and GRU. It clearly shows that LSTM is better than GRU because the LSTM loss rate is very much less than GRU.

4.3.2. Outputs

The results are shown in a graphical representation also.

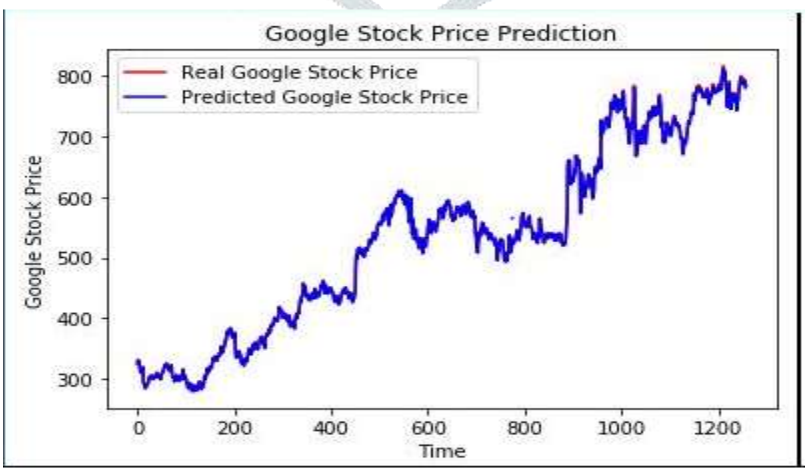

Figure 4.3.2.1: GRU (RMSE)

```
1257/1257 [==============================] - 0s 30us/step - loss: 2.6200e-04
```

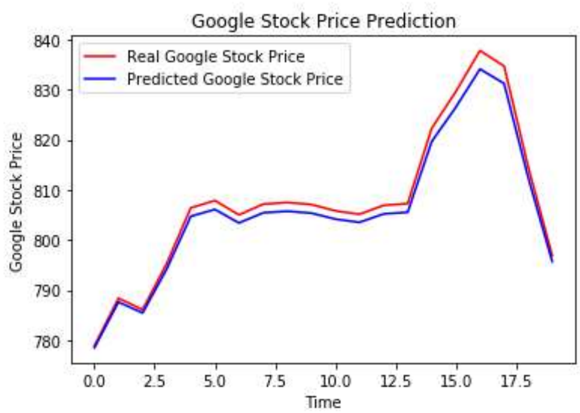

Figure 4.3.2.2: GRU with loss rate (RMSE)

```
1257/1257 [==============================] - 0s 112us/step - loss: 7803.3526
```

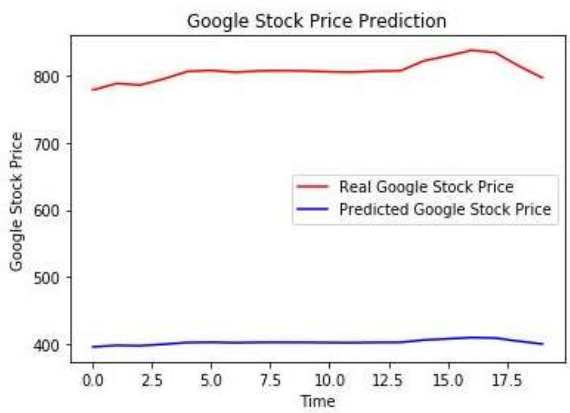

Figure 4.3.2.3: GRU with loss rate (MAPE)

Figure 4.3.2.4: GRU (MAPE)

```
1257/1257 [==============================] - 0s 99us/step - loss: 0.0103
```

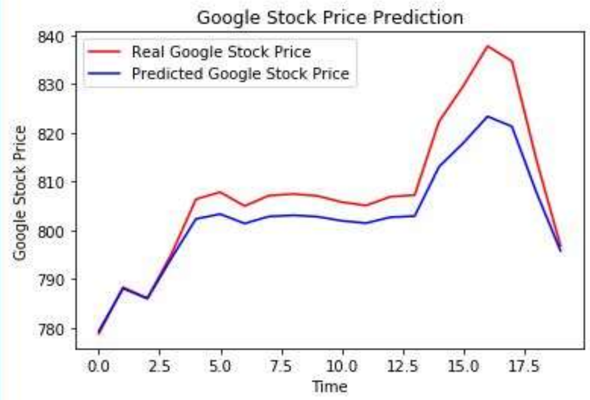

Figure 4.3.2.5: GRU with loss rate (MAE)

Figure 4.3.2.6: GRU (MAE)

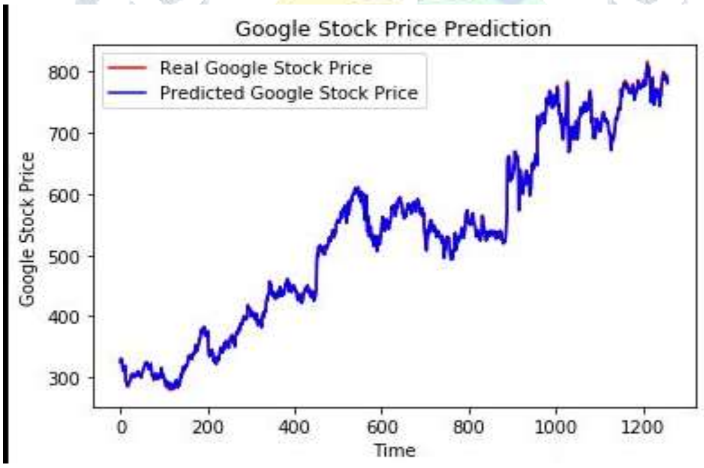

Figure 4.3.2.7: LSTM (RMSE)

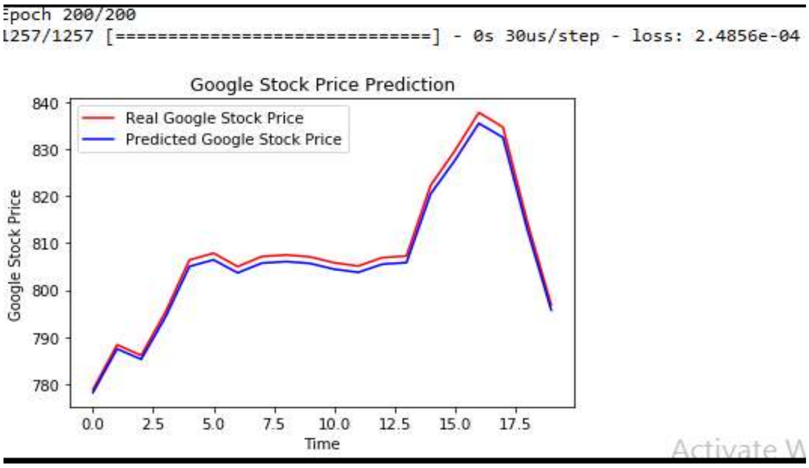

Figure 4.3.2.8: LSTM with loss rate (RMSE)

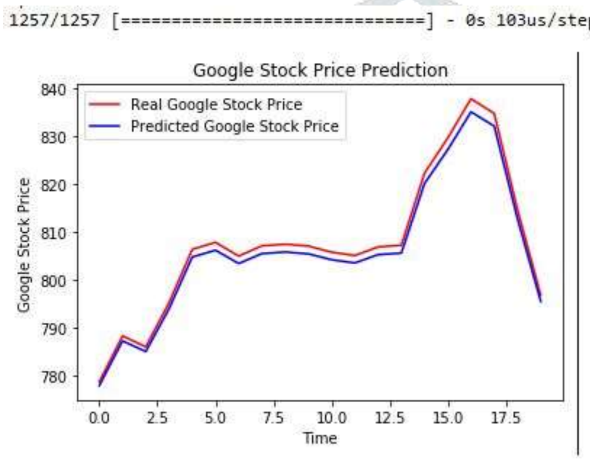

Figure 4.3.2.9: LSTM with loss rate (MAE)

Figure 4.3.2.10: LSTM (MAE)

Figure 4.3.2.11: LSTM (MAPE)

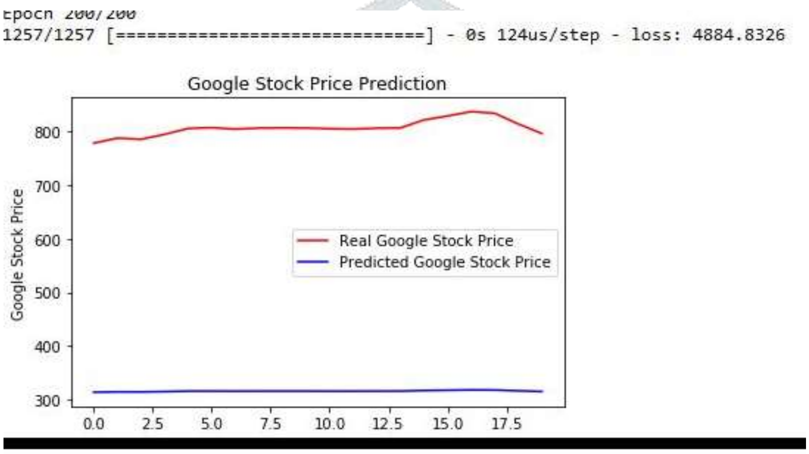

Figure 4.3.2.12: LSTM with loss rate (MAPE)

V.CONCLUSION

Stock market forecasting is gaining more attention day by day. They are lot of methods in deep learning to predict the stock market prices.one such popular method is prediction using deep learning with GRU. The drawback of GRU is that it has high loss rate and GRU's have been shown to exhibit better performance on only smaller datasets. So, our proposed method chooses LSTM which is part of RNN in ANN gives better results when compared to GRU. The performance comparison can be done by 3 types of errors i.e., RMSE, MAE, MAPE. These are all gives better results in an efficient manner and also give better accuracy. LSTM deals with larger datasets where GRU cannot. The proposed method checks all the errors with both LSTM and GRU practically and our results proved that LSTM gives better performance compared to GRU.

VII.REFERENCES

[1] Weiling Chen, Yan Zhang, Chai Kiat Yeo, Chiew Tong Lau, Bu Sung Lee "Stock Market Prediction Using Neural Networks through News on Online Social Networks," International journal, 2017.
[2] https://in.finance.yahoo.com
[3] https://www.google.com/finance
[4] Machine Learning in Stock Price Trend Forecasting. Yuqing Dai, Yuning Zhang
[5] Stock Market Forecasting Using Machine Learning Algorithms. Shunrong Shen, Haomiao Jiang. Department of Electrical Engineering. Stanford University
[6] How can machine learning help stock investment? Xin Guo
[7] Scikit-Learn
[8] Theano
[9] Recurrent Neural Networks - LSTM Models
[10] ARIMA Models
[11] https://github.com/dv-lebedev/google-quote-downloader
[12] http://www.investopedia.com/articles/basics/09/simplified-measuring-interpreting-volatility.asp
[13] https://github.com/dzitkowskik/StockPredictionRNN
[14] https://stackoverflow.com/questions/45271344/importerror-no-module-named-keras

[15] David M. Q. Nelson, Adriano C. M. Pereira, Renato A. de Oliveira "Stock Market Prediction Using Recurrent Neural Networks," International journal, 2017.
[16] O.H. Luca DiPersio, "Artificial neural networks approach to the forecast of stock market price movements," international Journal of Economics and Management Systems, vol. 1, pp. 158–162, 2016.
[17] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in Big Data (Big Data), 2015 IEEE International Conference on, Oct 2015, pp. 2823–2824.
[18] S. Hochreiter and J. Schmid Huber, "Long short-term memory," Neural Compute., vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735
[19] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning in finance," CoRR, vol. abs/1602.06561, 2016. [Online]. Available: http://arxiv.org/abs/1602.06561.
[20] S. G. Chowdhury, S. Routh, and S. Chakrabarti, "News analytics and sentiment analysis to predict stock price trends," International Journal of Computer Science and Information Technologies, vol. 5, no. 3, pp. 3595–3604, 2014
[21] S. Quayes and A. M. Jamal, "Impact of demographic change on stock prices," The Quarterly Review of Economics and Finance, vol. 60, pp. 172–179, 2016.