STOCK PRICE PREDICTION USING LSTM WITH REST API & ANDROID APP IMPLEMENTATION

A PROJECT REPORT

Submitted by

MD. SHARIFUL ISLAM MUBIN (ROLL: 1407034)

&

MUNSI AL MOAZ (ROLL: 1407053)

Under the supervision of

Sk. Imran Hossain

Assistant Professor

Department of Computer Science & Engineering

Khulna University of Engineering & Technology
Khulna-9203

APPROVAL SHEET

This project entitled **“STOCK PRICE PREDICTION USING LSTM WITH REST API & ANDROID APP IMPLEMENTATION”**, Prepared and submitted by **MD. SHARIFUL ISLAM MUBIN & MUNSI AL MOAZ** in partial fulfilment of the requirement for the degree of **Bsc. Engg.** in **Computer Science and Engineering** has been examined and hereby recommended for approval and acceptance.

Supervisors:

____

(Sk. Imran Hossain)

(Asst. Prof.)

Department of Computer Science and Engineering

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

MD. SHARIFUL ISLAM MUBIN	MUNSI AL MOAZ
Date: __________	Date: __________

TABLE OF CONTENTS

ACKNOWLEDGEMENT

It is a matter of great pleasure to present this progress report on " STOCK PRICE PREDICTION USING LSTM WITH REST API & ANDROID APP IMPLEMENTATION ". We are grateful to our project supervisor **Sk. Imran Hossain**, who has been of invaluable assistance to our project with his advice and suggestions. We are very much thankful to our sir for giving us the opportunity to do this project.

Group Members

MD. SHARIFUL ISLAM MUBIN

MUNSI AL MOAZ

ABSTRACT

In a financially volatile market, as the stock market, it is important to have a very precise prediction of a future trend. Because of the financial crisis and scoring profits, it is mandatory to have a secure prediction of the values of the stocks. Stock price forecasting contains uncovering the market trends, planning investment tactics, identifying the best time to purchase the stocks and which stocks to purchase

Predicting a non-linear signal requires advanced algorithms of machine learning. One of the trendy ways of forecasting is time series analysis.The literature contains studies with machine learning algorithm Long Short Time Memory (LSTM) with the help of Adam optimization. We have made our own API for collecting DSE (DHAKA STOCK EXCHANGE) stock price. We also developed an android app for end users. The app will also show graphical view of our predicted data.

List Of Figures

INTRODUCTION

Background:

Today we live and breathe data. Forecasting the stock exchange data is an important financial subject which involves an assumption that the fundamental information publicly available in the past has some predictive relationships to the future stock returns. A stock exchange or equity business sector is a non-direct, non-parametric framework that is difficult to model with any sensible exactness. It is the mix of speculators who need to purchase or offer or hold a share at a specific time. Prediction will continue to be an exciting locale of research, making scientists in the analytics field always desiring to enhance the existing forecasting models. The motivation is that companies and individuals are empowered to make investment decisions to develop viable systems about their future endeavors.

Previous Work:

Stock price prediction is a heated topic in prediction study of the financial area. The stock market is essentially a non-linear, nonparametric system that is extremely hard to model with any reasonable accuracy. Investors have been trying to find a way to predict stock prices and to find the right stocks and the right timing to buy or sell. Most of the techniques used in technical analysis are highly subjective in nature and have been shown not to be statistically valid.

Recently, data mining techniques and artificial intelligence techniques like decision trees, rough set approach, and artificial neural networks have been applied to this area. Data mining refers to extracting or mining knowledge from large data stores or sets.

Some of its functionalities are the discovery of concept or class descriptions, associations and correlations, classification, prediction, clustering, trend analysis, outlier and deviation analysis, and similarity analysis.

Main Issues in Stock price prediction:

LSTM is the extension of RNN that can store memory for a long time. It is widely used in pattern recognition, forecasting , sentiment analysis etc.

LSTM has 3 gates input, forget and output. Input gate takes the input , the forget gate deletes the memory if not necessary and output gate outputs the prediction. We first predict the future closing price of different companies with the help of LSTM. This prediction will be done on historical data & the future prediction will be done.

In different time spans, we calculate the growth of those companies. Then by analyzing the deviations of closing price for each time span, we took the resulting time span which has maximum growth, i.e. less error for the particular sector.

Objectives:

The main objective of this project is to use machine learning to get a stock price prediction tool to obtain more accurate stock prediction price and to evaluate them with some performance measures. This project can be used to develop android smartphone apps for share market investors.

After getting the idea about stock market forecasting techniques we can understand that by using LSTM methods we will get more accurate results. Then we will be able to reduce the amount of error by which investors can invest their valuable money in the stock market at the right time.

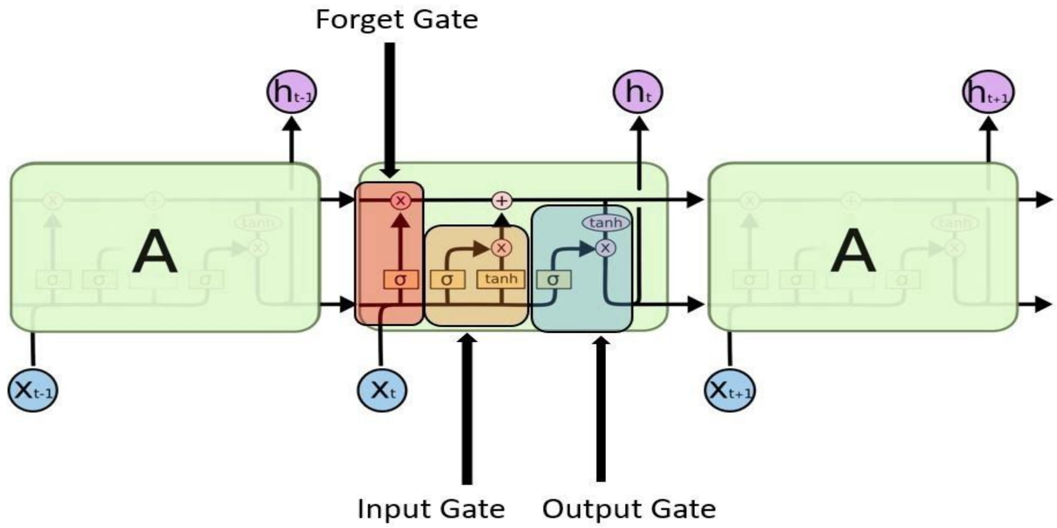
Finally we will serve our findings to the customer level with the help of android application.As we know smartphones are very common nowadays and most of them run android as their operating system,thus our customers can easily have our android app and use for stock trading.

PREDICTION METHODOLOGY

In this project we have worked with LSTM for stock price prediction methods. By which any investor can invest their money in the right time. They can sell, hold or buy stocks at the perfect time by using this methodology.

LONG SHORT TERM MEMORY (LSTM):

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. In an LSTM network, three gates are present:



Fig(1): different gates and inputs of LSTM

1. Input gate — discover which value from input should be used to modify the memory. **Sigmoid** function decides which values to let through **0,1.** and **tanh** function gives weightage to the values which are passed deciding their level of importance ranging from **-1** to **1.**

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

2. Forget gate — discover what details to be discarded from the block. It is decided by the **sigmoid function.** it looks at the previous state(**ht-1**) and the content input(**Xt**) and outputs a number between **0(***omit this*)and **1(***keep this***)**for each number in the cell state **Ct−1**.

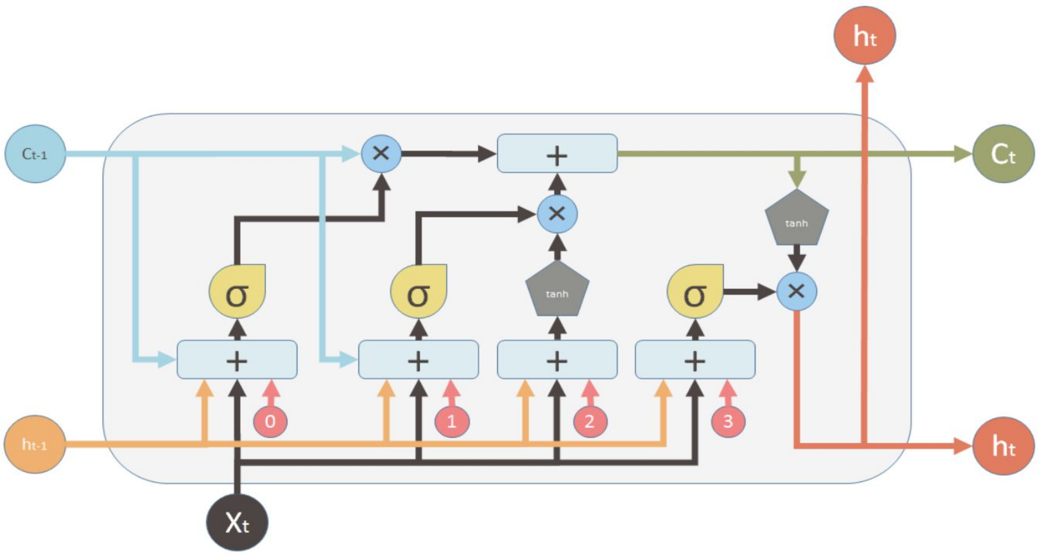
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

3. Output gate — the input and the memory of the block is used to decide the output. **Sigmoid** function decides which values to let through **0,1.** and **tanh** function gives weightage to the values which are passed deciding their level of importance ranging from**-1** to **1** and multiplied with output of **Sigmoid.**

$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

The Working Procedure of LSTM:



Fig(2): workflow diagram of LSTM

The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.The cell state is like a conveyor belt. This runs straight down the entire chain, having some minor linear interactions. LSTM has the ability to add or remove information to the cell state, controlled by structures called gates. Gates are used for optionally letting information through. Gates are composed of a sigmoid neural net layer and a pointwise multiplication operation. The sigmoid layer outputs numbers between 0 and 1, describing how much of each component should be let through. A value of 0 means “let nothing through,” while a value of

1 means “let everything through!" An LSTM has three of these gates, to protect and control the cell state.

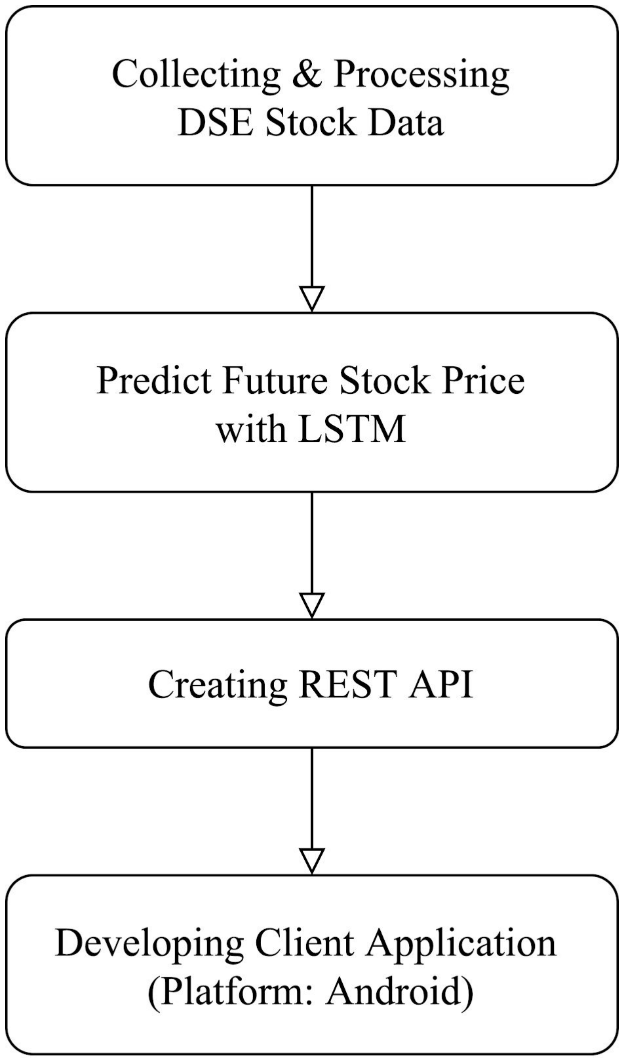
The first step of LSTM is to decide what information is to be thrown out from the cell state. It is made by a sigmoid layer called the “forget gate layer.” It looks at **ht−1**and **xt**, and outputs a number between 0 and 1 for each number in the cell state **Ct−1**. A 1 represents “completely keep this” while a 0 represents “completely remove this.”

In the next step it is decided what new information is going to be stored in the cell state. It has two parts. First, a sigmoid layer called the “input gate layer” decides which values are to be updated. Thereafter, a tanh layer creates a vector of new candidate values, **C~t**, that could be added to the state. In the next step, these two are combined to create an update to the state. It is now time to update the old cell state, **Ct−1**, into the new cell state **Ct**. We multiply the old state by **ft**. Then we add it **C~t**. This is the new candidate values, scaled by how much we decide to update each state value.

Finally, we need to decide on the output. The output will be a filtered version of the cell state. First, we run a sigmoid layer which decides what parts of the cell state we’re going to output. Then, we put the cell state through tanh-tanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

IMPLEMENTATION

We can break down implementation in 4 parts:



Fig(3): Phases of Implementation

Collecting & Processing DSE Stock Data:

We collect DSE stock data from DSE official website and process that data in python.

Sample stock data URL format:

https://www.dsebd.org/php_graph/monthly_graph.php?inst=**TRADE_CODE**&duration=**DURATION_IN_MONTHS**&type=price

Here,

TRADE_CODE data type is “string”

DURATION_IN_MONTHS data type is “integer”

We call this url with the proper trade code of a company and get the data for our desired number of months. After that, we process the response data in python with some string manipulation techniques.

```
<script type="text/javascript">
  g = new Dygraph(

    // containing div
    document.getElementById("graphdiv2"),

    // CSV or path to a CSV file.
    "Date,Price\n" +
    "2019-11-28,318.2\n"+"2019-12-01,311.4\n"+"2019-12-
02,311.2\n"+"2019-12-03,307.3\n"+"2019-12-04,307\n"+"2019-12-
05,307.4\n"+"2019-12-08,305.9\n"+"2019-12-09,302.6\n"+"2019-12-
10,299.1\n"+"2019-12-11,299\n"+"2019-12-12,296.9\n"+"2019-12-
15,300\n"+"2019-12-17,291.1\n"+"2019-12-18,290\n"+"2019-12-
19,293.3\n"+"2019-12-22,284.2\n"+"2019-12-23,288.8\n"+"2019-12-
24,287.2\n"+"2019-12-26,286.9\n"+"2019-12-29,285.8\n"+"2019-12-
30,285.8\n"+"2020-01-01,281.4\n"+"2020-01-02,282.9\n"+"2020-01-
05,282.4\n"+"2020-01-06,275\n"+"2020-01-07,272\n"+"2020-01-
08,259.7\n"+"2020-01-09,250.8\n"+"2020-01-12,253\n"+"2020-01-
13,241.2\n"+"2020-01-14,234.2\n"+"2020-01-15,231.4\n"+"2020-01-
16,242.2\n"+"2020-01-19,263.3\n"+"2020-01-20,278.6\n"+"2020-01-
21,272.8\n"+"2020-01-22,269.3\n"+"2020-01-23,272\n"+"2020-01-
```

Fig(4): Stock data before processing (Trade Code: **GP**)

	Price
Date	
2018-02-27	474.7
2018-02-28	478.7
2018-03-01	483.1
2018-03-04	484.6
2018-03-05	481.7
...	...
2020-02-20	295.6
2020-02-23	319.6

Fig(5): Stock data after processing (Trade code: **GP**)

Predict Future Stock Price with LSTM:

Stock data is a time series data. LSTM is good for working with time series data. Because of this reason, LSTM gives a better accuracy in stock price prediction.

After collecting and processing a company stock data, we collected previous several years data of a company. Example: we took the previous 24 months data of **GP.**

Then, we took 80% of the data as training data and rest 20% of the data as testing data.

We split the 24 months dataset into several small chunks of data. Example: 45 days data in each chunk of the dataset. These chunks of data go as a input in LSTM.

First chunk of data will be 1-45 days of data. This will predict the 46th day price.

Second chunk of data will be 2-46 days of data. Which will predict the 47th day price.

Third chunk of data will be 3-48 days of data. Which will predict the 48th day price.

…

Nth chunk of data will be `n` to `45+n` days of data. Which will predict `45+n`th day price.

We create a sequential model and feed our data in LSTM layer. Our sequential model has 2 LSTM hidden layers and 2 dense layers.

Input size of hidden layer may differ from various company data.

We also notice that, some company stock prediction works well with previous 24 months data, some works well with previous 12 months data. So, it also varies from company to company. Because of these reasons, we've to manually find out the **MONTH_SIZE, ANALYZE_SIZE, EPOCHS** for every company data.

After analyzing **GP** stock dataset several times, we get this:-

Fig(6): Stock Price Prediction (Trade code: **GP**)

Here, MONTH_SIZE = 24, ANALYZE_SIZE = 45, EPOCHS = 4

This way, we analyze & predict other companies' data also. After analyzing and predicting, we store the model and scaler pickle file in our local storage for the exposing our REST API in public.

Creating REST API:

We use Flask Framework in python for creating our REST API for exposing company stock data & predicting future stock price of a company. We expose company data for a special feature for the client application which is a graphical representation of a company stock data.

URL format of company data API: http://localhost:5000/data/**TRADE_CODE**/**MONTHS**

Example: http://localhost:5000/data/GP/6


```
[
  {
    "date": "2019-08-28",
    "price": 316.8
  },
  {
    "date": "2019-08-29",
    "price": 312.0
  },
  {
    "date": "2019-09-01",
    "price": 310.0
  },
  {
    "date": "2019-09-02",
    "price": 304.1
  },
```


Fig(7): JSON response of stock data (Trade Code: **GP)**

By creating Stock Price Prediction API, we encapsulate the underlying logic of our whole mechanism from client application.

URL format of predicting stock price: http://localhost:5000/predict/**TRADE_CODE**

Example: http://localhost:5000/predict/GP


```
{
  "meta": {
    "algo": "LSTM",
    "code": "GP",
    "epoch": 4,
    "month": 24,
    "size": 45
  },
  "prediction": "301.9647"
}
```


Fig(8): JSON response of predicted stock price (Trade Code: **GP**)

Flask server loads all the .pkl files when running the server first. If we make any changes on .pkl files or update any data in our server, we've to restart our Flask server.

All the routes of our REST API responses in JSON format.

Developing Client Application (Platform: Android):

Android's mobile operating system is based on the Linux kernel and it is a software stack for mobile devices. This operating system is one of the world's best-selling Smartphone platform

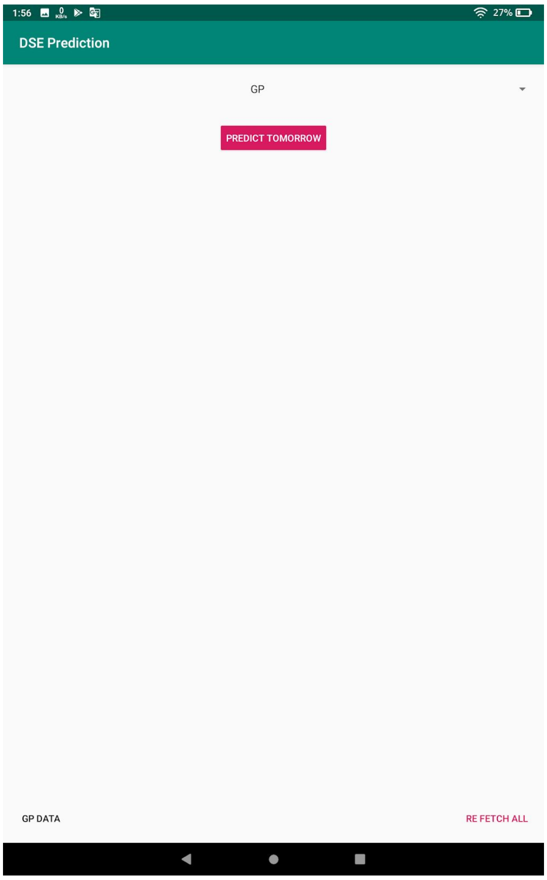
The main supported platform for Android is the ARM (Advanced Risc Machines) architecture.ARM is one of the most licensed and thus widespread processor cores in the world. It is used especially in portable devices due to low power consumption and reasonable performance

We use Android Studio IDE and Kotlin language for our android app development.

RESULT

Our DSE Prediction android application calls the REST API routes to get a company stock data or, predict next day stock data.

Android app processes the API response data and shows it to the client in UI.

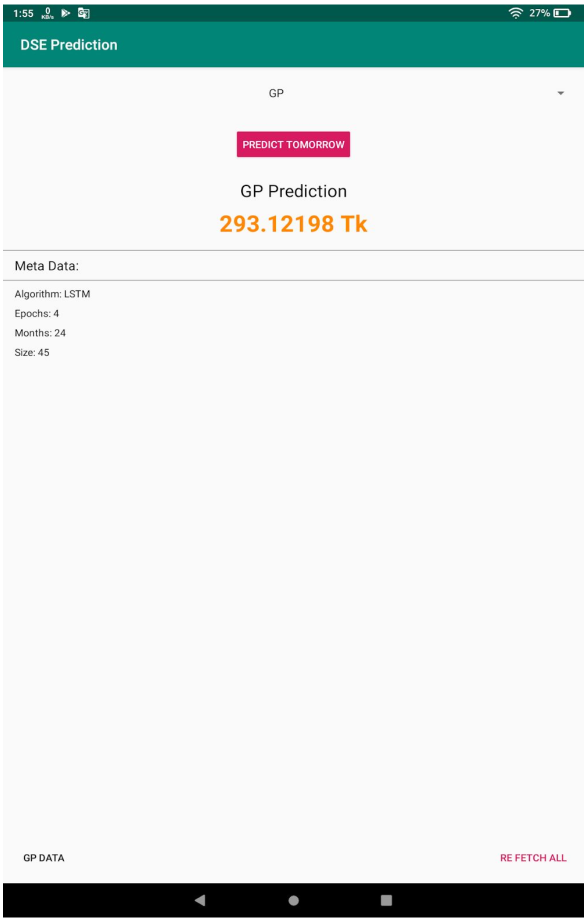


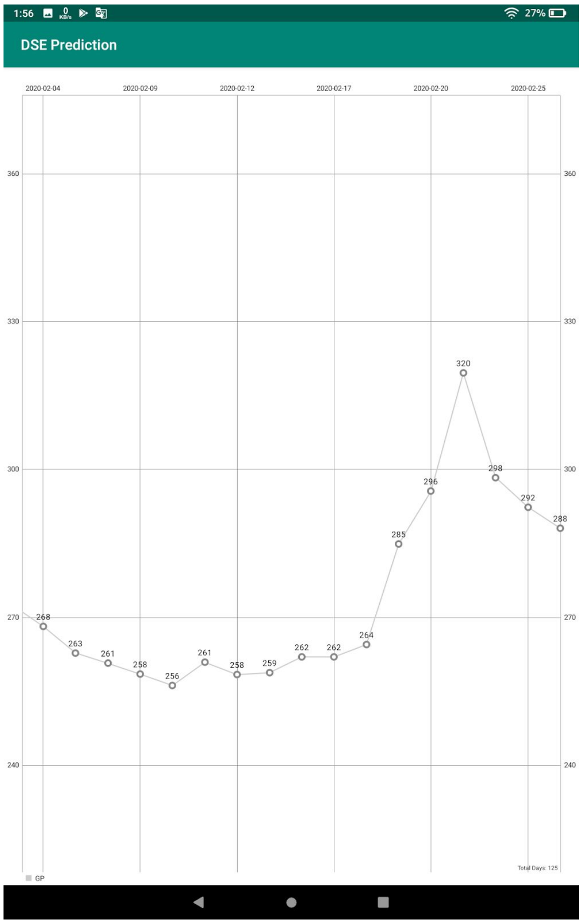
Fig(10): Fetching available stocks Fig(11): Company list dropdown

After fetching all the available stocks from our server, we can choose any of the available stock names from the dropdown menu and after clicking "PREDICT TOMORROW", it will send the request to the server.

Server will process the request and return the JSON response to our Android application. Android application will process the response and show the result to us.

We can also see the graphical representation of **GP** stock data, by clicking on `GP DATA`.





Fig(12): Predicted price of **GP** Fig(13): Stock data graph of **GP**

CONCLUSION

In this project we have implemented a machine learning strategy for Stock Price Prediction which will help the investor for making the correct decision to buy or sell the stocks. We made an attempt to evaluate different methods of forecasting the stock market trends by which any investor can find the best method by which they can predict the stock market much more accurately than previously done methods. Based on the technical analysis using historical time series stock market data data mining techniques.

The experimental results obtained demonstrated in an android app which will assist share investors for making better decisions. This analysis can be used to reduce the error percentage in predicting the future stock prices. It increases the chances for the investors to predict the prices more accurately by reducing the error percentage and hence increase their profit in share markets. Utilizing neural network models together with other forecasting tools and techniques can be considered yet another valuable advancement in the age of technology.

REFERENCES

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is dicult. IEEE Transactions on Neural Networks, 5(2):157-166.

Cleeremans, A., Servan-Schreiber, D., and McClelland, J. L. (1989). Finite-state automata and simple recurrent networks. Neural Computation, 1:372-381.

de Vries, B. and Principe, J. C. (1991). A theory for neural networks with time delays. In Lippmann, R. P., Moody, J. E., and Touretzky, D. S., editors, Advances in Neural Information
Processing Systems 3, pages 162-168. San Mateo, CA: Morgan Kaufmann.

Doya, K. (1992). Bifurcations in the learning of recurrent neural networks. In Proceedings of 1992 IEEE International Symposium on Circuits and Systems, pages 2777-2780.

Doya, K. and Yoshizawa, S. (1989). Adaptive neural oscillator using continuous-time backpropagation learning. Neural Networks, 2:375-385.

Elman, J. L. (1988). Finding structure in time. Technical Report CRL Technical Report 8801, Center for Research in Language, University of California, San Diego.

Fahlman, S. E. (1991). The recurrent cascade-correlation learning algorithm. In Lippmann, R. P.Moody, J. E., and Touretzky, D. S., editors, Advances in Neural Information Processing Systems 3, pages 190-196. San Mateo, CA: Morgan Kaufmann.

Hochreiter, S. and Schmidhuber, J. (1995). Long short-term memory. Technical Report FKI-207-95, Fakultat fur Informatik, Technische Universitat Munchen

Lang, K., Waibel, A., and Hinton, G. E. (1990). A time-delay neural network architecture for isolated word recognition. Neural Networks, 3:23-43.

Miller, C. B. and Giles, C. L. (1993). Experimental comparison of the e
ect of order in recurrent neural networks. International Journal of Pattern Recognition and Artificial Intelligence,7(4):849-872.
Mozer, M. C. (1989). A focused back-propagation algorithm for temporal sequence recognition. Complex Systems, 3:349-381.