Project Title

STOCK MARKET PREDICTION USING DEEP LEARNING

<u>Team Members Details:-</u>

Student Name	Course/Batch	Institute Name
Aadi Bhardwaj	B.Tech in Computer Science and Engineering with Big DATA Analytics	Netaji Subhash University of Technology, Delhi
Arkadeep Roy	B.Tech in Computer Science and Engineering	Heritage Institute of Technology, Kolkata
Deepjyoti Dutta	B.Tech in Computer Science and Engineering	Central Institute of Technology, Kokrajhar
Souvik Das	B.Tech in Computer science and Engineering	Heritage Institute of Technology, Kolkata
Souvik Mukherjee	B.Tech in Computer science and Engineering	Heritage Institute of Technology, Kolkata
Vaibhav Shaw	B.Tech in Computer and Communication Engineering	Amrita Vishwa Vidyapeetham, Coimbatore

Project Guide / Mentor Name: **Udit Das** and **Ganesh Aggarwal**

Period of Internship: 19th May 2025 - 15th July 2025

Report submitted to: IDEAS – Institute of Data Engineering, Analytics and Science Foundation, ISI Kolkata

1. Abstract

This project develops a sophisticated deep learning model for accurate stock market prediction. Leveraging Long Short-Term Memory (LSTM) networks, the model excels at capturing complex temporal dependencies and historical patterns within financial data. A key innovation is its focus on identifying universal market patterns across diverse datasets, enabling robust predictions for any new, unseen company. Our methodology involves rigorous data preprocessing, comprehensive model training, and performance evaluation using metrics like Mean Squared Error and Mean Absolute Error. The LSTM model has demonstrated superior accuracy and efficiency compared to alternative approaches. It is designed to forecast 30-day stock prices, adapting to unique market dynamics through advanced cross-market analysis. An intuitive HTML/CSS/JavaScript front-end provides seamless user interaction for company selection and prediction visualization. The Flask API framework powers the back-end, handling core logic, prediction services, and data management. This system aims to provide a reliable, adaptable, and accessible tool for advanced financial analytics, offering immediate insights into market trends.

2. Introduction

Our project leverages cutting-edge deep learning to tackle the complexities of stock market prediction, offering a robust and adaptive tool for financial insights and informed decision-making.

Relevance

Providing critical foresight for investors and analysts in dynamic financial markets. In today's volatile financial markets, providing critical foresight for investors and analysts is paramount. Our model aims to demystify complex market movements, offering predictive insights that can significantly enhance investment strategies and risk management in highly dynamic environments.

Technology

Our solution is built on a powerful technological stack, primarily utilizing Long Short-Term Memory (LSTM) networks for deep learning, chosen for their superior ability to process and predict time-series data. The backend is orchestrated with Python and the Flask API framework, ensuring scalability and efficient data handling, while a responsive web interface crafted with HTML, CSS, and JavaScript delivers a seamless user experience.

Background

The foundation of this project stems from extensive research into various aspects of time-series analysis and advanced deep learning architectures. This includes a deep dive into historical stock market data, exploring various indicators, and understanding the unique challenges of financial forecasting, leading to the selection and refinement of LSTM models.

Procedure

Data preprocessing, LSTM training, generalization, and full-stack development. Meticulous data preprocessing to ensure data quality and relevance, comprehensive LSTM model training on curated datasets, a focus on generalization to uncover universal market patterns, and full-stack development covering both the robust Flask API backend and the intuitive web-based frontend.

Purpose

The ultimate purpose of this endeavor is to deliver accurate, adaptable stock price forecasts that are not confined to specific companies but can reliably predict for any new, unseen company stock. This aims to create a versatile tool capable of providing actionable insights across the broader market, making advanced financial analytics accessible to a wider audience.

INTERNSHIP TRAINING TOPICS: WEEKS 1 & 2

Our initial two weeks of internship focused on building a strong foundation in Python, data science, and emerging AI technologies, crucial for our stock market prediction project.

WEEK 1: FOUNDATIONS

- Introduction to Python and core Data Science fundamentals, setting the stage for advanced analytics.
- Guidance on Power BI installation and its practical application for data analytics using downloaded datasets.
- In-depth discussion on Python fundamentals, reinforcing programming concepts essential for the project.

WEEK 2: ADVANCED TOPICS & AI

- Comprehensive sessions covering Python 1, Python 2, and Python 3, building progressive programming skills.
- Principles of Survey and Questionnaire Design, vital for data collection and user feedback.
- Exploration of Sentiment Analysis techniques for understanding market sentiment.
- Practical application of Text Analytics using no-code tools for efficient text data processing.
- Introduction to the Foundations of Generative AI and Large Language Models (LLMs), a key area for future advancements.
- Advanced Python 4 session, delving into more complex programming paradigms.
- Understanding GenAI support for Business Intelligence, bridging AI with data-driven decision-making.

3. Project Objective

This project aims to achieve several key objectives, driving towards a more accurate, accessible, and versatile stock market prediction solution:-

➢ *Accurate LSTM Forecasting*

To develop a sophisticated deep learning model, utilizing Long Short-Term Memory (LSTM) networks, capable of precisely forecasting 30-day stock prices by effectively capturing complex temporal dependencies and historical patterns in financial data. This involves meticulous data preprocessing, feature engineering, and hyperparameter tuning to ensure the model learns from the most relevant market signals and minimizes prediction errors over the short to medium term.

➢ *Universal Market Generalization*

To identify and leverage universal market patterns, enabling the model to provide robust and reliable predictions for any new, unseen company, transcending limitations of pre-trained datasets. This objective focuses on developing a highly adaptable framework that can generalize across different industries and market conditions, rather than being confined to specific company historical data. This approach aims to make the model a versatile tool for broader market analysis.

➢ *Empowering Informed Decisions*

To deliver actionable insights into market trends, empowering users with the necessary data for more informed and strategic investment decisions and enhanced risk management. By providing clear, concise, and accurate predictions, the project seeks to demystify complex financial movements, helping both novice and experienced investors navigate volatile markets with greater confidence and a data-driven approach to their portfolio management.

➢ *Intuitive User Experience*

To design and implement a seamless, user-friendly front-end interface that simplifies data selection, prediction generation, and visualization of complex financial forecasts. This includes creating a clean, responsive web application that allows users to easily input company parameters, trigger predictions, and interpret results through interactive charts and accessible tabular data, ensuring that advanced analytics are available to a wider audience.

➢ *Robust & Scalable Backend*

To build a reliable and scalable Flask API framework for the backend, efficiently handling core logic, orchestrating prediction services, and managing diverse company datasets. This objective ensures the system's foundational strength, supporting high-volume requests and secure data processing, while also facilitating easy integration of future model updates and expansion to accommodate a growing range of financial data sources and prediction capabilities.

4. Methodology

There are two parts of our project are:-

- Deep Learning Model (LSTM Model)
- UI feature(Full Stack work)

i. Deep Learning Model-LSTM Model

The project involves collecting historical stock price data, preprocessing it, developing a predictive model(LSTM Model), and evaluating its performance.

Steps Followed:-

1. Data Collection

- We have collected historical stock price data for 5-8 companies for each of the 5 sectors -IT, Automobile, Business, Semiconductor, Telecom using the Yahoo Finance API (yfinance library in Python) separately. The data spans 5 years.
- We have plotted the graph of all the datasets for each sector separately.

2. Data Preprocessing

The following steps were taken to preprocess the data:

- Handling missing values: we have replaced any missing values present in the datasets with the mean of the existing values using np.nan_to_num.
- Data normalization: We used the Min-Max Scaler class from scikit-learn to normalize the data('close' values) between 0 and 1.

3. Splitting the datasets for training and testing

- We have split the datasets separately for each sector such that 80% of each dataset is used for training the model and 20% of each dataset is used for test the model on unseen data.
- We have created two separate lists for each sector to store the training data and the test data.

4. *Model Development*

We developed a predictive model using the LSTM (Long Short-Term Memory) architecture, which is a type of Recurrent Neural Network (RNN) suitable for time series forecasting.

Key Features of LSTM Model:-

- **Memory Cells:** LSTM has memory cells that can store information for long periods.
- **Gates:** LSTM uses gates to control the flow of information, allowing it to learn long-term dependencies.

The model was trained on the historical stock price data, and following steps were taken:-

- Model architecture: The LSTM model consisted of 50 units, followed by a dense layer with 1 unit.
- Model compilation: The model was compiled with the mean squared error (MSE) loss function and the Adam optimizer.

5. *Model Training using train dataset*

- We have created a variable, look_back = 60, which sets the number of previous time steps to use as input for learning the prediction of next time step. So, the model will learn to map last 60 days of input sequences to the output value of next day.
- The model was trained for 50 epochs with a batch size of 32.
- We also have plot the epoch vs Model loss to visualize the loss of model during training.
- The model is trained with 80% of all the datasets for each sector separately.

6. *Model Testing using test dataset*

- We have created a variable window_size = 60. While testing, to predict the stock price of a specific day (on unseen dataset), the model analyses last 60 days price and based on that the predicts the stock price on unseen dataset and the window continues for the whole test dataset.
- The accuracy of the model on predicting the unseen dataset (test dataset) is analyzed by the Mean Absolute Error (MAE) and Mean Square Error (MSE) metrices.
- We have print and plot the MAE and MSE values for all companies of each sector.
- The model is tested with 20% of all the datasets for each sector separately.

7. *Model Evaluation*

We evaluated the model's accuracy using the following metrics:

- Mean Squared Error (MSE): We calculated the MSE for all the companies of each dataset separately using the mean_squared_error function from scikit-learn.
- Mean Absolute Error (MAE): We calculated the MSE for all the companies of each dataset separately using the mean_absolute_error function from scikit-learn.

8. *Next 30 days stock price prediction*

- The trained model predicts the stock price of a specific day based on last 60 days of stock price. The model uses its own predicted price as an input (including last 60 days) to predict the price of a specific day.
- We also inverse transform the data to actual values as we previously reshaped it.
- We have also plotted the days vs price graph for next 30 days with historical price and predicted price for users' visualization and print the stock prices for next 30 days from the current date.

9. *Hypertuning of parameters*

We have hypertuned the parameters like length of train data set, test dataset, window_size, epochs, batch_size , model layer for the better performance of our model.

Tools and Libraries Used:

The project used the following tools and methods:

- **Python:** The programming language used for the project.
- **yfinance**: A Python library, which we used to collect historical stock price datasets from Yahoo Finance.
- **scikit-learn**: A Python library, which we used for data preprocessing (Min-Max Scaler) and model evaluation (MSE and MAE).
- **Keras and Tensorflow**: A Python library, which we used to develop and train the LSTM model.
- **Matplotlib and Tabulate**: These two python libraries were used for visualizing and presenting the results respectively.

Model Selection

We choose LSTM and CNN(Convolutional Neural Network) models for our stock price prediction. The performance and accuracy of these models were analyzed by the metrices **Mean Square Error (MSE)** and **Mean Absolute Error(MAE)**. LSTM model showed less MSE and MAE than those of CNN i.e., its accuracy was better than CNN model. So, we choose LSTM as our final project model.

Model Validation

The validation process involved evaluating the performance of the chosen LSTM model on the test data. The model's performance was evaluated using MSE and MAE, which provided a measure of the model's accuracy and reliability.

By selecting the LSTM model and validating its performance, the project ensured that the final model was well-suited for predicting stock prices and provided reliable results.

Github link for code:- https://github.com/DeepJDUTTA/Stock_prediction.git

ii. UI Feature (Full Stack work)

After building the model, we have saved the trained model data. To deploy this deep learning model we have used flask as a backend server and for frontend we have used HTML, CSS, JavaScript .

For this fullstack development we have used python libraries such as - json, os, numpy, matplotlib, io, base64, tensorflow, keras, load_model, joblib, datetime, tabulate.

There are three files- app.py, predictor.py, model_mapping.json and two folder named models and static. In these folders the trained models were saved as their respected sector_ name.keras and sector_name.pkl. After starting our index.html file in our front-end part when user selects sector name and company name from a drop-down menu, these data will go to the app.py file and after that in our predictor.py file, the model will call the predict_stock_price() function which receives the company ticker name as input parameter and the model will predict the upcoming 30 days stock price and generate the graph of the day vs historical price and predicted price and this graph and predicted data will go to the frontend part and using JavaScript and HTML the user can visualize graph and the predicted stock prices will be shown in a tabular format.

5. Data Analysis and Results

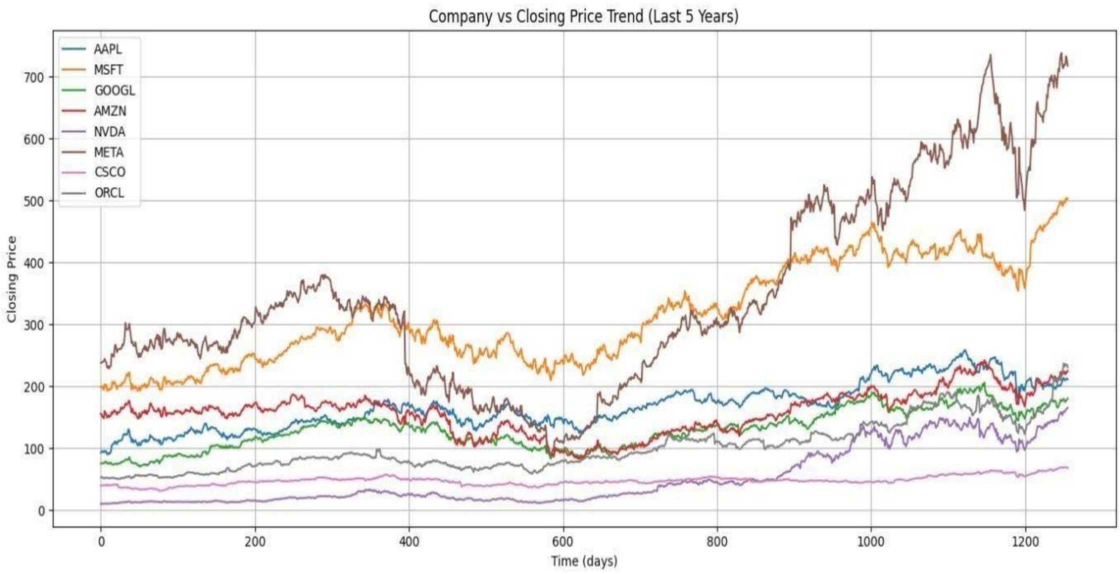
Descriptive Analysis:-

Summary Tables of companies for each sector:

1. IT Sector:-

	Company	Mean	Median	Standard Deviation	Min	Max
0	AAPL	166.870240	164.465904	36.808338	90.028175	258.396667
1	MSFT	315.140957	297.446884	80.798074	192.454880	503.510010
2	GOOGL	129.426281	130.134544	31.203300	70.049393	205.893341
3	AMZN	156.868939	160.310501	34.448833	81.820000	242.059998
4	NVDA	51.138913	26.749267	45.573067	10.096469	164.975006
5	META	345.323894	310.083878	157.201506	88.424889	738.090027
6	CSCO	47.539172	47.047688	7.135510	30.981733	69.370003
7	ORCL	100.563802	86.614548	38.621533	50.242886	236.816803

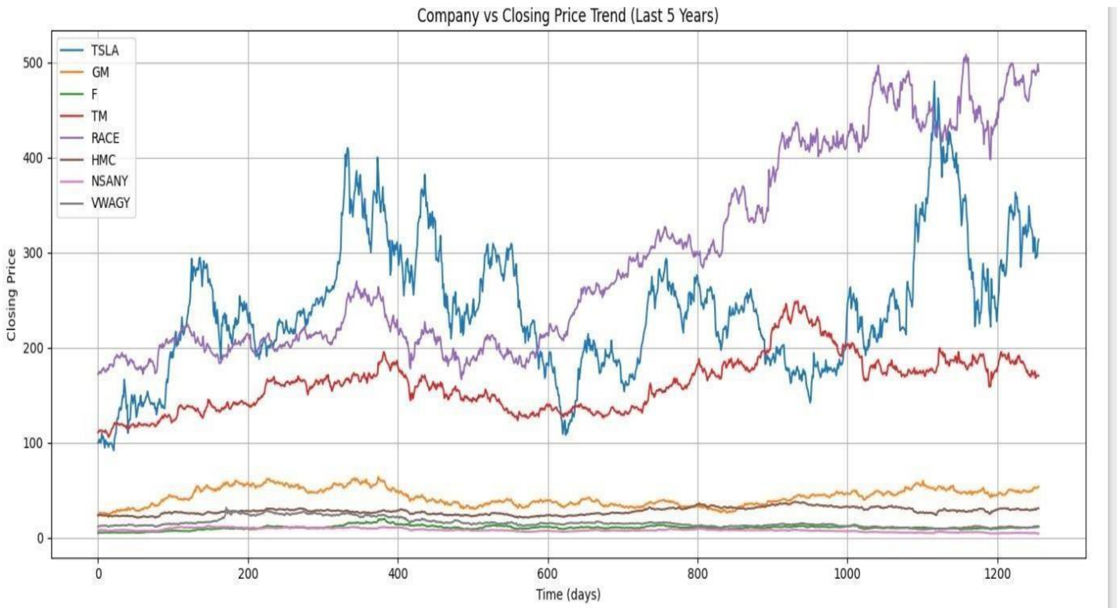
Plot:-



2. Automobile Sector:-

	Company	Mean	Median	Standard Deviation	Min	Max
0	TSLA	241.421496	236.366661	69.000671	91.625999	479.859985
1	GM	42.709905	42.312973	9.137948	24.124842	63.719048
2	F	10.471340	10.431626	2.244439	4.947021	19.692282
3	TM	161.855371	162.613983	28.697074	105.636215	248.562637
4	RACE	294.054283	255.374214	104.769383	166.475159	508.149994
5	HMC	27.882240	27.813386	3.708638	20.872398	37.680000
6	NSANY	7.914729	7.750000	1.838454	4.220000	12.580000
7	VWAGY	15.556712	14.399071	4.846223	8.620000	31.505114

Plot:-



3. Business Sector:-

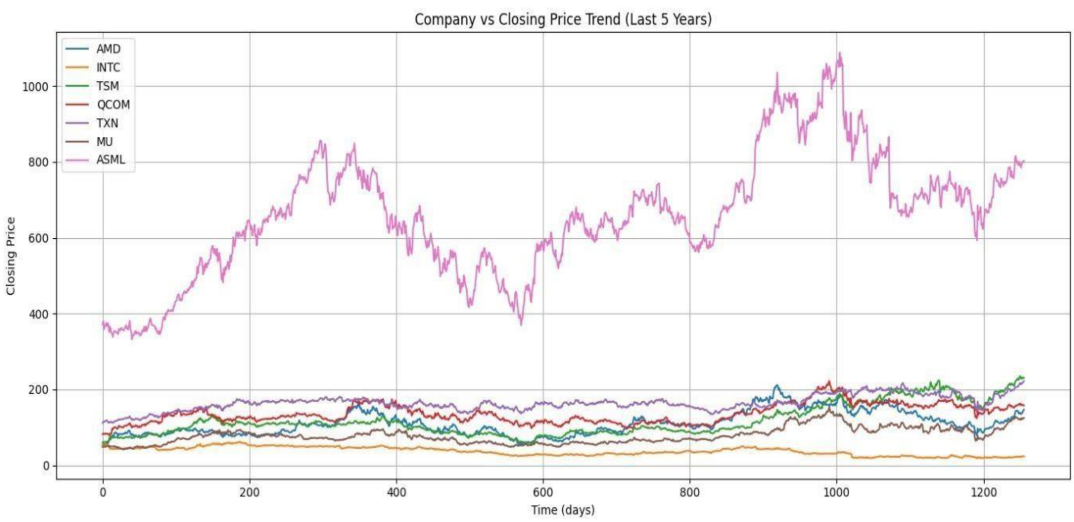
	Company	Mean	Median	Standard Deviation	Min	Max
0	JPM	154.545626	139.376923	49.253708	81.024666	296.000000
1	BAC	34.253123	34.225967	6.699300	20.528545	48.930000
2	WFC	46.046526	42.785873	14.208713	18.963947	83.599998
3	C	53.481385	52.567738	11.583959	34.410297	88.720001
4	GS	363.967833	332.472992	115.415860	166.077835	723.679993
5	MS	85.054409	82.257172	21.686682	39.796291	144.139999
6	AXP	180.528518	161.030403	58.585441	85.831245	328.130005
7	V	240.191004	224.583435	46.798091	173.941284	373.309998
8	MA	395.199094	366.027924	76.407588	278.809387	589.941772
9	PYPL	124.210840	80.649998	77.674947	50.389999	308.529999

Plot:-

4. Semiconductor Sector:-

	Company	Mean	Median	Standard Deviation	Min	Max
0	AMD	110.124132	104.334999	31.421777	54.720001	211.380005
1	INTC	36.829495	35.762850	11.030532	18.129999	62.083328
2	TSM	117.129034	105.614552	41.147458	57.751678	234.800003
3	QCOM	135.928685	131.341331	25.017704	79.903702	222.206665
4	TXN	164.208073	163.526146	19.918749	111.649223	221.250000
5	MU	78.095747	74.891006	21.152268	41.666119	152.602234
6	ASML	657.801613	660.177490	159.281942	332.162445	1088.696533

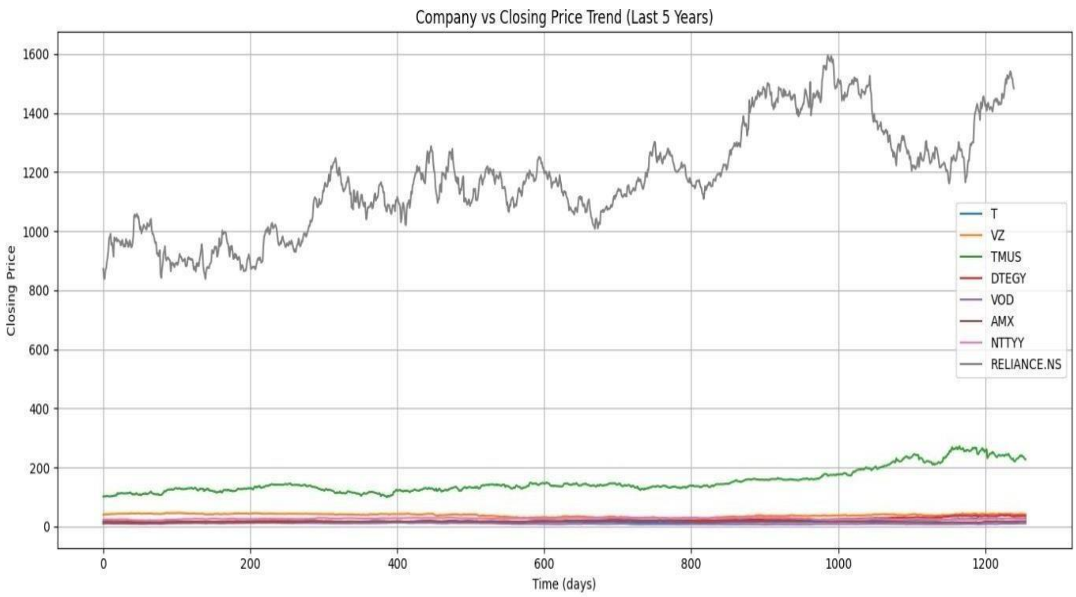
Plot:-



5. Telecom Sector:-

	Company	Mean	Median	Standard Deviation	Min	Max
0	T	17.143191	16.076673	3.645041	12.023441	28.653690
1	VZ	38.601228	40.003222	4.948891	27.364235	46.543446
2	TMUS	152.704594	138.357491	41.451940	99.009804	271.826691
3	DTEGY	21.859552	19.817948	6.092395	13.555486	38.869999
4	VOD	10.084353	9.488894	1.873880	7.271991	14.387831
5	AMX	16.397342	16.862934	2.579068	10.402264	21.682449
6	NTTYY	27.162659	27.235000	2.323015	20.660000	32.430000
7	RELIANCE.NS	1176.846693	1167.578247	179.745558	838.533386	1595.484985

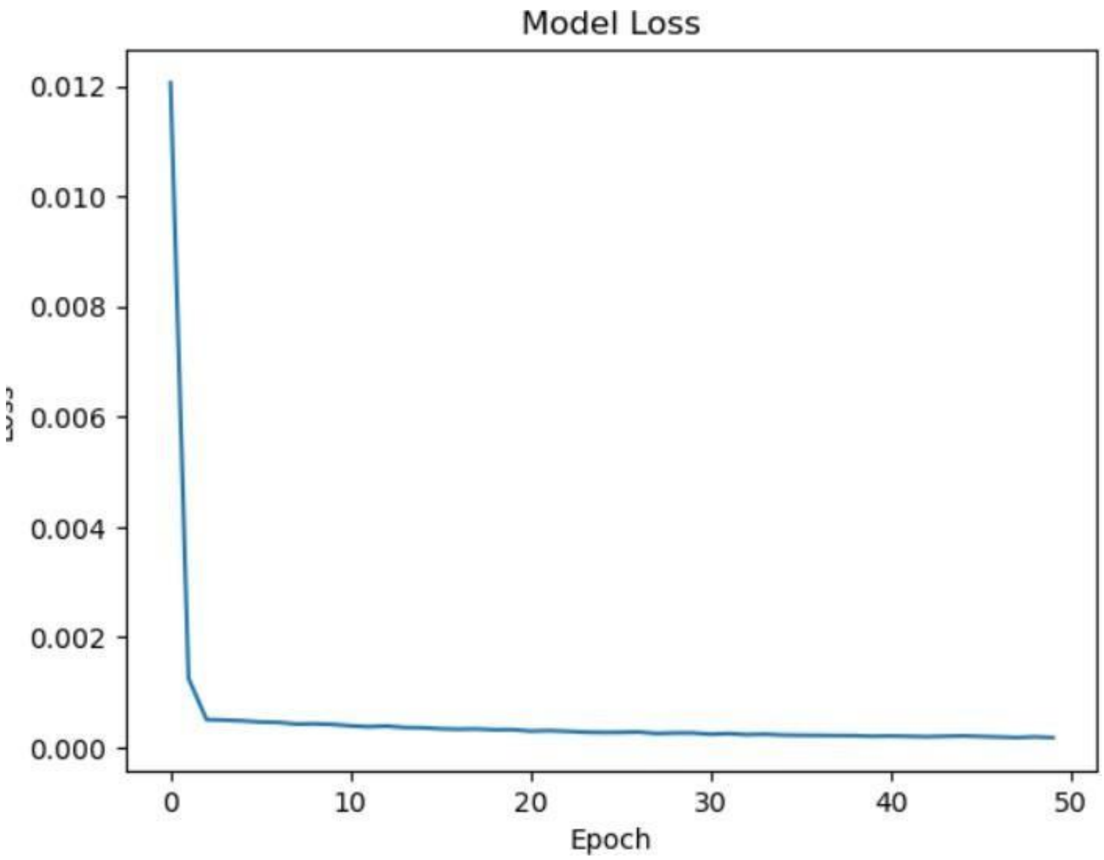
Plot:-



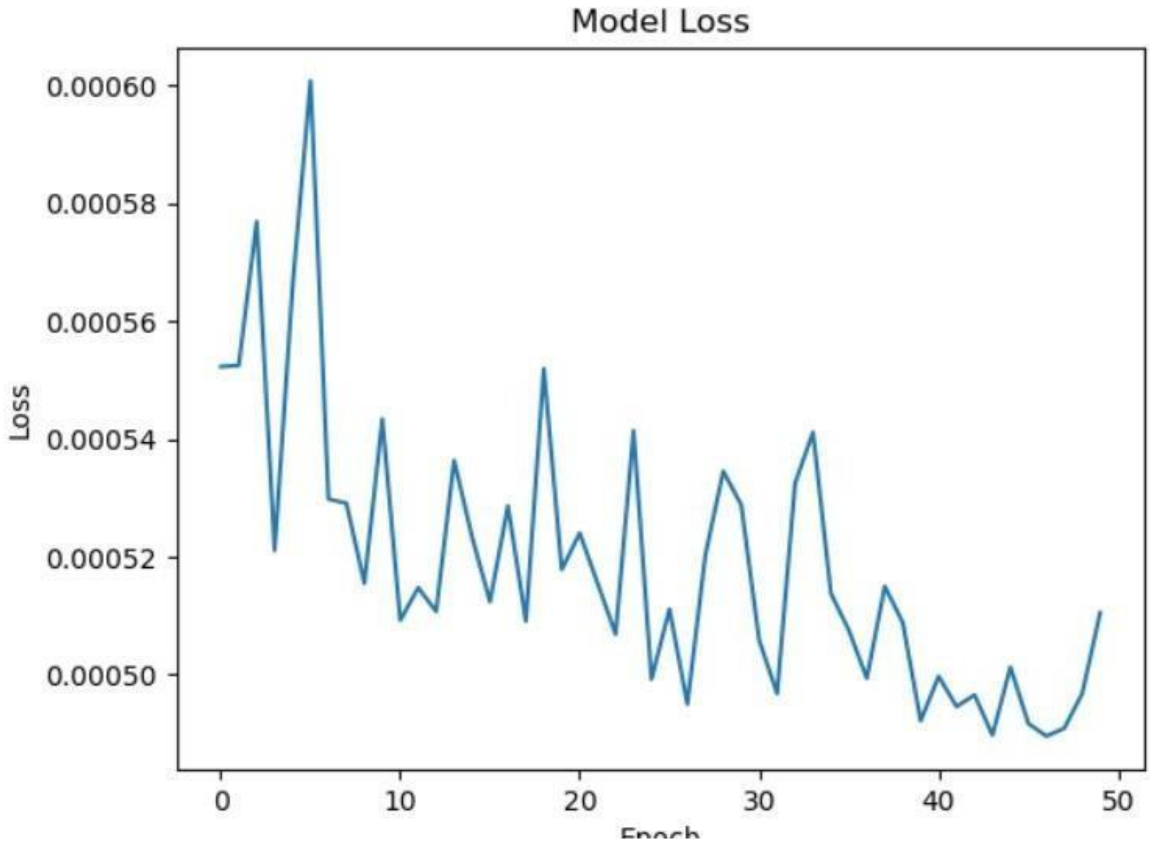
Inferential Analysis(LSTM Model):-

- Epoch vs model loss plots for one company from each sector for LSTM Model:

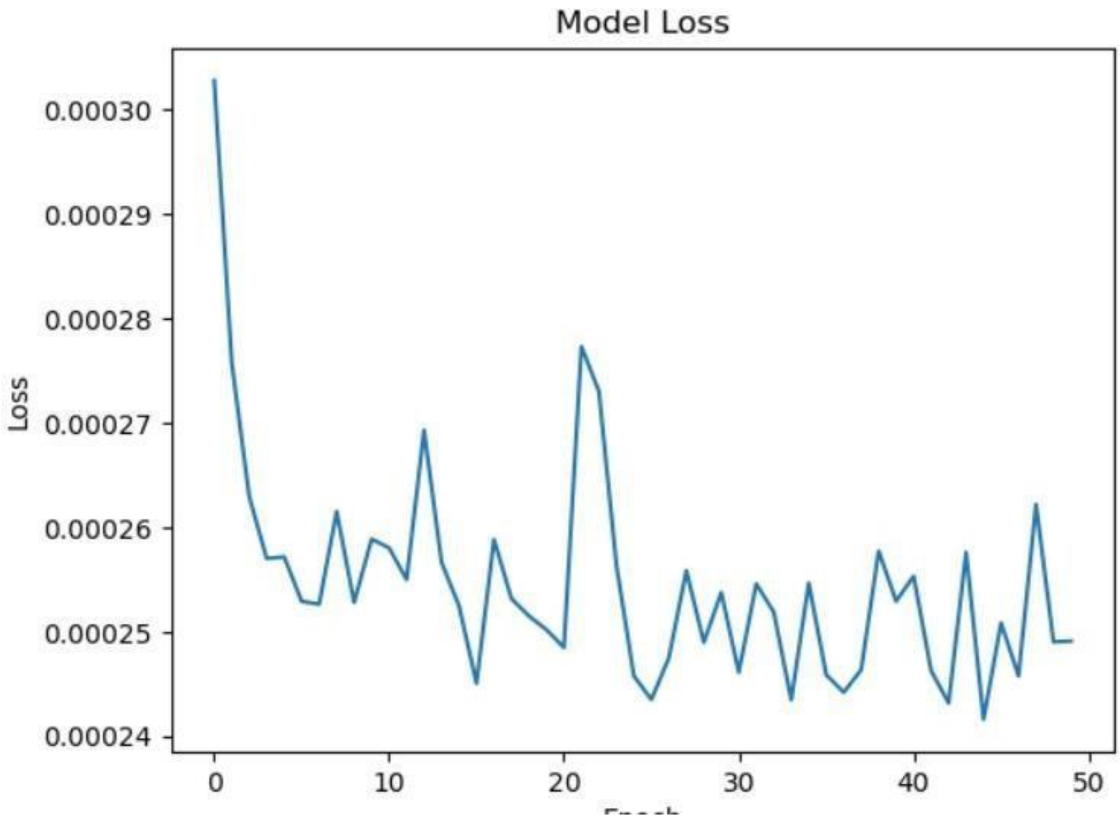
1. APPLE (IT Sector):-



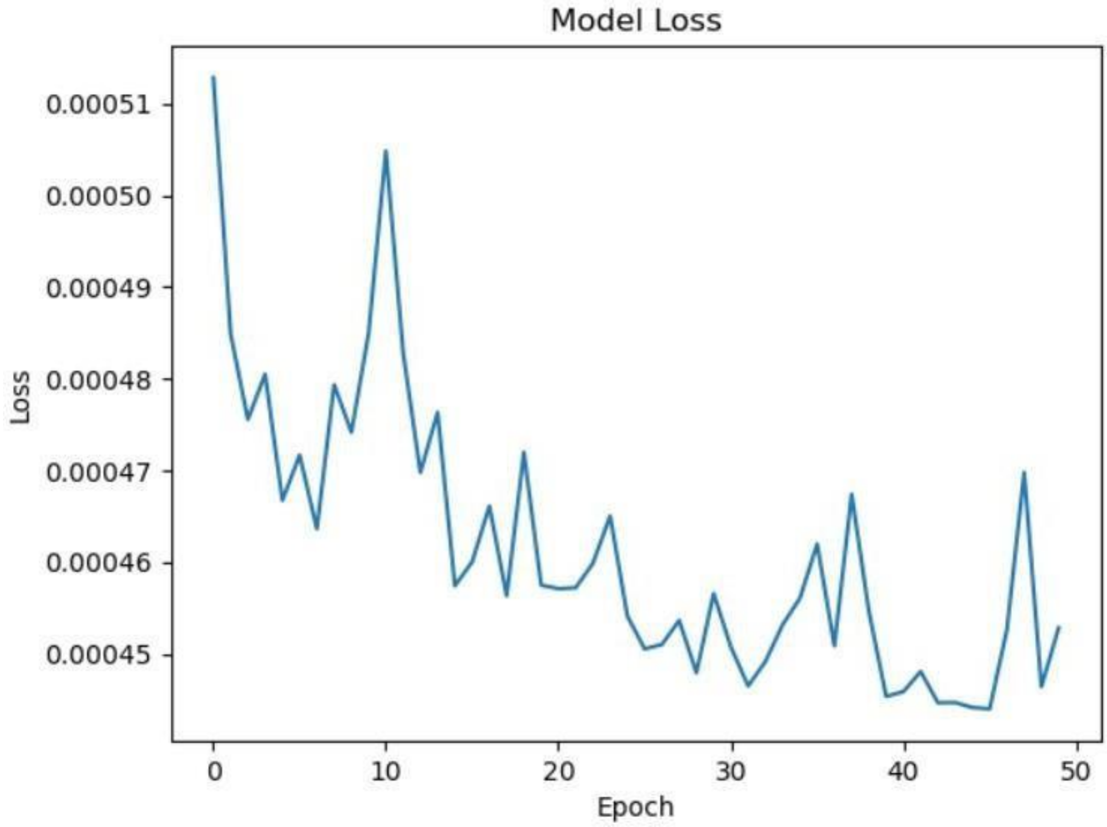
2. *Tesla (Automobile Company):-*



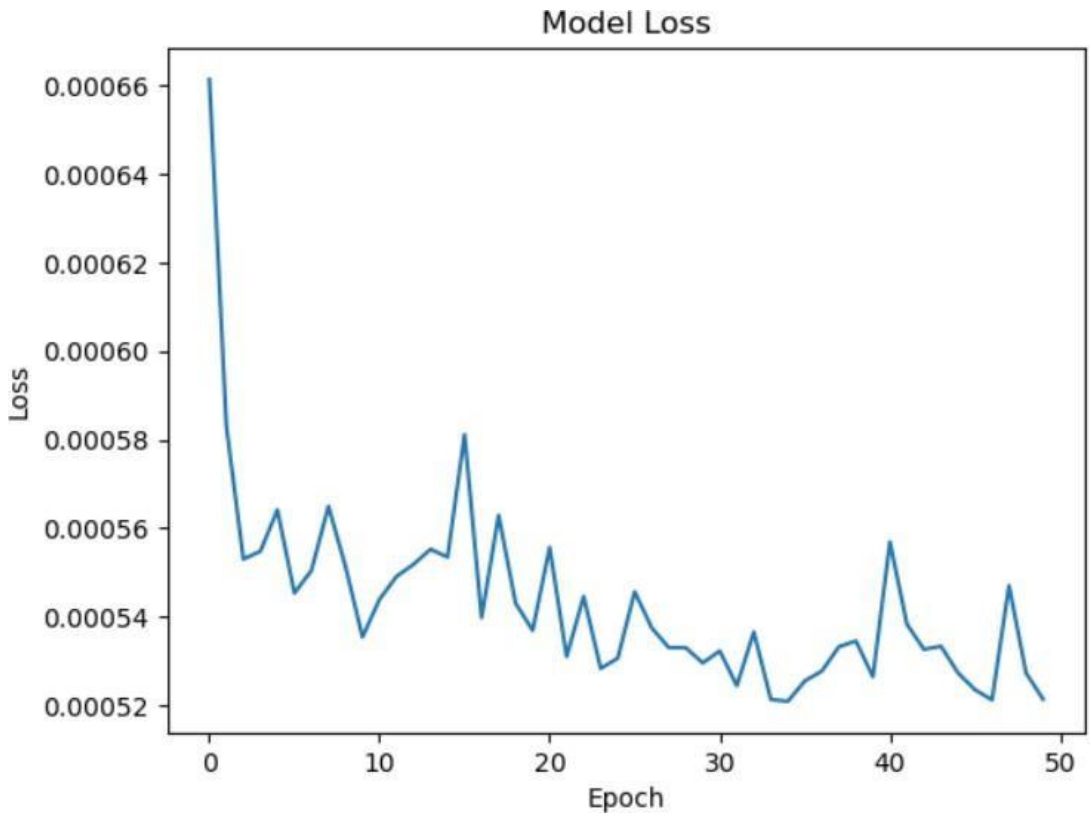
3. *JPMorgan (Business Sector):-*



4. Advanced Micro Devices(Semiconductor Sector):-



5. AT&T(Telecom Sector):-



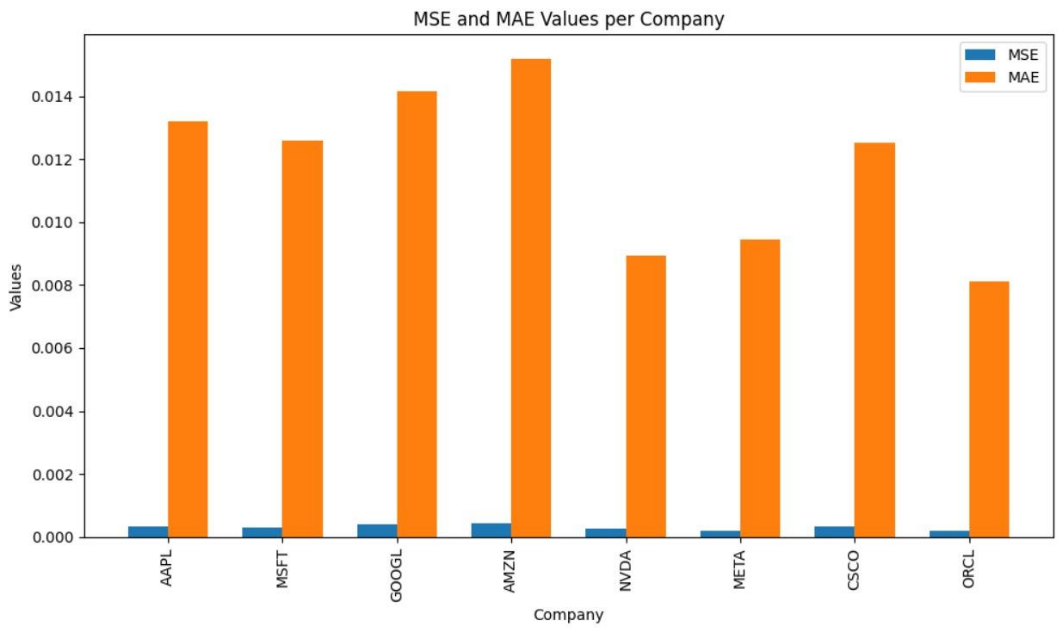
a. Mean Square Error(MSE) and Mean Absolute Error (MAE) analysis of each sector(Accuracy Analysis) for LSTM Model:

We have chosen Mean Absolute Error (MAE) and Mean Square Error(MSE) metrices for the inferential analysis of our LSTM model. The model is tested by predicting next day's stock price based on past 60 days stock values on test datasets of each sector (20 % of whole datasets for each sector) and its MAE and MSE values are calculated for accuracy check.

1. IT Sector:-

Company	MSE	MAE
AAPL	0.000339918	0.0132121
MSFT	0.000287231	0.0125814
GOOGL	0.000384844	0.0141597
AMZN	0.000444864	0.0151948
NVDA	0.000258034	0.00893412
META	0.000212224	0.00944656
CSCO	0.00032987	0.0125025
ORCL	0.000201036	0.0081299

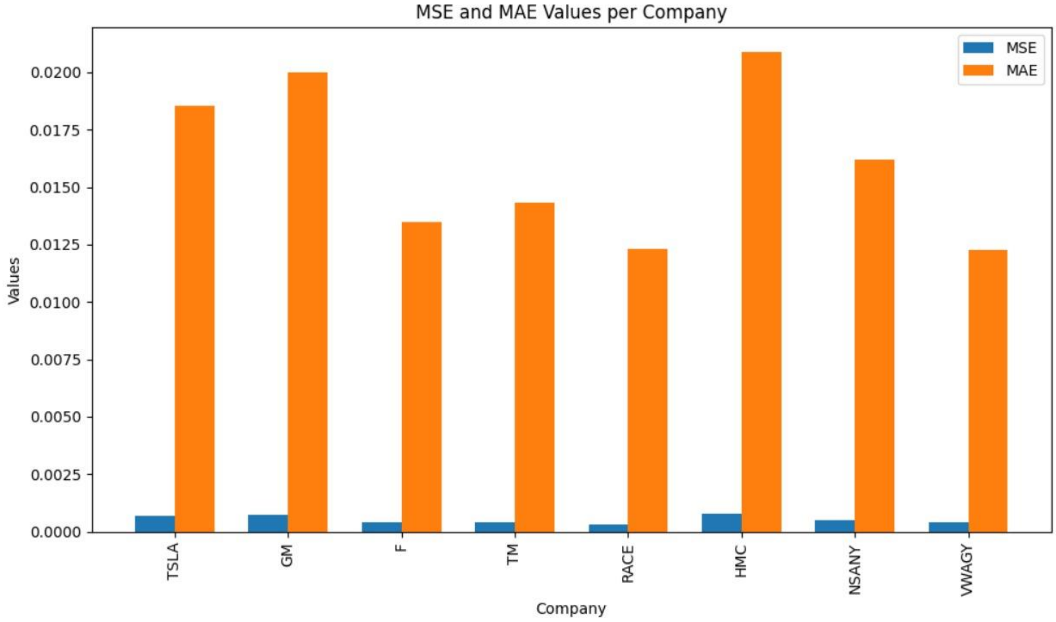
Plot:-



2. Automobile Sector:-

Company	MSE	MAE
TSLA	0.000681852	0.0185593
GM	0.000737473	0.0199751
F	0.000382727	0.0134932
TM	0.000384264	0.0143281
RACE	0.000308745	0.0122984
HMC	0.000763823	0.020901
NSANY	0.000492387	0.0161994
VWAGY	0.000419038	0.0122435

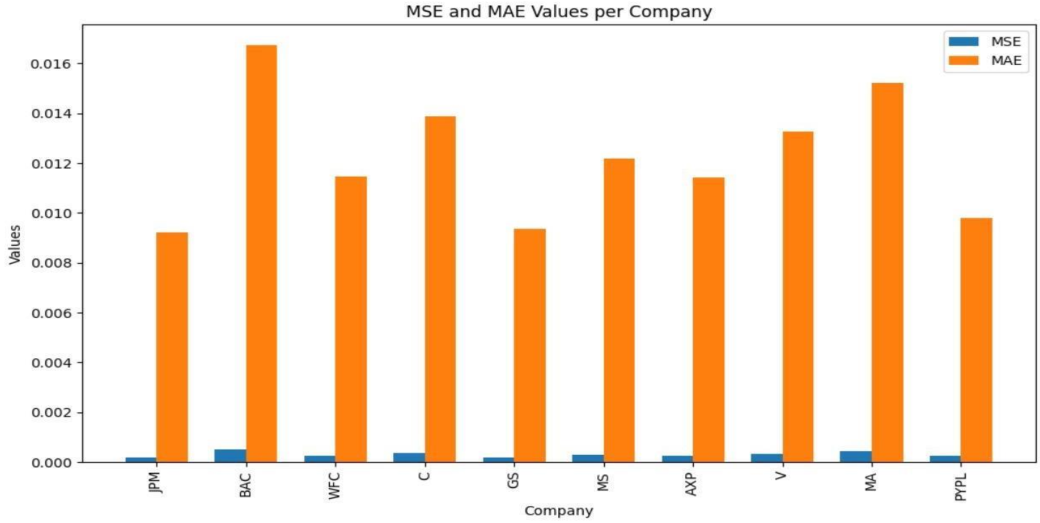
Plot:-



3. Business Sector:-

Company	MSE	MAE
JPM	0.000181829	0.00923746
BAC	0.000513833	0.0167215
WFC	0.000259363	0.0114663
C	0.000378699	0.0138934
GS	0.000188739	0.00934713
MS	0.000297143	0.0121669
AXP	0.000272048	0.0114088
V	0.000344522	0.0132722
MA	0.000431959	0.0152155
PYPL	0.000249976	0.00980173

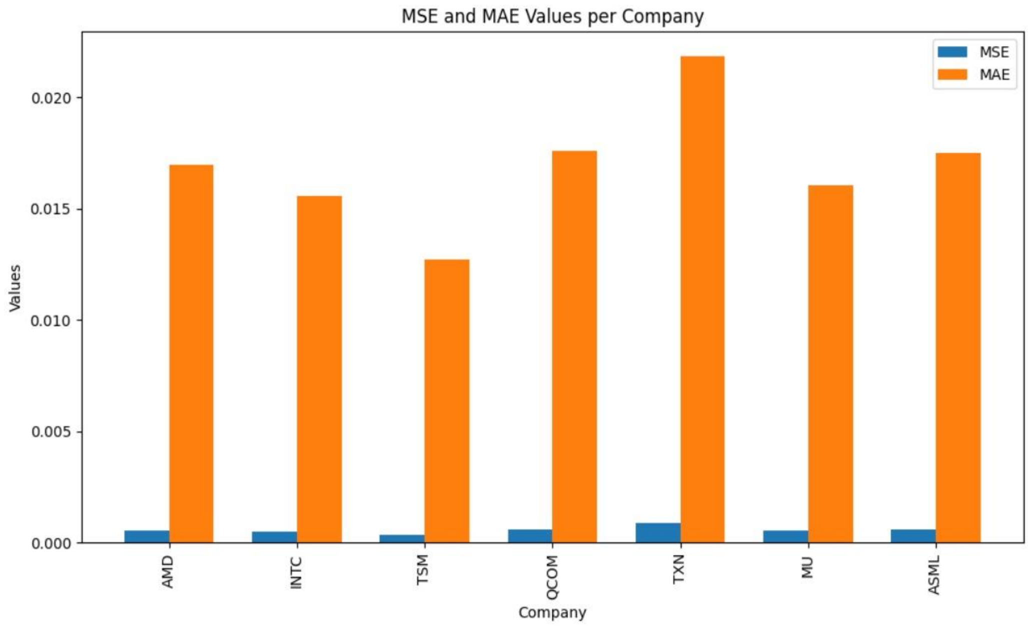
Plot:-



4. Semiconductor Sector:-

Company	MSE	MAE
AMD	0.000556086	0.016965
INTC	0.000486608	0.0155802
TSM	0.00035986	0.0127223
QCOM	0.00060782	0.0175871
TXN	0.000868805	0.0218425
MU	0.000528559	0.0160368
ASML	0.000594598	0.0175056

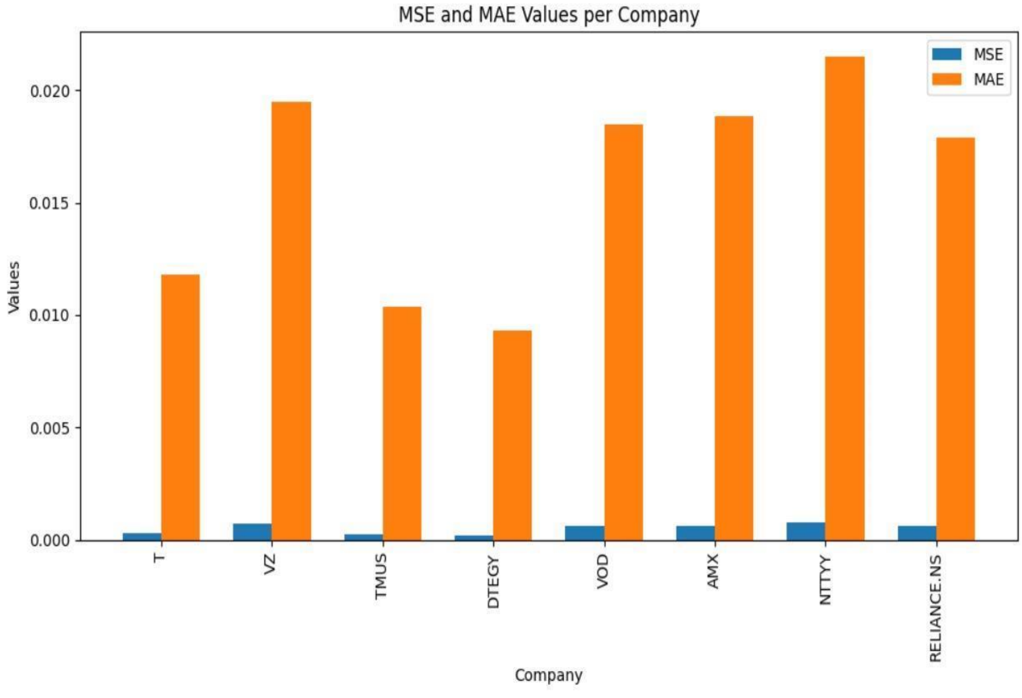
Plot:-



5. Telecom Sector:-

Company	MSE	MAE
T	0.000293711	0.0118175
VZ	0.000712154	0.0194858
TMUS	0.000248056	0.0103895
DTEGY	0.000185659	0.00928623
VOD	0.000644334	0.0184676
AMX	0.000596213	0.0188208
NTTYY	0.000803684	0.0215154
RELIANCE.NS	0.000595882	0.0179137

Plot:



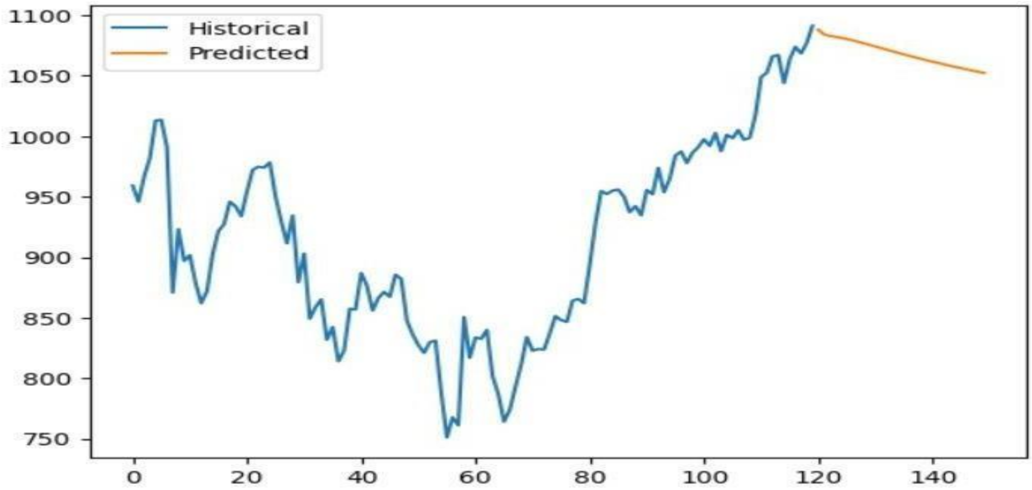
b. OUTPUT of the project – Future 30 days stock prediction of user input company using LSTM Model (one company from each sector):

1. User Input Company : NVIDIA (IT Sector)

   30 days stock prediction:

```
Predicted stock prices for NVDA for the next 30 days:
Day 1: 1088.141845703125
Day 2: 1084.130859375
Day 3: 1083.0042724609375
Day 4: 1082.209228515625
Day 5: 1081.3446044921875
Day 6: 1080.36328125
Day 7: 1079.26953125
Day 8: 1078.0843505859375
Day 9: 1076.831787109375
Day 10: 1075.534423828125
Day 11: 1074.2115478515625
Day 12: 1072.8787841796875
Day 13: 1071.5478515625
Day 14: 1070.22802734375
Day 15: 1068.925537109375
Day 16: 1067.644775390625
Day 17: 1066.3883056640625
Day 18: 1065.158203125
Day 19: 1063.9554443359375
Day 20: 1062.780029296875
Day 21: 1061.6318359375
Day 22: 1060.51025390625
Day 23: 1059.41455078125
Day 24: 1058.343994140625
Day 25: 1057.297607421875
Day 26: 1056.274658203125
Day 27: 1055.2740478515625
Day 28: 1054.294921875
Day 29: 1053.3365478515625
Day 30: 1052.39794921875
```

Plot of date vs historic price and predicted price:-

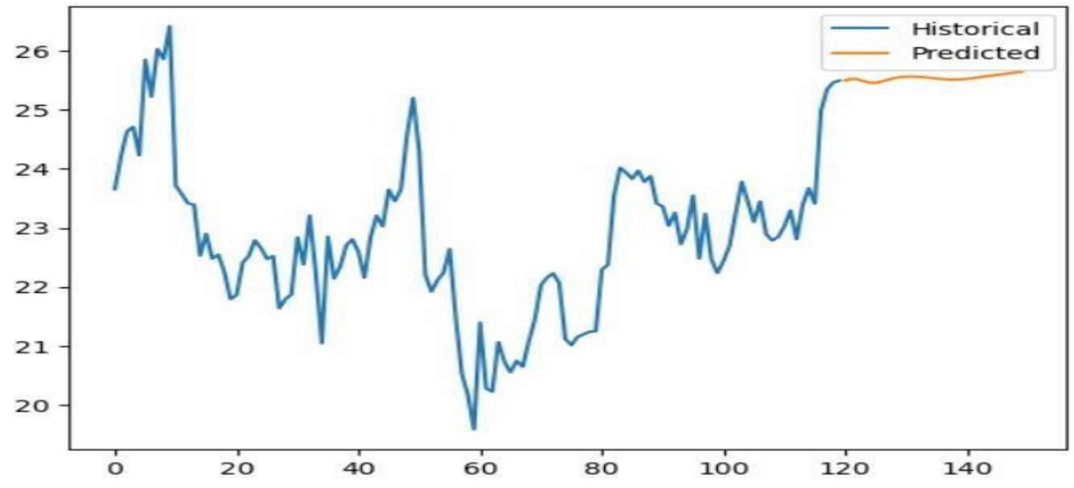


2. User Input Company : General Motors Company(Automobile Sector)

30 days stock prediction:-

```
Predicted stock prices for GM for the next 30 days:
Day 1: 25.500368118286133
Day 2: 25.526063919067383
Day 3: 25.521129608154297
Day 4: 25.485919952392578
Day 5: 25.459671020507812
Day 6: 25.459375381469727
Day 7: 25.48061180114746
Day 8: 25.51048469543457
Day 9: 25.53717613220215
Day 10: 25.5546932220459
Day 11: 25.56239128112793
Day 12: 25.56285858154297
Day 13: 25.558629989624023
Day 14: 25.55146026611328
Day 15: 25.542367935180664
Day 16: 25.531801223754883
Day 17: 25.5224552154541
Day 18: 25.51653289794922
Day 19: 25.515016555786133
Day 20: 25.51830291748047
Day 21: 25.526504516601562
Day 22: 25.538101196289062
Day 23: 25.55234146118164
Day 24: 25.566858291625977
Day 25: 25.580652236938477
Day 26: 25.593950271606445
Day 27: 25.60686683654785
Day 28: 25.619369506835938
Day 29: 25.63194465637207
Day 30: 25.645153045654297
```

Plot of date vs historic price and predicted price:-

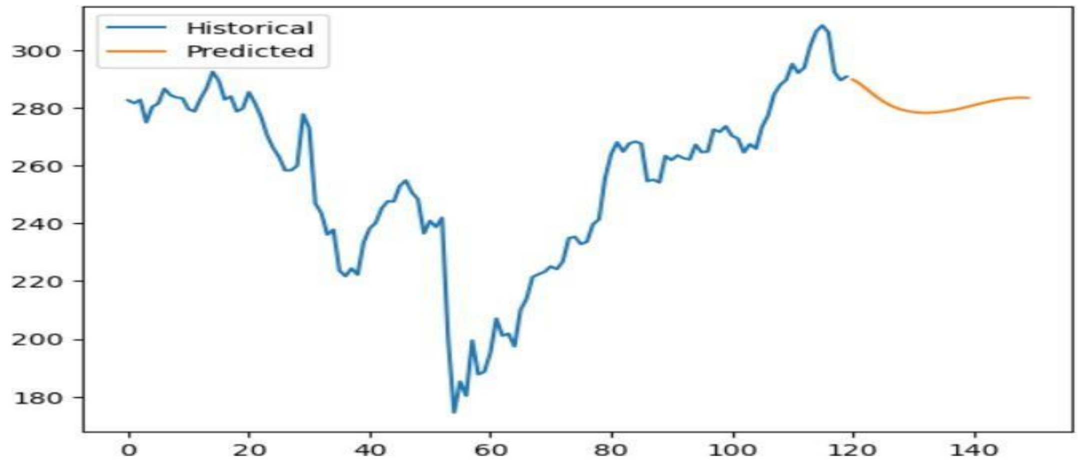


3. Bank of America Corporation(Business Sector)

30 days stock prediction:-

```
Predicted stock prices for BAC for the next 30 days:
Day 1: 289.7830505371094
Day 2: 288.63385009765625
Day 3: 287.0346984863281
Day 4: 285.3437194824219
Day 5: 283.7363586425781
Day 6: 282.3165283203125
Day 7: 281.126953125
Day 8: 280.1742858886719
Day 9: 279.44207763671875
Day 10: 278.90704345703125
Day 11: 278.5447082519531
Day 12: 278.3329162597656
Day 13: 278.2538146972656
Day 14: 278.2917785644531
Day 15: 278.4321594238281
Day 16: 278.66455078125
Day 17: 278.9803161621094
Day 18: 279.3692321777344
Day 19: 279.8180847167969
Day 20: 280.3125
Day 21: 280.83154296875
Day 22: 281.3540954589844
Day 23: 281.85858154296875
Day 24: 282.3257751464844
Day 25: 282.73480224609375
Day 26: 283.06732177734375
Day 27: 283.3089904785156
Day 28: 283.45440673828125
Day 29: 283.4975891113281
Day 30: 283.43756103515625
```

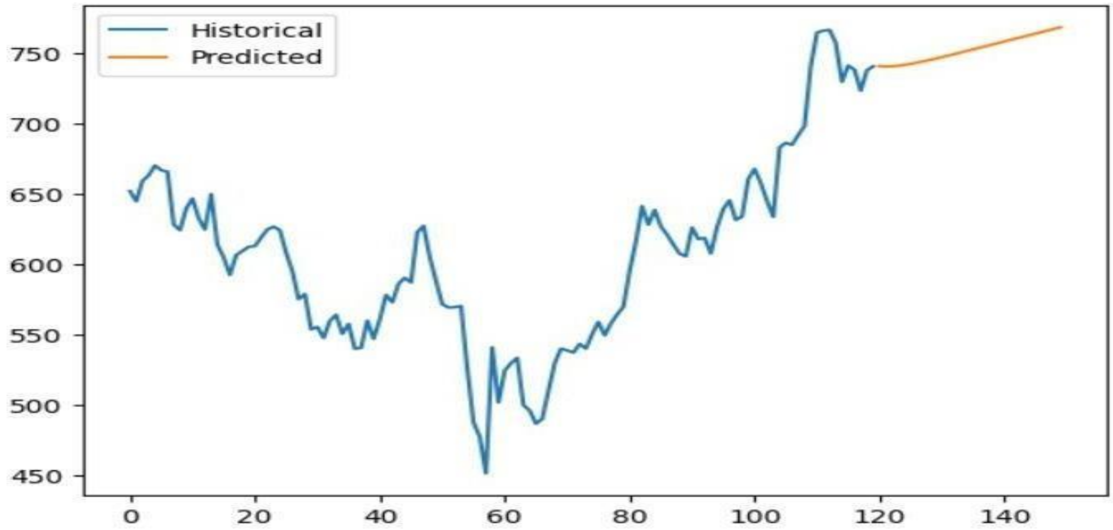
Plot of date vs historic price and predicted price:-



4. **Advanced Micro Devices (Semiconductor Sector) 30 days stock prediction:**

```
Predicted stock prices for AMD for the next 30 days:
Day 1: 740.8438110351562
Day 2: 740.503662109375
Day 3: 740.6214599609375
Day 4: 741.0590209960938
Day 5: 741.6784057617188
Day 6: 742.4224243164062
Day 7: 743.26123046875
Day 8: 744.1742553710938
Day 9: 745.1455078125
Day 10: 746.161865234375
Day 11: 747.2123413085938
Day 12: 748.2882690429688
Day 13: 749.3825073242188
Day 14: 750.4893188476562
Day 15: 751.6043701171875
Day 16: 752.7244873046875
Day 17: 753.8473510742188
Day 18: 754.97119140625
Day 19: 756.09521484375
Day 20: 757.21875
Day 21: 758.3414306640625
Day 22: 759.4634399414062
Day 23: 760.5848388671875
Day 24: 761.7061157226562
Day 25: 762.8274536132812
Day 26: 763.9491577148438
Day 27: 765.0715942382812
Day 28: 766.195068359375
Day 29: 767.3201293945312
Day 30: 768.4467163085938
```

Plot of date vs historic price and Predicted Price:-

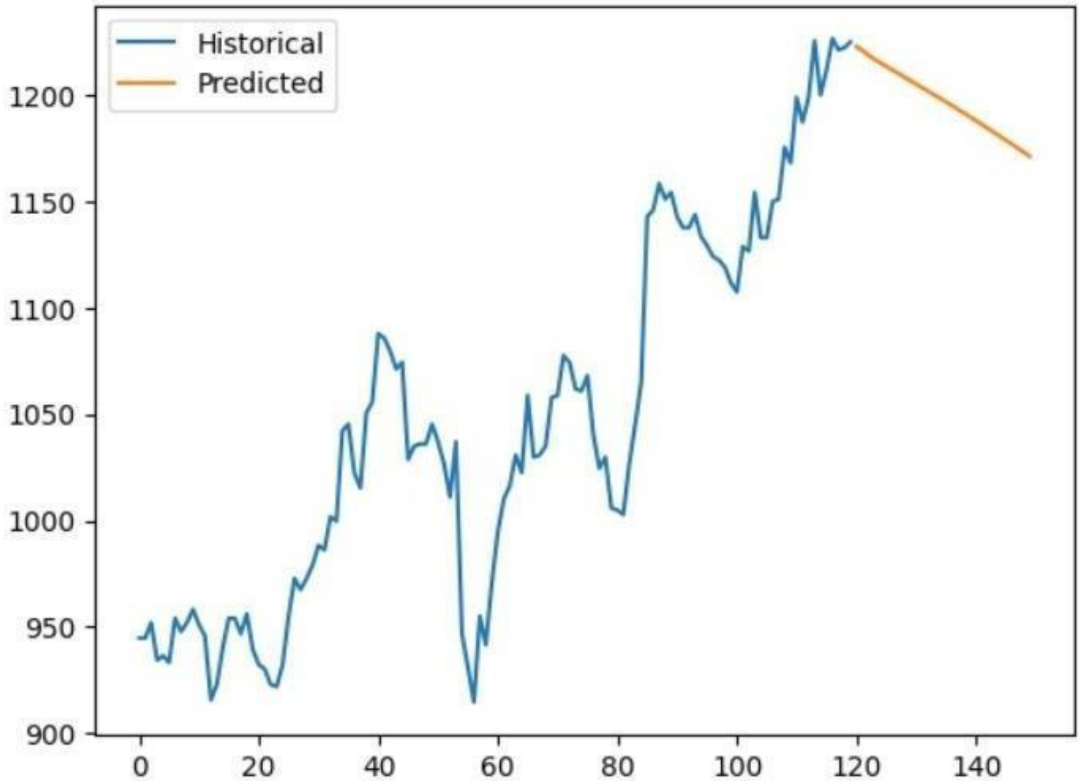


5. Vodafone Group (Telecom Sector)

30 days stock prediction:-

```
Predicted stock prices for VOD for the next 30 days:
Day 1: 1222.82666015625
Day 2: 1220.9866943359375
Day 3: 1218.809326171875
Day 4: 1216.9302978515625
Day 5: 1215.218017578125
Day 6: 1213.5423583984375
Day 7: 1211.86181640625
Day 8: 1210.179931640625
Day 9: 1208.505859375
Day 10: 1206.84228515625
Day 11: 1205.1854248046875
Day 12: 1203.528564453125
Day 13: 1201.865478515625
Day 14: 1200.19091796875
Day 15: 1198.5015869140625
Day 16: 1196.794921875
Day 17: 1195.0703125
Day 18: 1193.3271484375
Day 19: 1191.5660400390625
Day 20: 1189.7874755859375
Day 21: 1187.9925537109375
Day 22: 1186.181884765625
Day 23: 1184.3564453125
Day 24: 1182.5169677734375
Day 25: 1180.6641845703125
Day 26: 1178.798583984375
Day 27: 1176.9207763671875
Day 28: 1175.03125
Day 29: 1173.1300048828125
Day 30: 1171.21752929687 5
```

Plot of date vs historic price and predicted price:-



c. Comparative Analysis:-

We choose LSTM (Long Short-Term Memory) and CNN(Convolutional Neural Network) to build our stock prediction project. Here is a comparative analysis of these two models.

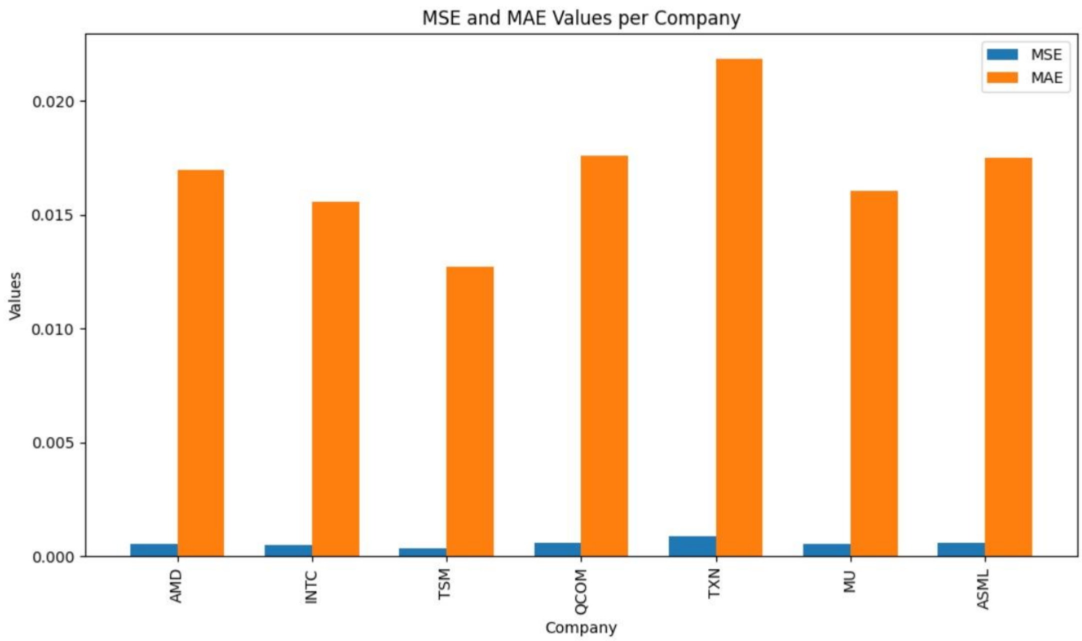
Comparison based on the Mean Square Error(MSE) and Mean Absolute Error(MAE) metrices (2 sectors for example):

1. ***Semiconductor Sector:-***

a. LSTM Model:-

Company	MSE	MAE
AMD	0.000556086	0.016965
INTC	0.000486608	0.0155802
TSM	0.00035986	0.0127223
QCOM	0.00060782	0.0175871
TXN	0.000868805	0.0218425
MU	0.000528559	0.0160368
ASML	0.000594598	0.0175056

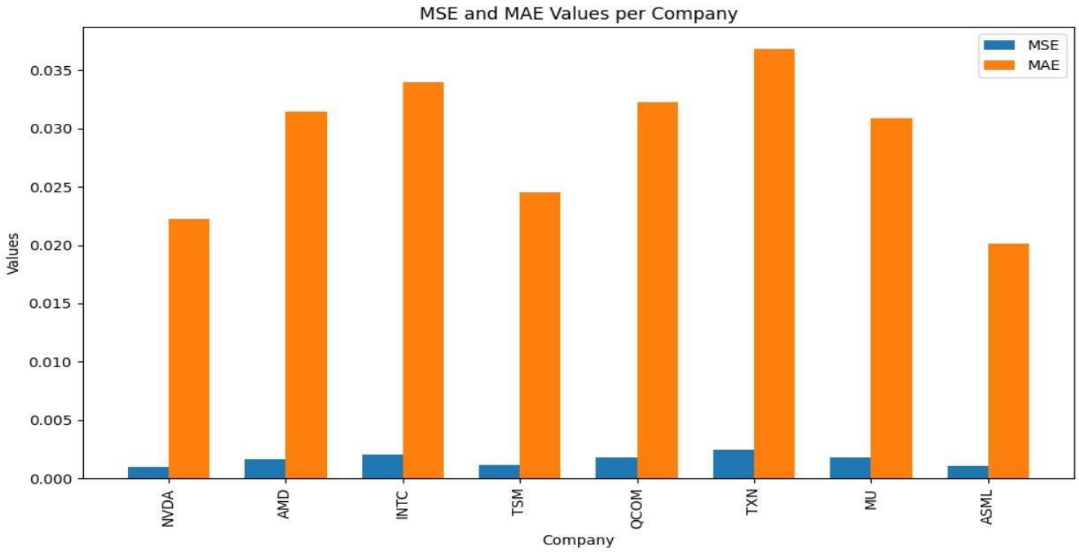
Plot:-



b. CNN Model:-

Company	MSE	MAE
NVDA	0.000982873	0.022246
AMD	0.0016733	0.0314022
INTC	0.00204163	0.0339617
TSM	0.00116947	0.0245016
QCOM	0.00180601	0.0322875
TXN	0.00242823	0.0368141
MU	0.00181401	0.0309
ASML	0.00104011	0.0201522

Plot:-

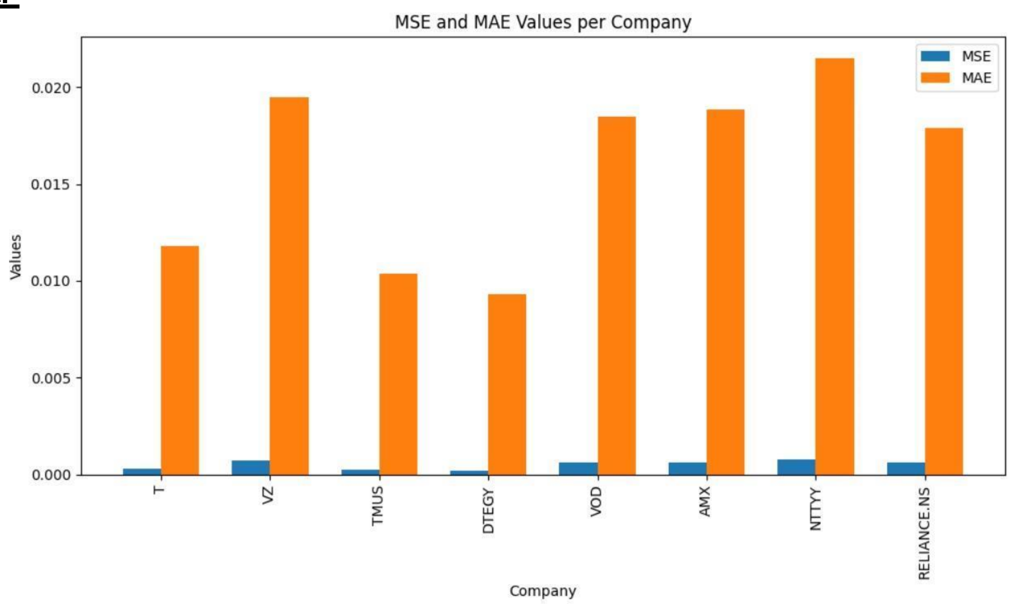


2. *Telecom Sector:-*

a. LSTM Model:-

Company	MSE	MAE
T	0.000293711	0.0118175
VZ	0.000712154	0.0194858
TMUS	0.000248056	0.0103895
DTEGY	0.000185659	0.00928623
VOD	0.000644334	0.0184676
AMX	0.000596213	0.0188208
NTTYY	0.000803684	0.0215154
RELIANCE.NS	0.000595882	0.0179137

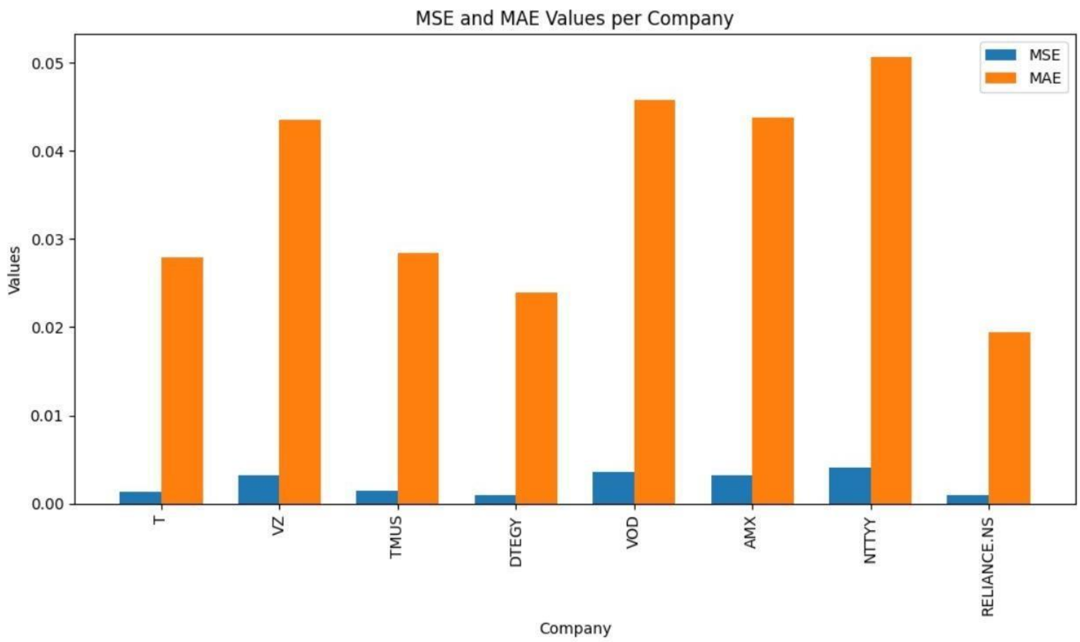
Plot:-



b. CNN Model:-

Company	MSE	MAE
T	0.00135446	0.0279079
VZ	0.00326936	0.043558
TMUS	0.00145208	0.0284489
DTEGY	0.00101069	0.0239105
VOD	0.00363149	0.0457365
AMX	0.00318693	0.0437134
NTTYY	0.00411258	0.0506727
RELIANCE.NS	0.000928171	0.0194802

Plot:-



Average MSE for LSTM Model : 0.000359363

Average MAE for LSTM Model : 0.01028623

Average MSE for CNN Model: 0.00281401

Average MAE for CNN Model: 0.0299105

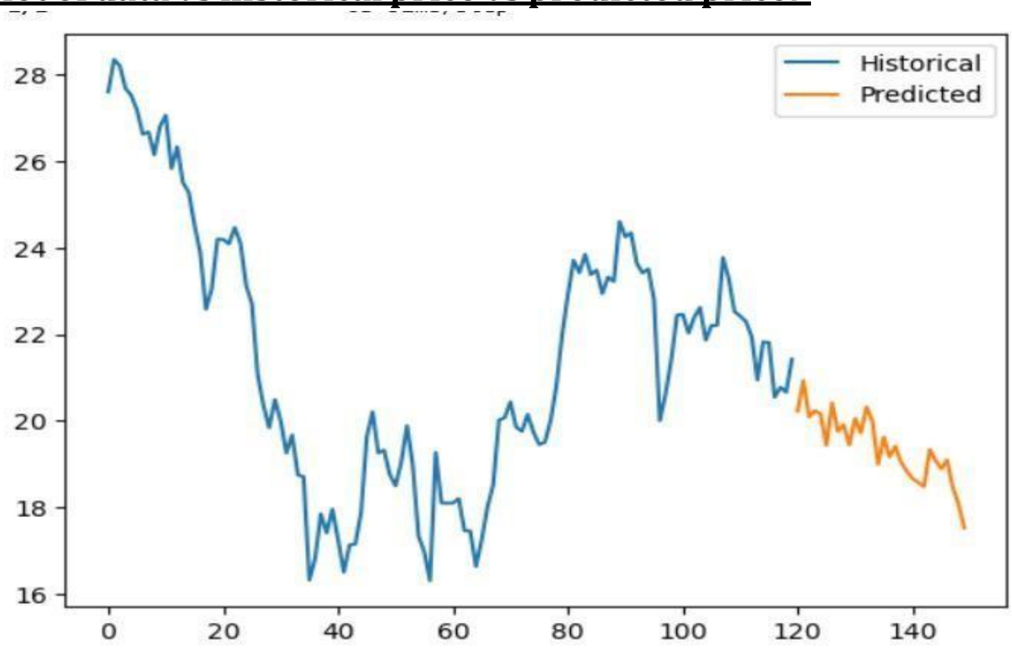
OUTPUT OF CNN MODEL – 30 days stock prediction using CNN model(2 companies for example):-

1. **User Input Company – Tesla (Automobile Sector)**

30 days stock price:-

```
Predicted stock prices for TSLA for the next 30 days:
Day 1: 20.226137161254883
Day 2: 20.93264389038086
Day 3: 20.08633041381836
Day 4: 20.234285354614258
Day 5: 20.15495491027832
Day 6: 19.428953170776367
Day 7: 20.412092208862305
Day 8: 19.753902435302734
Day 9: 19.903337478637695
Day 10: 19.438770294189453
Day 11: 20.055700302124023
Day 12: 19.718984603881836
Day 13: 20.320510864257812
Day 14: 20.004684448242188
Day 15: 18.9974308013916
Day 16: 19.621557235717773
Day 17: 19.164817810058594
Day 18: 19.40819549560547
Day 19: 19.03841209411621
Day 20: 18.83202362060547
Day 21: 18.66323471069336
Day 22: 18.569469451904297
Day 23: 18.47687530517578
Day 24: 19.33493995666504
Day 25: 19.085599899291992
Day 26: 18.884485244750977
Day 27: 19.093149185180664
Day 28: 18.45977020263672
Day 29: 18.08025360107422
Day 30: 17.518741607666016
```

<u>Plot of data vs historical price vs predicted price:-</u>

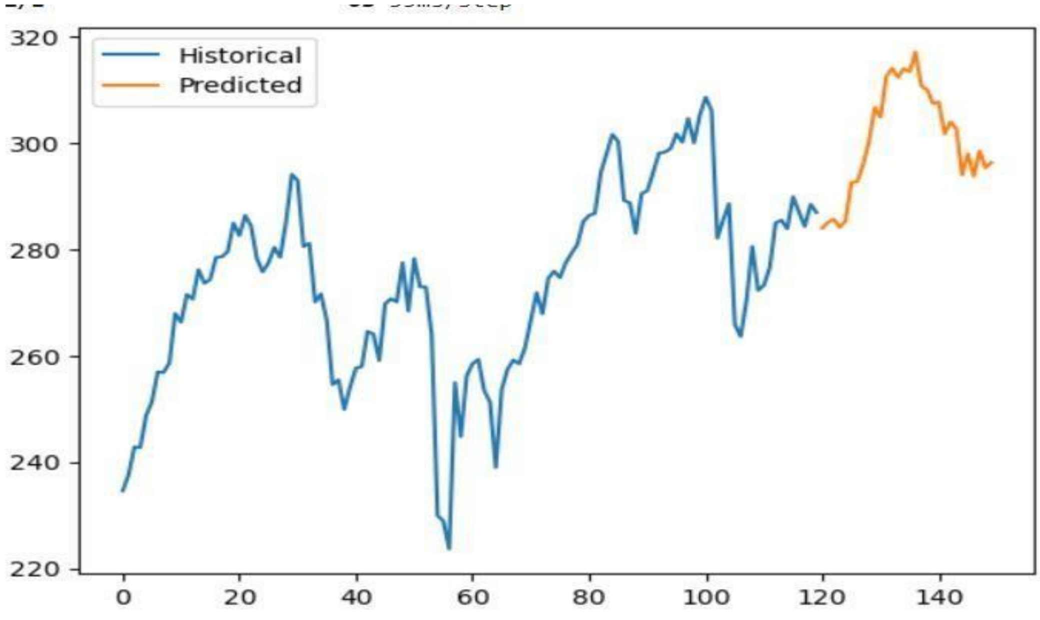


2. User Input Company – VISA (Business Sector)

30 days future stock price:-

```
Predicted stock prices for V for the next 30 days:
Day 1: 283.9145812988281
Day 2: 285.0324401855469
Day 3: 285.59423828125
Day 4: 284.0526428222656
Day 5: 285.3116149902344
Day 6: 292.4775390625
Day 7: 292.6728820800781
Day 8: 295.8462219238281
Day 9: 300.07305908203125
Day 10: 306.5377197265625
Day 11: 304.763916015625
Day 12: 312.44677734375
Day 13: 313.9352722167969
Day 14: 312.2372131347656
Day 15: 313.90179443359375
Day 16: 313.2931823730469
Day 17: 317.0180969238281
Day 18: 310.6014099121094
Day 19: 309.9266662597656
Day 20: 307.30975341796875
Day 21: 307.5994567871094
Day 22: 301.62359619140625
Day 23: 303.8854675292969
Day 24: 302.5882568359375
Day 25: 293.9245910644531
Day 26: 297.865966796875
Day 27: 293.6742858886719
Day 28: 298.4500427246094
Day 29: 295.2609558105469
Day 30: 296.23455810546875
```

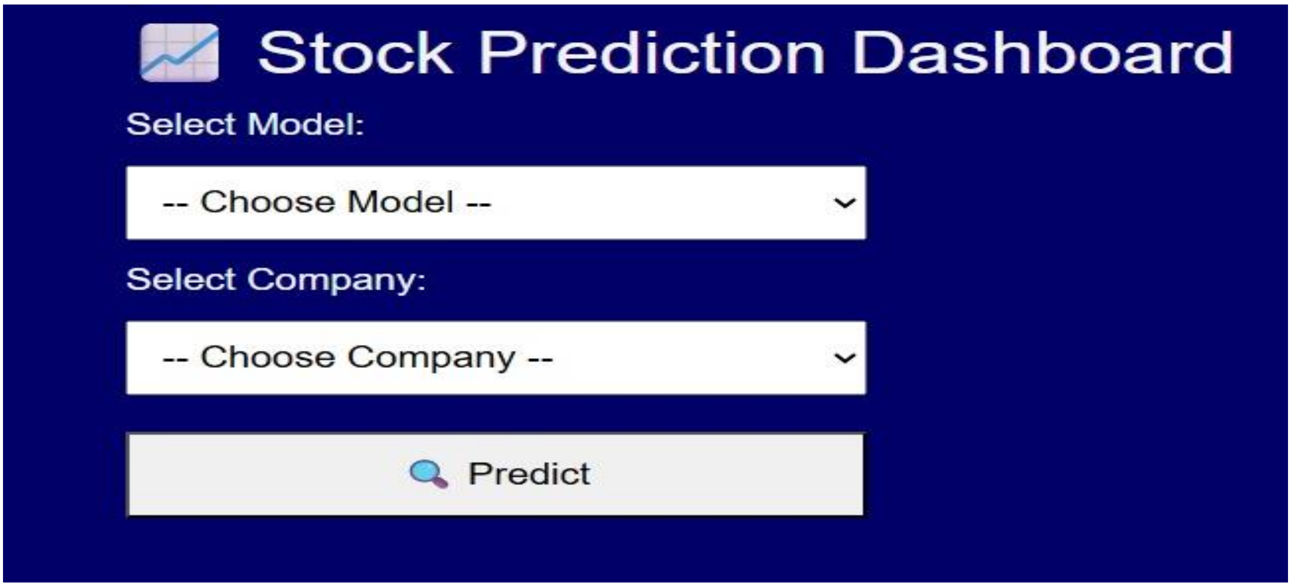
<u>Plot of date vs historical price and predicted price:-</u>

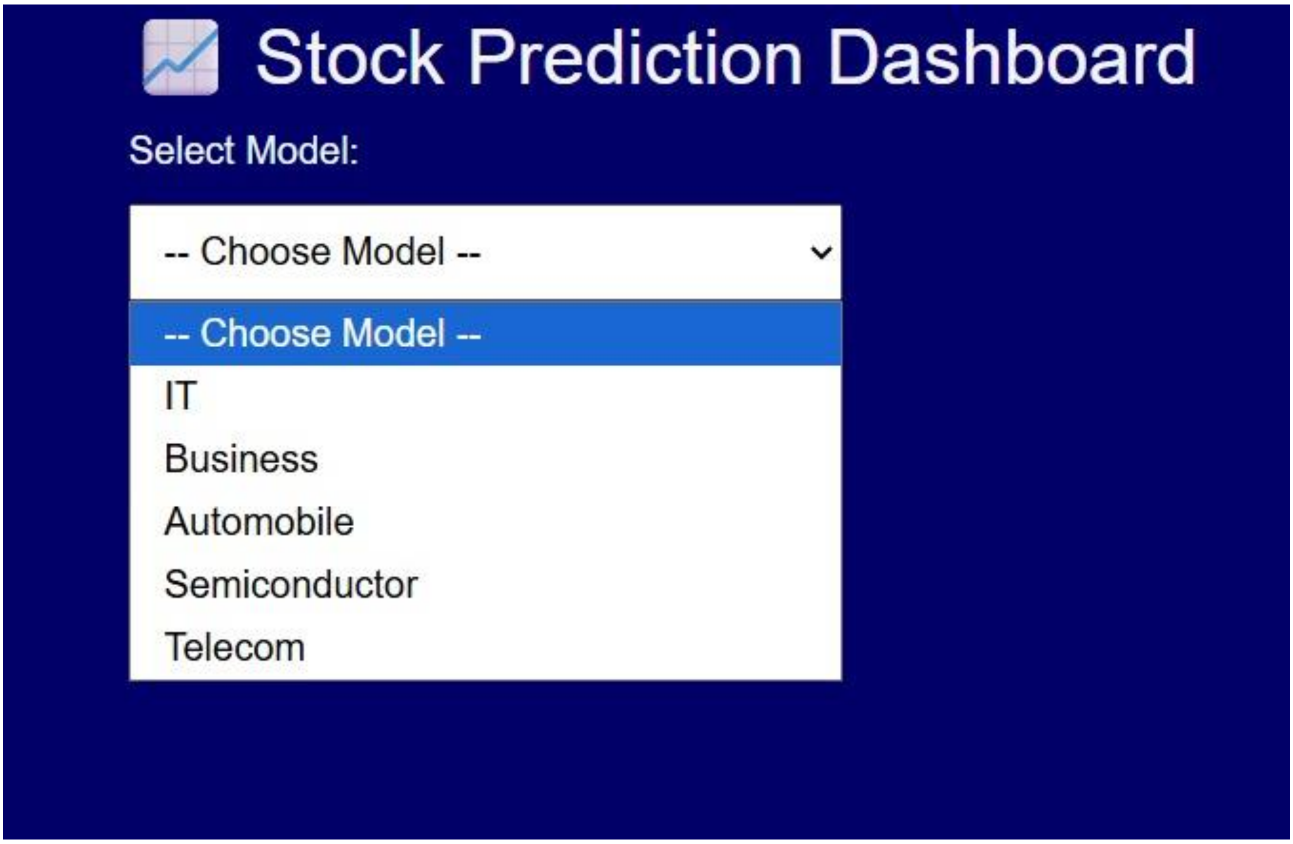


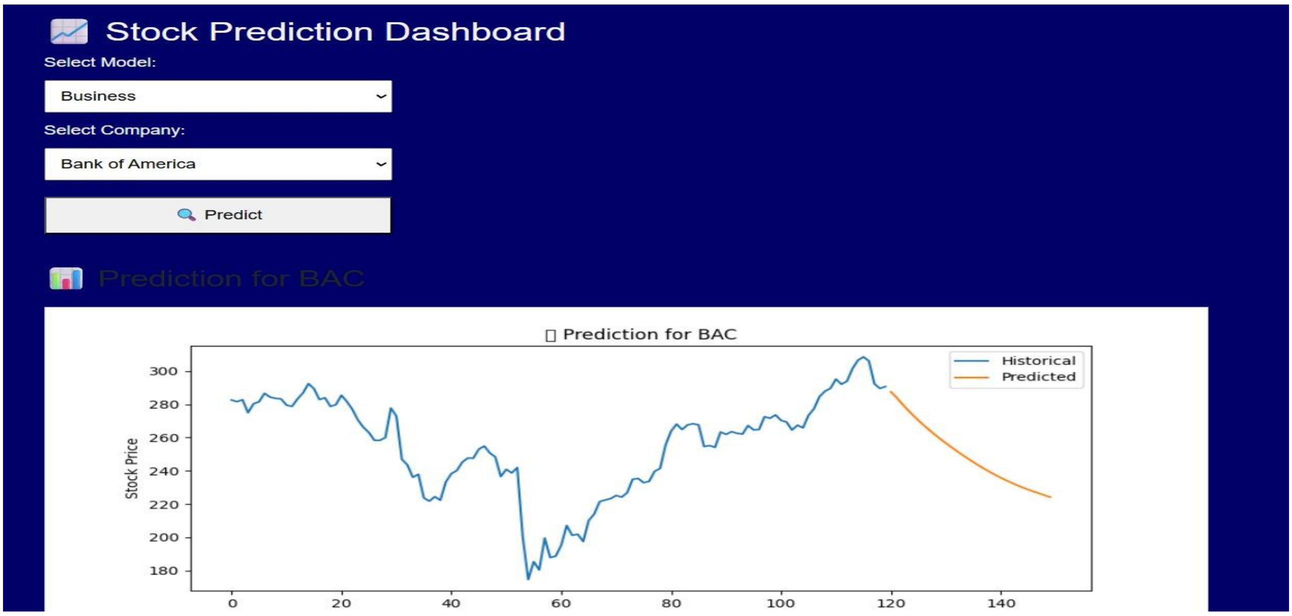
Observation:-

The MSE and MAE values for LSTM Model is comparatively lesser than those of CNN Model. It indicates that LSTM model has higher accuracy than CNN model on prediction of future stock prices. So, we choose LSTM Model for our final project.

Final output after the model deployment in a full stack project







Date	Predicted Price
2025-07-16	287.54
2025-07-17	284.18
2025-07-18	280.46
2025-07-19	276.90
2025-07-20	273.53
2025-07-21	270.36
2025-07-22	267.36
2025-07-23	264.51
2025-07-24	261.79
2025-07-25	259.17
2025-07-26	256.65
2025-07-27	254.22
2025-07-28	251.86
2025-07-29	249.58

6. Conclusion

Our comprehensive project culminated in the successful development of an advanced Long Short-Term Memory (LSTM) model, specifically designed for accurate 30-day stock price forecasting. This endeavor has not only validated the model's exceptional ability to capture the intricate, non-linear dynamics of various stock markets but has also laid a robust foundation for delivering truly actionable insights, empowering users to make more informed and strategic investment decisions.

1. Confirmed Forecasting Capability

The developed LSTM model demonstrated remarkably strong performance in predicting stock price movements, effectively capturing intricate temporal dependencies, trends, and even subtle shifts in market sentiment embedded within historical financial data. Through rigorous training and validation, the model proved its capacity to learn from past patterns, enabling it to generate reliable short-to-medium term forecasts. This predictive power is a critical asset for investors and analysts, providing a crucial edge for developing more resilient investment strategies, optimizing portfolio allocations, and making timely entry or exit decisions in volatile markets.

2. Future Work: Universal Market Generalization

Our next significant phase is dedicated to transcending company-specific predictions by uncovering and leveraging universal market patterns across diverse datasets. This ambitious objective includes the implementation of advanced normalization techniques and sophisticated feature engineering to identify hidden correlation columns and establish interconnections between seemingly disparate companies or sectors. By understanding these broader market forces, our aim is to develop a highly adaptable and robust prediction framework that can reliably generalize and forecast for any new, unseen company stock, moving beyond the limitations of pre-trained data and paving the way for a truly versatile and scalable financial analytics tool applicable across the entire market landscape.

7. APPENDICES

1. *References:-* **Hands-On-Machine Learning with Scikit-Learn and Tensorflow By Aurelien Geron**

2. *Github link for code:-* https://github.com/DeepJDUTTA/Stock_prediction.git

3. *Presentation link:-* https://docs.google.com/presentation/d/1Blr_GH6WkjkAJ1v8h8pnJxkQyIT89zUa/edit?usp=drivesdk&ouid=111858391680779286222&rtpof=true&sd=true