

A STUDY ON STOCK PRICE PREDICTION USING LSTM MODEL

Submitted by

MD Naimur Rahaman

(Registration Number :-163632 of 2022-2023)

(Class Roll Number :- 002210503040)

(Examination Roll Number :-MCA244039)

Under the supervision of

Prof.Mahantapas Kundu

in the

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Jadavpur University, Kolkata - 700032

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF RECOMMENDATION

To Whom It May Concern

This is to certify that the thesis entitled “**A STUDY ON STOCK PRICE PREDICTION USING LSTM MODEL**” has been satisfactorily carried out by **MD Naimur Rahaman** (University Registration Number- 163632 of 2022-2023, Class Roll Number- 002210503040, Examination Roll Number- MCA244039). It is a bona-fide record of work carried out under my guidance and supervision and be accepted in partial fulfilment of the requirement for the degree of **Master of Computer Application (MCA)** from the **Department of Computer Science & Engineering, Faculty of Engineering and Technology, Jadavpur University, Kolkata - 700032.**

Prof. Mahantapas Kundu
Project Supervisor,
Department of Computer Science and Engineering,
Jadavpur University

Prof. (Dr.) Nandini Mukhopadhyay
Head of the Department,
Department of Computer Science and Engineering,
Jadavpur University

Prof. Dipak Laha
Dean,
Faculty Council of Engineering and
Technology, Jadavpur University, Kolkata-032

Faculty of Engineering and Technology
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled "**A STUDY ON STOCK PRICE PREDICTION USING LSTM MODEL**" is a bona-fide record of work carried out by **MD Naimur Rahaman** (University Registration Number :- 163632 of 2022-2023, Class Roll Number :- 002210503040, Examination Roll:- MCA244039) in partial fulfilment of the requirements for the award of the degree of **Master of Computer Application (MCA)** from the **Department of Computer Science & Engineering, Faculty of Engineering and Technology, Jadavpur University, Kolkata -700032.**

It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion withdrawn therein, but approves the thesis only for the purpose for which it has been submitted.

Signature of Examiner 1

Date :-

Signature of Examiner2

Date :-

DECLARATION OF COMPLIANCE WITH ACADEMIC ETHICS

I hereby declare that the work described in this thesis titled **“A STUDY ON STOCK PRICE PREDICTION USING LSTM MODEL”** is an outcome of my requirements of my genuine and independent work, as part of my **Master of Computer Application (MCA)** program.

All information in this document have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, all materials and results that are not original to this work have been fully cited and referenced.

MD Naimur Rahaman

Registration Number :- 163632 of 2022-2023

Class Roll Number :- 002210503040

Examination Roll Number :- MCA244039

Thesis Title: A study on stock price prediction using LSTM model

Signature with Date

ACKNOWLEDGEMENTS

I would like to extend my sincerest gratitude to my honourable thesis guide, **Prof. Mahantapas Kundu**, from the Department of Computer Science and Engineering, Jadavpur University, Kolkata, for his invaluable and meticulous guidance, constant encouragement, and motivating words during the entire period of my research work which culminated into my master's thesis.

I am thankful to all the teaching and non-teaching staff who helped me to have a smooth journey during the time of my research.

I would also like to thank my parents and my friends for their never-ending support during the time of my research. Any kind of suggestions or criticism will be highly appreciated and acknowledged.

Date :-

Name :- MD Naimur Rahaman

Registration Number :-163632 of 2022-2023

Class Roll Number :-002210503040

Examination Roll Number:-MCA244039

ABSTRACT

Stock market is an industry where lots of data is generated daily and benefits are reaped on the basis of accurate prediction. Many people invest in stock market having some prediction and more luck. A decision in stock market plays an important role in the investor's life. Also, stock market is a very complex system and non-linear in nature. So, then it is very difficult to analyse all the impacting factors before making a decision. Making decision with traditional techniques may be time consuming and may not ensure the reliability of the prediction.

Data from stock market is a time series data and different variations of neural networks are widely being used for stock forecasting and prediction problems. Among various architectures of deep neural networks, Long Short-Term Memory (LSTM) is one that supports time steps of arbitrary sizes and is free of vanishing gradient problem. Furthermore, stock price analysis is becoming popular in predicting the stock market.

Research in recent years has shown that LSTM network models perform well in stock price prediction, and it is considered one of the most precise prediction techniques, especially when it is applied to longer prediction ranges.

This research attempts to explore the capabilities of Long Short-Term Memory a type of Recurrent Neural Networks in the prediction of future stock prices. We will use multiple layers of neurons to train the model and try to predict the actual price using different dataset of the stocks for the upcoming days. It is assumed that the prediction will be done for the shorter timeframe.

CONTENTS

	Page
1. CHAPTER - 1: INTRODUCTION	
1.1 Stock Price Prediction.	9
1.2 Background and Motivation.	10
1.3 Challenges	10
1.4 Machine Learning Techniques in Stock Market Prediction.	10
1.5 Advantages of Neural Networks in Financial Forecasting.	11
1.6 LSTM Neural Networks for Stock Price Prediction.	11
1.7 Comparison with Other Machine Learning Models.	11
2. CHAPTER - 2: RELATED WORK	12-14
3. CHAPTER - 3: METHODOLOGY	
3.1 Long Short-Term Memory(LSTM)	15
3.1.1 Three Main Components of LSTM.	15-16
3.2 Working of LSTM.	16-19
4. CHAPTER -4: EXPERIMENTAL RESULTS	
4.1 Implementation.	20
4.1.1 Technique Methods Used.	20-22
4.2 Steps for Model Implementation.	23
4.3 Dataset Description	24
4.3.1 Raw Data	24
4.3.2 Dataset used for Testing	25
4.4 Visualization of Feature Data	27
4.3.2 Data Pre-processing	27
4.3.3 Feature Extraction	27-29

	Page
4.5 Model Performance	30
4.5.1 Model Summary	30
4.5.2 Loss Function vs Epoch	30-31
4.6 MEAN SQUARED ERROR(MSE)	32
4.5.1 Formula of MSE	32
4.5.2 MSE of the experiment	33
4.7 Mean Absolute Percentage Error (MAPE)	34
4.8 Model Accuracy	34
4.9 RESULTS.....	35-36
 5. CHAPTER- 5:CONCLUSIONS	
5.1 Limitations.	37
5.2 Future Work	37
5.2.1 Expanding the Dataset	38
5.2.2 Improving Model Performance	38
 REFERENCES.....	 39-40

CHAPTER 1: INTRODUCTION

1.1 Stock Price Prediction

Stock price prediction is the process of using historical data, statistical analysis, and machine learning algorithms to forecast the future movement of stock prices. It's a critical aspect of financial analysis and trading, as accurate predictions can help investors make informed decisions about buying, selling, or holding stocks.

Modeling and Forecasting of the financial market has been an attractive topic to scholars and researchers from various academic fields. Financial market is an abstract concept where financial commodities such as stocks, bonds and precious metals transactions happen between buyers and sellers. In the present scenario of financial market world, especially in the stock market, forecasting the trend or the price of stocks using machine learning techniques and artificial neural networks are the most attractive issue to be investigated. As Giles et. al [1] explained, Financial forecasting is an instance of signal processing problem which is difficult because of high noise, small sample size, non-stationarity, and non-linearity. The noisy characteristics means the incomplete information gap between past stock trading price and volume with future price. Stock market is sensitive with political and macroeconomic environment.

However, these two kinds of information are too complex and unstable to gather. The above information that cannot be included in features are considered as noise. The sample size of financial data is determined by real world transaction records. On one hand, a larger sample size refers a longer period of transaction records; on the other hand, large sample size increases the uncertainty of financial environment during the sample period. In this research, we use hourly stock data instead of daily data in order to reduce the probability of uncertain noise, and relatively increase the sample size within a certain period of time. By non-stationarity, one means that the distribution of stock data is various during time changing. Non-linearity implies that feature correlation of different individual stocks is various [2].

1.2 Background and Motivation

- **Financial Market Dynamics:** Explain the financial market as an abstract concept where transactions of stocks, bonds, and precious metals occur between buyers and sellers. Highlight the attractiveness of financial market modeling and forecasting to researchers from various fields.
- **Current Challenges:** Discuss the present challenges in forecasting stock prices, such as market volatility, complexity, and sensitivity to external factors. Emphasize the potential of machine learning and neural networks to address these challenges.[3]

1.3 Challenges

- **High Noise and Small Sample Size:** Elaborate on the difficulties posed by noisy data and limited historical records. Explain how these issues complicate the prediction of future stock prices.
- **Non-stationarity and Non-linearity:** Describe the dynamic nature of financial data, where statistical properties change over time, and relationships between variables are non-linear.
- **Sensitivity to Political and Macroeconomic Factors:** Discuss the impact of external factors on stock prices and the challenge of incorporating this complex, unstable information into prediction models.

1.4 Machine Learning Techniques in Stock Market Prediction

- **Classical Algorithms:** Review traditional methods like Linear Regression, Random Walk Theory, Moving Average Convergence/Divergence (MACD), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA).
- **Advanced Techniques:** Discuss the shift towards machine learning techniques such as Support Vector Machine (SVM), Random Forest (RF), and neural network models like Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) networks.

1.5 Advantages of Neural Networks in Financial Forecasting

- **Low Data Requirements and Flexibility:** Explain how neural networks can operate with less stringent data requirements and handle non-linear relationships.
- **Learning Complex Patterns:** Describe the ability of neural networks to learn from data through multiple hidden layers, capturing complex patterns that traditional models might miss.
- **Benefits of RNNs and LSTMs:** Highlight the unique strengths of Recurrent Neural Networks (RNNs) and LSTM networks in processing time-sequential data, making them suitable for financial forecasting.[4]

1.6 LSTM Neural Networks for Stock Price Prediction

- **Introduction to LSTM:** Introduce Long Short-Term Memory (LSTM) networks, focusing on their ability to retain information over long periods.
- **Suitability for Stock Price Time Series:** Discuss why LSTMs are ideal for handling the stochastic, non-stationary nature of stock price time series data.
- **Successful Applications:** Provide examples from the literature where LSTM networks have outperformed other models in financial forecasting tasks.

1.7 Comparison with Other Machine Learning Models

- **Performance of LSTM Models:** Compare the accuracy and efficiency of LSTM models with other machine learning techniques.
- **Handling Long-term Dependencies:** Explain how LSTMs' ability to manage long-term dependencies gives them an edge over other models in predicting stock prices.[14]

CHAPTER 2: RELATED WORK

Since the evolution of artificial intelligence many attempted to combine deep learning and machine learning using core principles. Artificial intelligence methods include convolutional neural network, multi-layer perceptron, naive Bayes network, back propagation network, recurrent neural network, single-layer LSTM, support vector machine and many more .

A study in 2018 by Sima et al. has shown a comparison between LSTM and ARIMA [5] a model used for analysing time series data. This study focused on implementing and applying financial data on an LSTM which was superior to the ARIMA model. Further, a study by Khaled A. Althelaya et al. in 2018 [6], evaluated the performance of bidirectional and stacked LSTM for stock market prediction. The performance of the tuned models where also compared with a shallow and an unidirectional LSTM.

The study concluded that the bidirectional and stacked LSTMs had better performance for short term prices opposed to the long term prediction results. Further, the results have shown that deep architecture outperformed their shallow counter parts. Another example is a study made in 2015 by Roondiwala et al. , which attempted to create a model based on LSTM for an accurate read of the future market indices. The researchers further analysed various hyperparameters of their model and mainly the number of epochs used with various variable combinations of the market.

At the end they concluded that using multiple variables (High/Low/Open/Close) resulted to the least errors. Later on, in 2020 Hao et al. ,proposed a hybrid model based on LSTM and multiple time scale feature learning and compared it to other existing models.

The study also compared models based on single time scale feature learning. Furthermore, design was made to combine the output representation from three LSTM based architectures. A study by David G. McMillan attempts to understand the variables proxy for changes in the expected future cash flow.

It was concluded that forecasting combinations outperform single future models. Looking into the combinations of the features, the hyperparameters and different LSTM variations would allow to better understand LSTMs and expand on the research of this topic in general. From the related studies we expect that deep architectures would outperform their shallow counterparts and multiple feature combinations would perform generally better than single features.

Kara et al. used support vector machine (SVM) and artificial neural network to predict movement in the daily Istanbul Stock Exchange National 100 Index from 1997 to 2007 (Kara et al., 2011). The authors selected ten technical indicators as input for their model. Experimental results showed that the average performance of the artificial neural network model was significantly better than that of the SVM model. Karmiani et al. compared LSTM, Backpropagation, SVM, and Kalman filter to predict stock price (Karmiani, Kazi, Nambisan, Shah, & Kamble, 2019). The stock price of selected nine companies were considered for the prediction. LSTM was the best choice in terms of prediction accuracy with low variance. Yu and Yan combined phase-space reconstruction method for time series analysis and LSTM model to predict the stock price (Yu & Yan, 2019). Various market environments such as the S&P 500, DJIA, Nikkei 225, Hang Seng Index, China Securities Index 300, and ChiNext index were considered for the analysis. This prediction model was built by taking the historical price only. The outcomes of LSTM with Multilayer Perception, Support Vector Regressor, and ARIMA were compared. Authors claimed that the LSTM model outperformed other models for S&P 500 data. Gao et al. conducted a comparative study of four machine learning algorithms — Multilayer Perceptron, LSTM, Convolutional Neural Network, and Uncertainty-Aware Attention—to predict the next day's stock price (Gao, Zhang, & Yang, 2020). The S&P 500 index, CSI 300 index, and Nikkei 225 index were taken to represent the most developed market, the less developed market, and the developing market. Open price, close price, trading volume, Moving Average Convergence Divergence, Average True Range, exchange rate, and interest rate were considered predictors. The outcome of the study suggested that Uncertainty-Aware Attention's performance was slightly better than other models. Moreover, additional predictors such as the volatility index and the unemployment rate could improve the model's performance.[17]

Few more studies and researches may also include:

- Fischer and C. Krauss (2018): In their paper "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions," Fischer and Krauss explored the effectiveness of LSTM networks in predicting stock prices. They compared LSTMs with traditional models such as logistic regression and support vector machines (SVMs). Their findings indicated that LSTMs outperformed these traditional models in capturing temporal dependencies and achieving higher prediction accuracy.
- Y. Bao, T. Xiong, and Z. Hu (2017): Their study, "A Deep Learning Framework for Financial Time Series Using LSTM Networks," examined the performance of LSTMs in stock price prediction. They utilized historical stock prices and technical indicators as inputs and demonstrated that LSTMs provided better predictive performance compared to ARIMA and GARCH models.
- M. A. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman (2017): In their research "Stock Price Prediction Using LSTM, RNN and CNN-Sentiment Analysis," the authors focused on real-time stock price prediction. They compared the performance of LSTM with recurrent neural networks (RNNs) and CNNs, finding that LSTM models were particularly effective in handling real-time data due to their ability to retain long-term dependencies.
- X. Ding, Y. Zhang, T. Liu, and J. Duan (2015): Their paper "Deep Learning for Event-Driven Stock Prediction" proposed a model using LSTM to predict stock prices based on real-time financial news and events. The results showed that event-driven LSTM models could significantly improve prediction accuracy in high-frequency trading scenarios.

CHAPTER3: METHODOLOGY

3.1 LONG SHORT-TERM MEMORY(LSTM):

LSTM were invented by Hochreiter and Schmidhuber in 1997 [7]. They can capture context specific temporal dependencies for long periods of time [8]. **LSTM** are aimed for a better performance by tackling the vanishing gradient issue that recurrent networks would suffer when dealing with long sequence of data. **LSTMs** are explicitly designed to avoid the long- term dependency problem. Remembering information for long periods of time is practically their default behavior.

An **LSTM** architecture consists of memory cells that can maintain its own state. **LSTM** networks work extremely well with sequence data and long-term dependencies due to their powerful learning capabilities and memory mechanisms.

The repeating module in an **LSTM** contains four interacting layers. An **LSTM** memory cell, has the following three components.

3.1.1 Three Main Components of LSTM:

- 1. Forget Gate:** The forget gate decides when specific portions of the cell state are to be replaced with more recent information. It outputs values close to 1 for parts of the cell state that should be retained, and zero for values that should be neglected.
- 2. Input gate:** Based on the input i.e. previous output $o(t-1)$, input $x(t)$ and previous cell state $c(t-1)$, this section of the network learns the conditions under which any information should be stored or updated in the cell state.
- 3. Output Gate:** Depending on the input and cell state, this portion decides what information is propagated forward and cell state to the next node in the network [9].

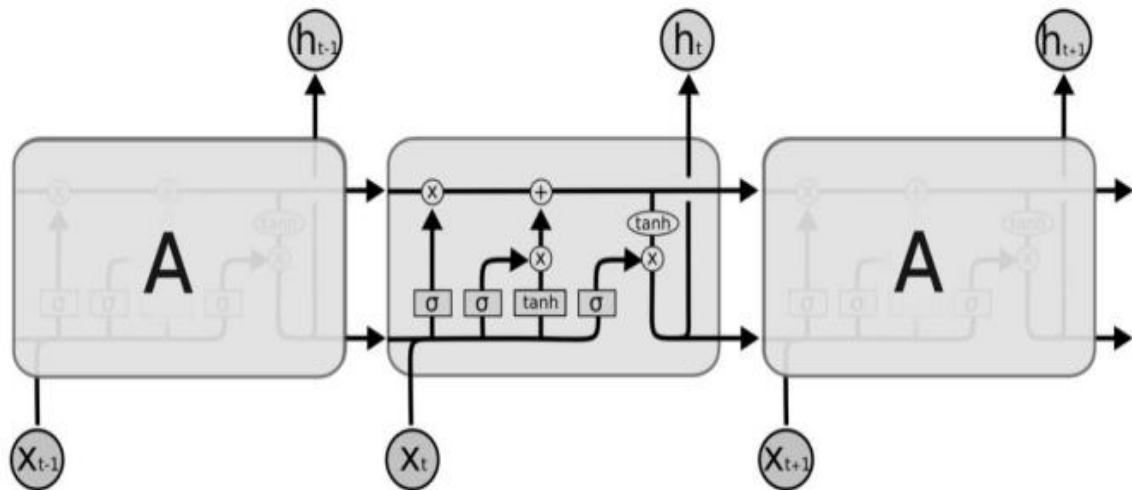


Figure: The repeating module in an LSTM contains four interacting layers

By introducing gates, they were able to improve memory capacity and control the memory cell. One gate is dedicated for reading out the entries from the cell, the output gate. Another gate is needed to decide when data should be read into the cell, this is called the input gate. Finally, a forget gate which resets the content of the cell. This design was used in order to decide when to remember and ignore inputs at the hidden state.[18]

3.2 WORKING OF LSTM

The role of forget gate begins the working of LSTM network. It decides what information is going to be thrown away from the cell state. The sigmoid layer helps in the decision process. The inputs to forget gate are h_{t-1} and x_t . The output from forget gate is a number between 0 and 1 for each number in the cell state C_{t-1} . The output 1 means to store the information while output 0 means to forget the information.

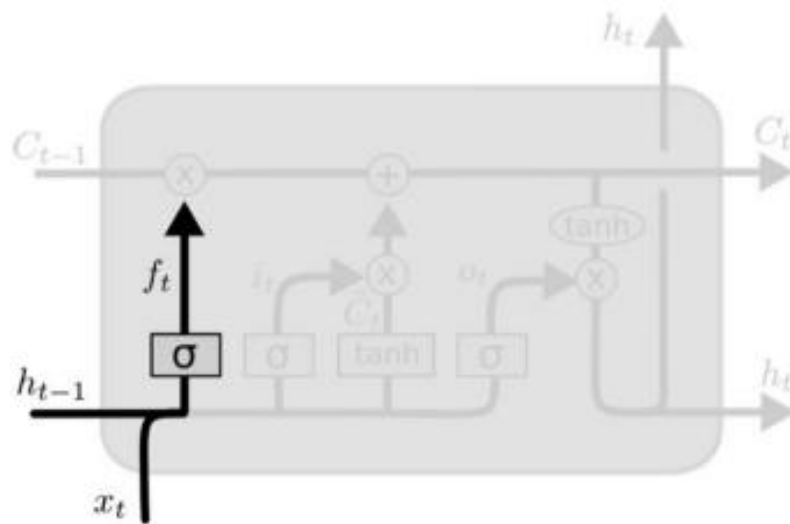


Figure: Forget gate in LSTM

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

In second step of working of LSTM, a decision is made about what new information are going to be stored in the cell state. Decision process is carried with the help of two layers input gate layer and tanh layer. Input gate layer is a sigmoid layer that decides which values will be updated. Tanh layer creates a vector of new candidate values $\tilde{C}_t$ that could be added to the state.

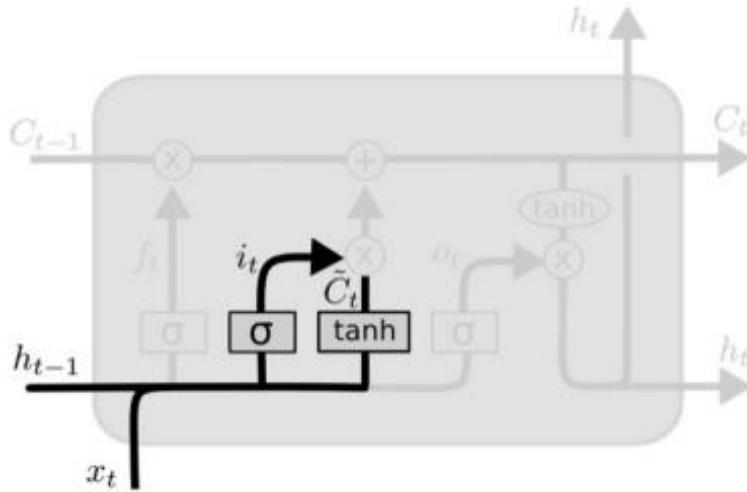


Figure: Decision made for Input in Input Gate

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

$$\tilde{C} = \tanh(W_c.[h_{t-1}, x_t] + b_c)$$

The old cell state C_{t-1} is updated into new cell state $\tilde{C}_t$. Previous two steps have already decided what to do. In this step the implementation is carried out. The old state is multiplied by f_t forgetting the things that is decided to forget in the earlier step. The result is then added with $i_t * \tilde{C}_t$ which gives the new candidate values that is being scaled.

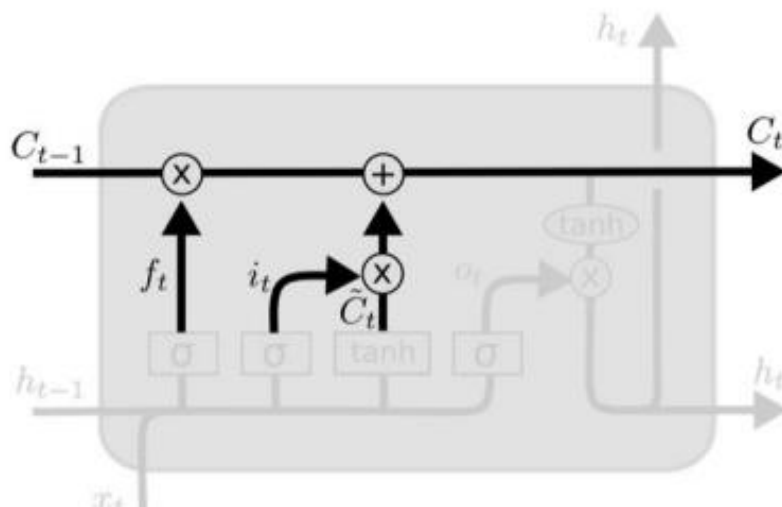


Figure: Implementation of Input gate

$$C_t = (f_t * C_{t-1} + i_t * \tilde{C}_t)$$

This output will be based on our cell state, but will be a filtered version. 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 (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 [10].

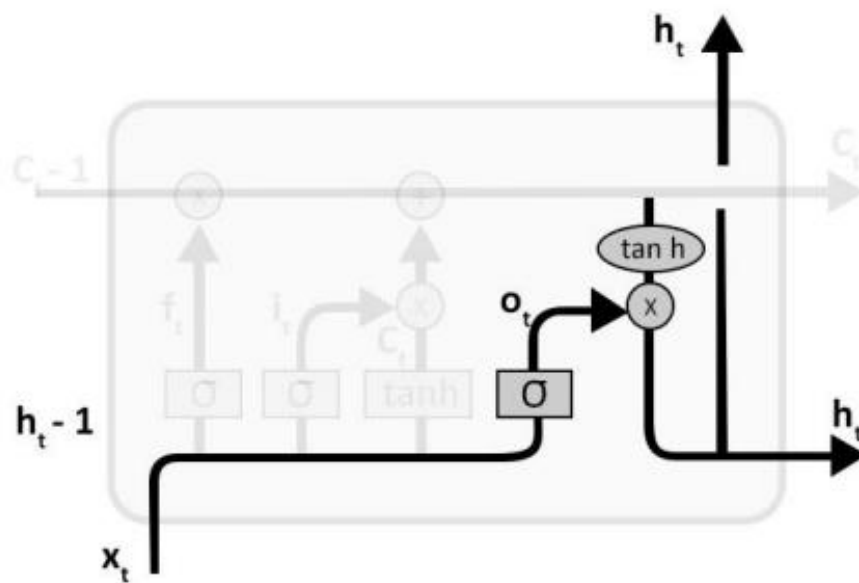


Figure: Output Gate

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

$$h_t = o_t * \tanh(c_t)$$

CHAPTER 4: EXPERIMENTAL RESULTS

4.1 IMPLEMENTATION

LSTM Model Details LSTMs in general are capable of coping with historical data, hence they are really good candidates for stock prediction. LSTMs can learn the order dependence between items in a sequence and are known for their good performance on sequence data-sets. For the purposes of selecting the best combination of features, a dropout based. LSTM model (DrLSTM) with four LSTM layers and 1st 60 units, 2nd with 60 units, 3rd with 80 units, 4th with 120 units for each hidden layer is trained and evaluated. Each hidden LSTM layer has a subsequent dropout layer and finally a dense layer is used to connect all the neurons followed by the last dropout. Dropout is a technique which selects neurons that will be ignored during training, this means that their contribution to the activation of downstream neurons is temporarily removed. The DrLSTM is trained with windows of 60 previous days predicting the next day. where X are the input arrays for the 60 days of data and y are the predicted prices per day, the outcomes of the model for each array X. Then trying to make predictions with the test set of 1-year and checking how close the prediction to the actual data, plotting the graph of the actual data vs the predicted price of the stock price.

4.1.1 TECHNIQUE METHODS USED

PYTHON PACKAGES USED:

- **PANDAS:**

The Pandas library is generally used for data science, but have you wondered why? This is because the Pandas library is used in conjunction with other libraries that are used for data science.

It is built on top of the NumPy library which means that a lot of the structures of NumPy are used or replicated in Pandas.

The data produced by Pandas is often used as input for plotting functions in Matplotlib, statistical analysis in SciPy, and machine learning algorithms in Scikit-learn.

Here is a list of things that we can do using Pandas. Data set cleaning, merging, and joining.

Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data.

Columns can be inserted and deleted from DataFrame and higher-dimensional objects.

Powerful group by functionality for performing split-apply-combine operations on data sets. Data Visualization.[11]

- **WHY TENSORFLOW?**

TensorFlow :

TensorFlow is an open-source platform for machine learning and a symbolic math library that is used for machine learning applications.

Advantages of TensorFlow:

Tensor flow has a better graph representation for a given data rather than any other top platform out there.

Tensor flow has the advantage that it does support and uses many backend software like GUI and ASIC.

When it comes to community support tensor flow has the best.

Tensor flow also helps in debugging the sub-part of the graphs.

Tensor flow has shown a better performance when compared with other platforms.

Easy to extend as it gives freedom to add custom blocks to build on new ideas.[10]

- **KERAS:**

It is an Open-Source Neural Network library that runs on top of Theano or Tensorflow. It is designed to be fast and easy for the user to use. It is a useful library to construct any deep learning algorithm of whatever choice we want.

Advantages of Keras:

- ❖ Keras is the best platform out there to work on neural network models.
- ❖ The API that Keras has a user-friendly where a beginner can easily understand.
- ❖ Keras has the advantage that it can choose any libraries which support it for its backend support.
- ❖ Keras provides various pre-trained models which help the user in further improving the models the user is designing.

- ❖ When it comes to community support Keras has the best like stack overflow.[11].

- **MATPLOTLIB:**

Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib's primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyze and understand. It was originally developed by John D. Hunter in 2003 and is now maintained by a large community of developers.

Key Features of Matplotlib:

- ❖ Versatility: Matplotlib can generate a wide range of plots, including line plots, scatter plots, bar plots, histograms, pie charts, and more.
- ❖ Customization: It offers extensive customization options to control every aspect of the plot, such as line styles, colors, markers, labels, and annotations.
- ❖ Integration with NumPy: Matplotlib integrates seamlessly with NumPy, making it easy to plot data arrays directly.
- ❖ Publication Quality: Matplotlib produces high-quality plots suitable for publication with fine-grained control over the plot aesthetics.
- ❖ Extensible: Matplotlib is highly extensible, with a large ecosystem of add-on toolkits and extensions like Seaborn, Pandas plotting functions, and Basemap for geographical plotting.
- ❖ Cross-Platform: It is platform-independent and can run on various operating systems, including Windows, macOS, and Linux.
- ❖ Interactive Plots: Matplotlib supports interactive plotting through the use of widgets and event handling, enabling users to explore data dynamically.[12]

4.2 STEPS FOR MODEL IMPLEMENTATION

In this section included is an introduction with all the major steps which will be considered in this research project:

1. Data used for this research will be extracted from Csv files downloaded from finance.yahoo.com [13].
2. Data will be normalized using a python library sklearn. Feature scaling is a method to normalize the range of independent feature variables. Data is then split into a training and a testing set.
3. Data will be used to train the single time scale feature models over many iterations to predict each variable independently. As many traditional predicting models Opening Price will be used as a feature to train the control model.
4. First a model will be trained using the best possible features of the standard data-set, which will be selected by running tests for each combination of features and comparing their losses, and the label will be set to the Open price. A standard dropout LSTM model will be optimized though experimentation of hyperparameters and compared with other variants of LSTM.
5. Evaluation of the methods will be made using mean squared error and checking the actual price vs the predicted value and plotting a graph to check how close we are to the actual stock price.



Figure: Depicting the sequence of the process that is followed for each model.

4.3 DATASET DESCRIPTION:

- **4.3.1 Raw Data:** The data used in this article is from Yahoo Finance and includes 7 features: Date, Open Price, High Price, Low Price, Close Price, Adjusted Closing Price and Volume. The data structure is shown in Table .
- ❖ Our data series cover the period going from 23/5/2002 to 11/11/2021.

TABLE: Stock Price of Netflix from 23/5/2002 to 11/11/2021

	Date	High	Low	Open	Close	Volume	Adj Close
0	23-05-2002	1.242857	1.145714	1.156429	1.196429	104790000	1.196429
1	24-05-2002	1.225000	1.197143	1.214286	1.210000	11104800	1.210000
2	28-05-2002	1.232143	1.157143	1.213571	1.157143	6609400	1.157143
3	29-05-2002	1.164286	1.085714	1.164286	1.103571	6757800	1.103571
4	30-05-2002	1.107857	1.071429	1.107857	1.071429	10154200	1.071429
...	...	...	...	...	...	...	...
4899	05-11-2021	663.969971	665.640015	645.010010	645.719971	5283500	645.719971
4900	08-11-2021	650.289978	656.000000	643.789978	651.450012	2887500	651.450012
4901	09-11-2021	653.700012	660.500000	650.520020	655.989990	2415600	655.989990
4902	10-11-2021	653.010010	660.330017	642.109985	646.909973	2405800	646.909973
4903	11-11-2021	650.239990	665.820007	649.710022	657.580017	2868300	657.580017

TABLE: Stock Price of Tesla from 28/5/2019 to 19/05/2023

	Date	Open	High	Low	Close	Adj Close	Volume
0	2019-05-28	12.746667	13.000000	12.523333	12.580000	12.580000	154693500
1	2019-05-29	12.473333	12.826000	12.336000	12.657333	12.657333	179529000
2	2019-05-30	12.583333	12.817333	12.468000	12.548000	12.548000	118897500
3	2019-05-31	12.340000	12.661333	12.273333	12.344000	12.344000	156100500
4	2019-06-03	12.367333	12.445333	11.799333	11.931333	11.931333	195966000
...	...	...	...	...	...	...	...
1002	2023-05-18	174.220001	177.059998	172.449997	176.889999	176.889999	109520300
1003	2023-05-19	177.169998	181.949997	176.309998	180.139999	180.139999	136024200
1004	2023-05-22	180.699997	189.320007	180.110001	188.869995	188.869995	132001400
1005	2023-05-23	186.199997	192.960007	185.259995	185.770004	185.770004	156952100
1006	2023-05-24	182.229996	184.220001	178.220001	182.899994	182.899994	137605100

1007 rows × 7 columns

4.3.2 DATA USED FOR TESTING

Table: Test dataset for Netflix

	Date	Open	High	Low	Close	Adj Close	Volume
0	2021-11-12	660.010010	683.340027	653.820007	682.609985	682.609985	4198400
1	2021-11-15	681.239990	685.260010	671.489990	679.330017	679.330017	2872200
2	2021-11-16	678.270020	688.359985	676.900024	687.400024	687.400024	2077400
3	2021-11-17	690.000000	700.989990	686.090027	691.690002	691.690002	2732800
4	2021-11-18	691.609985	691.739990	679.739990	682.020020	682.020020	2012900
...	...	...	...	...	...	...	...
247	2022-11-07	261.059998	261.149994	252.089996	258.600006	258.600006	7927000
248	2022-11-08	259.920013	266.079987	256.880005	263.459991	263.459991	7683800
249	2022-11-09	259.660004	260.899994	254.220001	254.660004	254.660004	7266000
250	2022-11-10	265.989990	275.359985	260.089996	274.970001	274.970001	9667300
251	2022-11-11	274.470001	290.660004	271.559998	290.130005	290.130005	9573800

Table: Test dataset for Tesla

	Date	Open	High	Low	Close	Adj Close	Volume
0	25-05-2023	186.539993	186.779999	180.580002	184.470001	184.470001	96870700
1	26-05-2023	184.619995	198.600006	184.529999	193.169998	193.169998	162061500
2	30-05-2023	200.100006	204.479996	197.529999	201.160004	201.160004	128818700
3	31-05-2023	199.779999	203.949997	195.119995	203.929993	203.929993	150711700
4	01-06-2023	202.589996	209.800003	199.369995	207.520004	207.520004	148029900
...	...	...	...	...	...	...	...
247	20-05-2024	177.559998	177.750000	173.520004	174.949997	174.949997	61727400
248	21-05-2024	175.509995	186.880005	174.710007	186.600006	186.600006	115266500
249	22-05-2024	182.850006	183.800003	178.119995	180.110001	180.110001	88313500
250	23-05-2024	181.800003	181.899994	173.259995	173.740005	173.740005	71975500
251	24-05-2024	174.839996	180.080002	173.729996	179.240005	179.240005	65479700

4.4 Visualization of Feature Data

4.4.1 Data Pre-processing: The pre-processing stage involves:

- a) Data discretization: Part of data reduction but with particular importance, especially for numerical data
- b) Data transformation: Normalization.
- c) Data cleaning: Fill in missing values.

After the dataset is transformed into a clean dataset, the dataset is divided into training, testing. In order to build our model, we are going to use the LSTM RNN.

4.4.2 Feature Extraction:

In this stage, only the features which are to be fed to the neural network are chosen. We will choose the feature from Date, Open, High, Low, Close, Adj_close and Volume.

- In this experiment the **‘Open’** price of the stock as a single feature.

Plot a Graph:

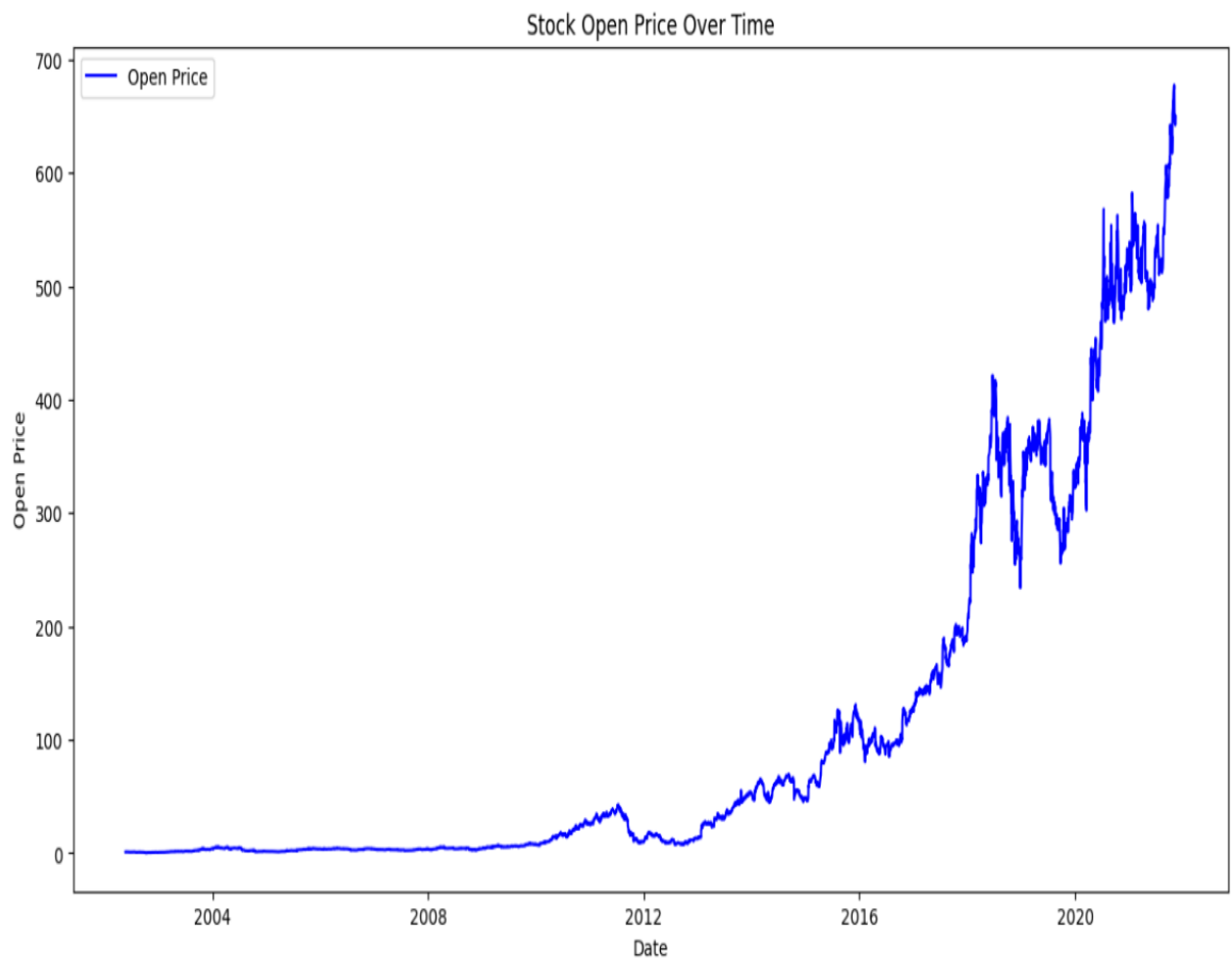


Figure: **OPEN PRICE VS DATE** of Netflix Stock Price

Plot a Graph:



Figure: **OPEN PRICE VS DATE** of Tesla Stock Price

4.5 MODEL PERFORMANCE

4.5.1 MODEL SUMMARY

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 60)	14,880
dropout (Dropout)	(None, 60, 60)	0
lstm_1 (LSTM)	(None, 60, 60)	29,040
dropout_1 (Dropout)	(None, 60, 60)	0
lstm_2 (LSTM)	(None, 60, 80)	45,120
dropout_2 (Dropout)	(None, 60, 80)	0
lstm_3 (LSTM)	(None, 120)	96,480
dropout_3 (Dropout)	(None, 120)	0
dense (Dense)	(None, 1)	121

Total params: 185,641 (725.16 KB)

Trainable params: 185,641 (725.16 KB)

Non-trainable params: 0 (0.00 B)

4.5.2 Loss Function vs Epoch

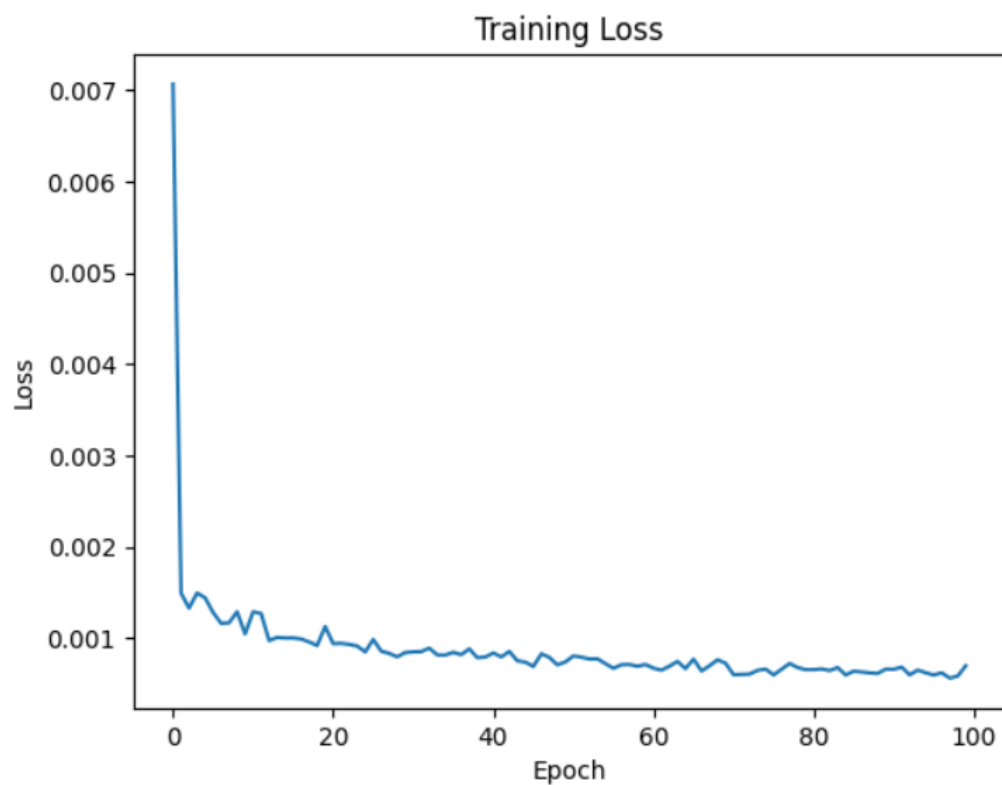


Figure: Netflix training loss vs Epoch graph

- loss Function vs Epoch:

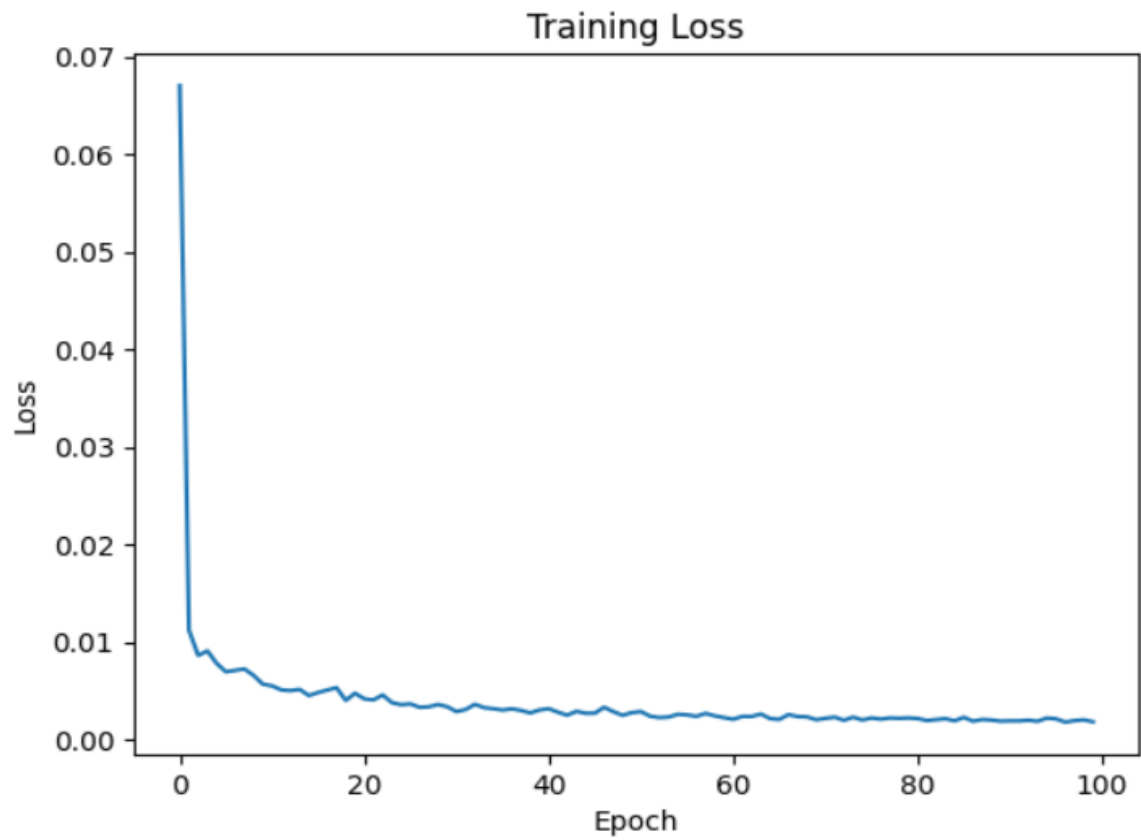


Figure: Tesla training loss vs Epoch graph

4.6 MEAN SQUARED ERROR(MSE)

- MSE values represent the average squared difference between the predicted and actual stock prices.
- Lower MSE values indicate better predictive performance, as they suggest that the model's predictions are closer to the actual values.
- Conversely, higher MSE values indicate larger errors between the predicted and actual values, suggesting poorer predictive performance.[15]

4.6.1 Formula of MSE

$$\text{MSE} = \overset{\text{Mean}}{\frac{1}{n} \sum_{i=1}^n} \overset{\text{Error}}{(Y_i - \hat{Y}_i)} \overset{\text{Squared}}{^2}$$

real value predicted value

$$\boxed{Y_i} - \boxed{\hat{Y}_i}$$

error

number of samples

$$\frac{1}{\boxed{n}} \sum_{i=1}^{\boxed{n}}$$

4.6.2 MSE OF THE EXPERIMENT

MSE VS Prediction:

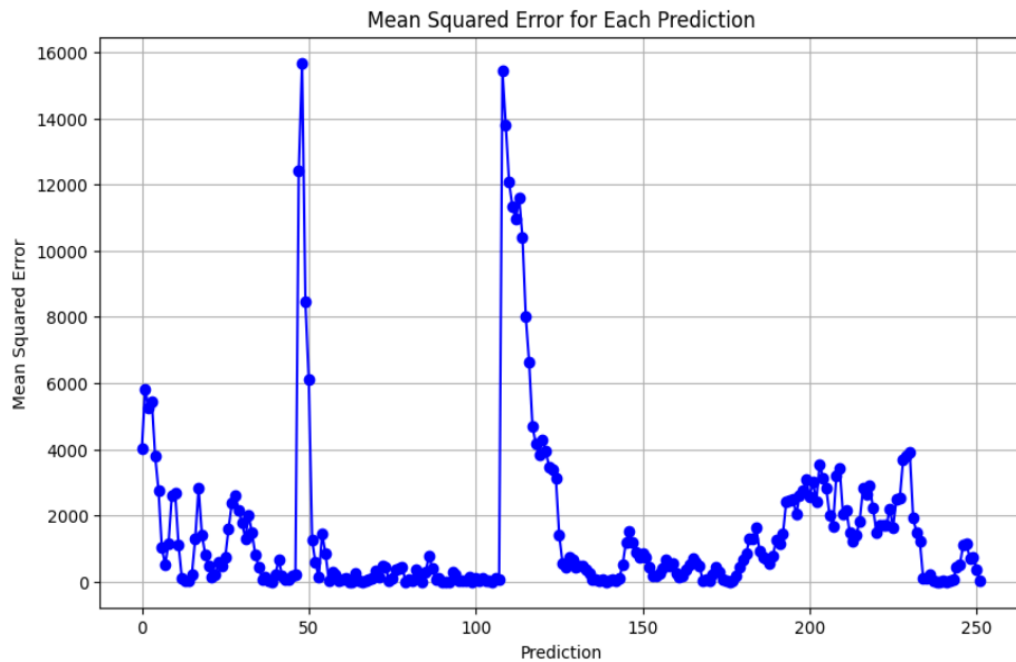


Figure: Netflix Mean Squared Error vs Prediction graph

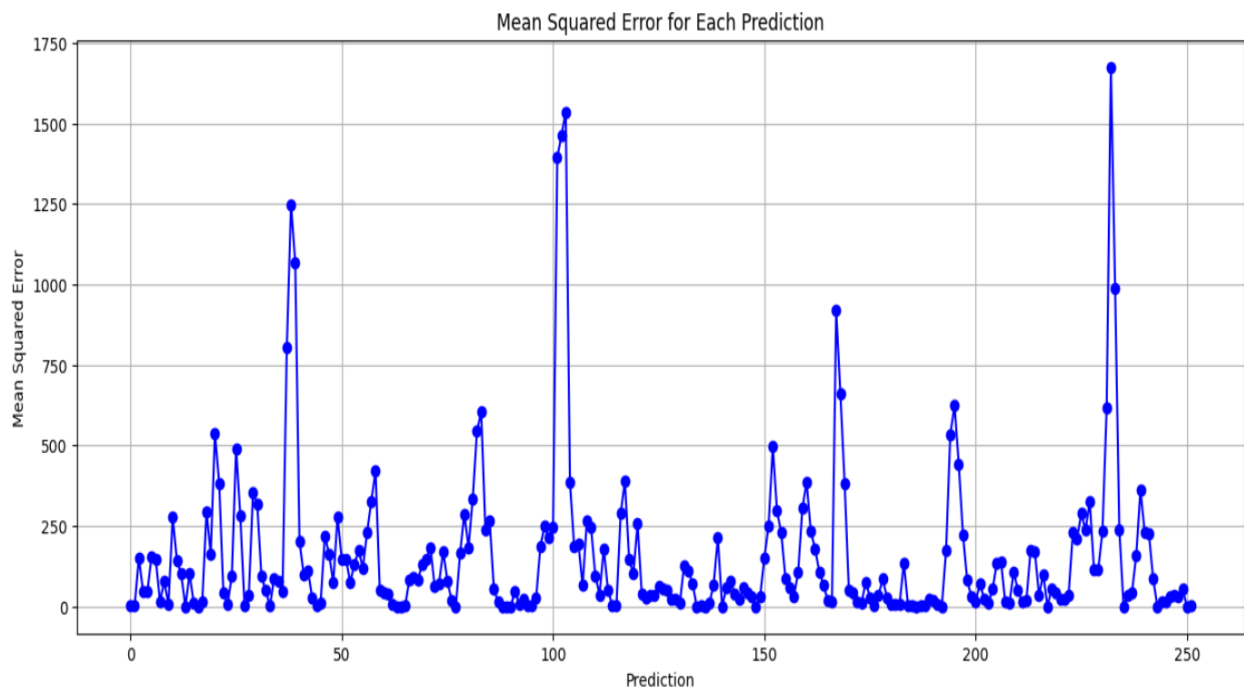


Figure: Tesla Mean Squared Error vs Prediction graph

4.7 Mean Absolute Percentage Error(MAPE)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|A_i - F_i|}{A_i}$$

A_i is the actual value

F_i is the forecast value

n is total number of observations

[16]

4.8 MODEL ACCURACY:

$$\text{Accuracy percentage} = 100 - (\text{MAPE} * 100)$$

Table: Accuracy rate for Tesla, Netflix

Dataset	Accuracy(in%)
Netflix	88.24%
Tesla	95.50%

4.9 RESULTS

->For Tesla,

REAL PRICE VS PREDICTED PRICE WITH RESPECT TO DATE:

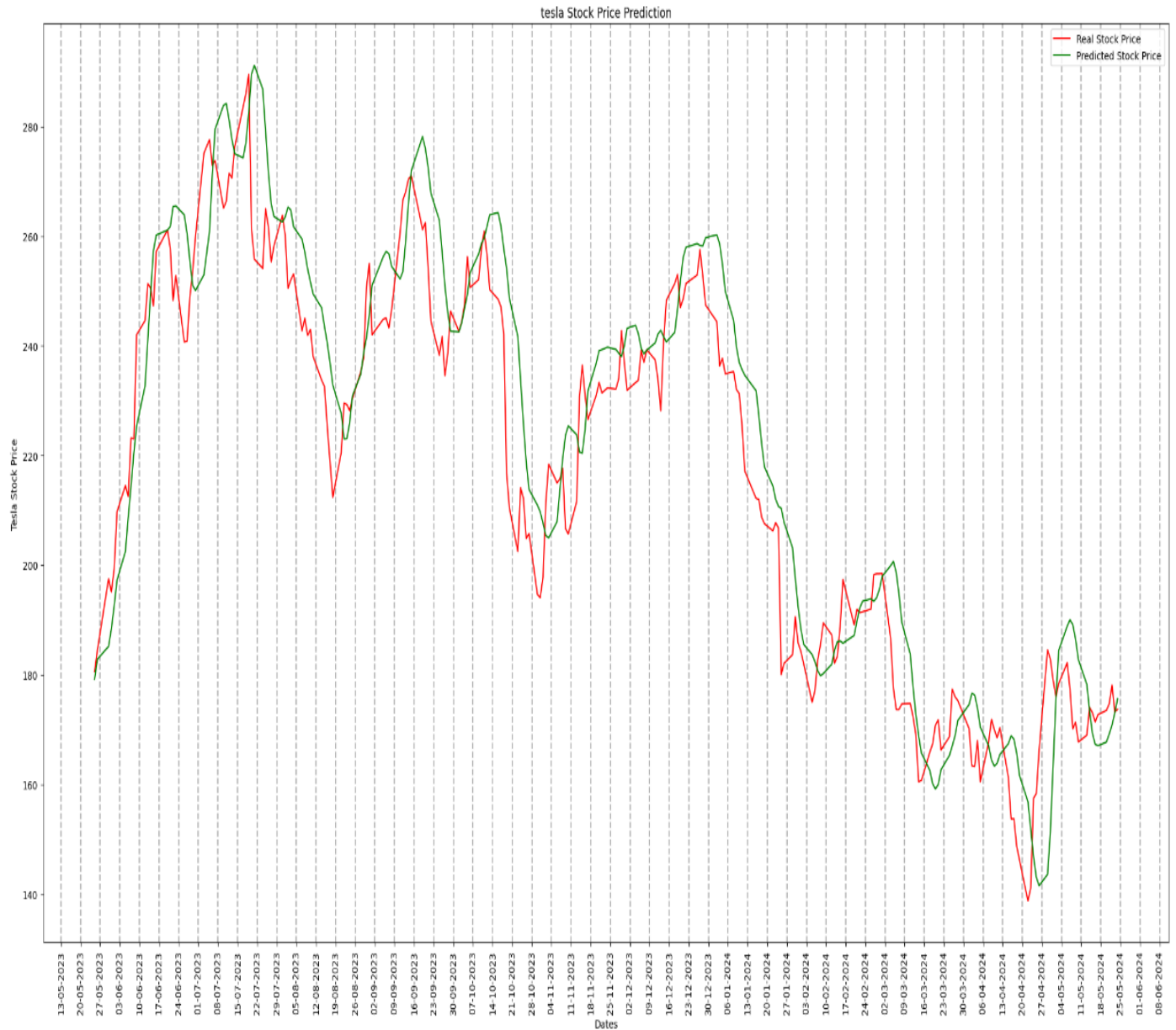


Figure: Tesla Stock Price Prediction graph (Real vs Predicted)

-> For Netflix,

REAL PRICE VS PREDICTED PRICE WITH RESPECT TO DATE:

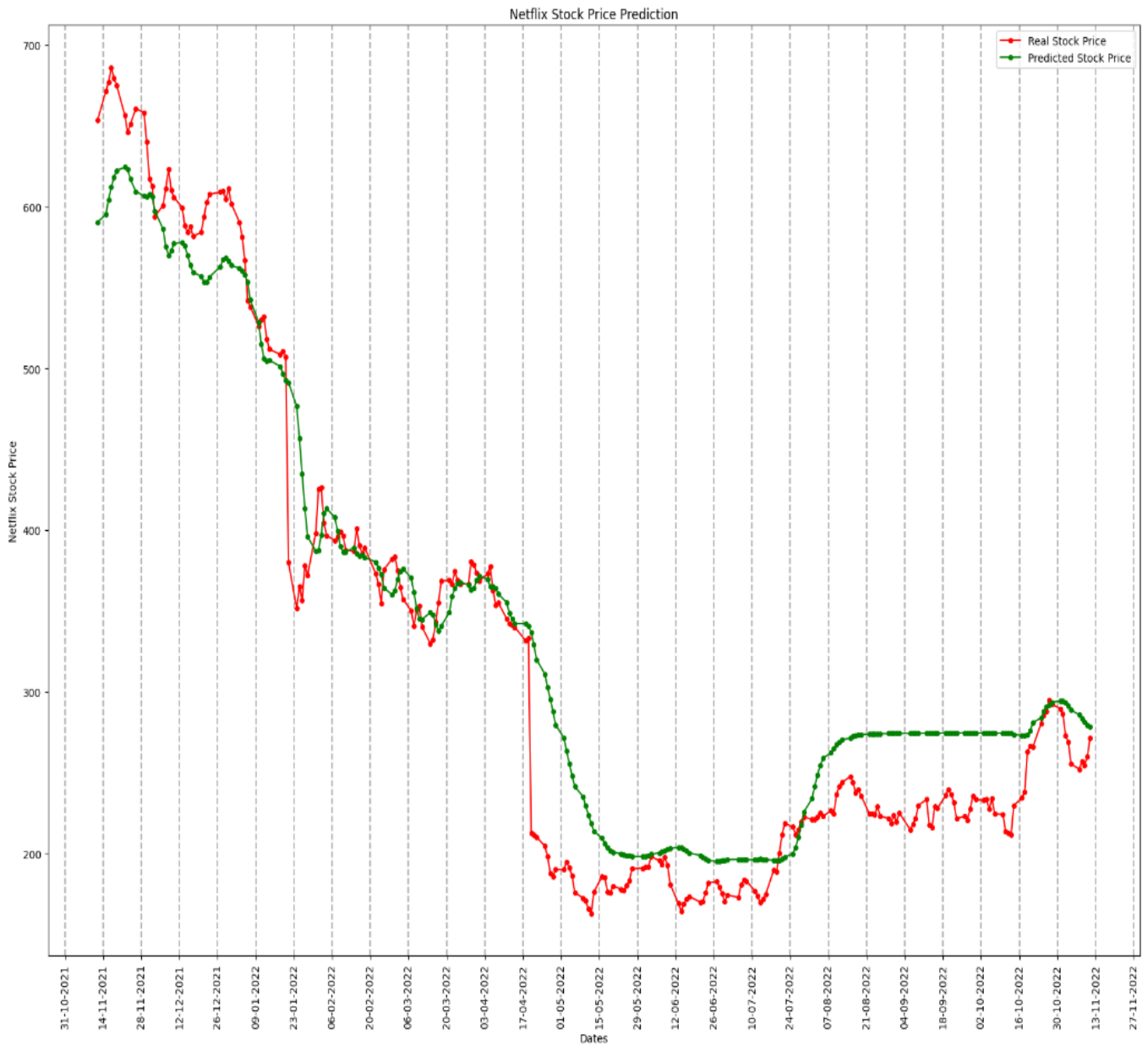


Figure: Netflix Stock Price Prediction graph (Real vs Predicted)

CHAPTER5: CONCLUSION

5.1 LIMITATIONS

Trading in the stock market is growing rapidly and investors, analysts are eager to find a method and technique to effectively predict future stock market trends. In recent years, many studies have shown that LSTM neural network models are effective in predicting the stock market, and compared with other machine learning algorithms, LSTM neural network models perform very well when applied to longer prediction horizons. In this paper, we compare the results of the LSTM neural network model in making short-term forecast range predictions with the results of an algorithm consisting of four layers and dropout layers and find that the LSTM neural network model is quite close to predicting actual prices. However, it has to be admitted that the LSTM neural network model is better at capturing trends and seasonality in long-term forecast range prediction. This will encourage more researchers to use new techniques to find new forecasting methods that can be applied to more situations, thus helping investors, analysts or anyone interested in investing in the stock market by providing them with a good knowledge of the future of the stock market.

5.2 FUTURE SCOPE

5.2.1 Expanding the Dataset:

- ❖ Extended Time Frames: Use a more extended period of historical data to train the model.
- ❖ Incorporating Multiple Stocks: Train the model on data from multiple stocks to improve generalizability and robustness.
- ❖ Alternative Data Sources: Include alternative data sources like social media sentiment, economic indicators, and geopolitical events.

5.2.2 Improving Model Performance:

- ❖ Feature Engineering: Introduce additional features such as trading volume, moving averages, technical indicators, sentiment analysis of news articles, and other relevant financial metrics.
- ❖ Hyperparameter Tuning: Optimize the model's hyperparameters (e.g., number of layers, number of units in each layer, learning rate, batch size) using techniques such as grid search or random search.
- ❖ Model Ensemble: Combine predictions from multiple models (e.g., different LSTM configurations, other neural networks, or traditional machine learning models) to improve prediction accuracy.

REFERENCES

- [1] C. Giles, S. Lawrence and A. C. Tsoi, "Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference," Machine Learning, pp. 161-183, July/August 2001.
- [2] F. E. Tay and L. Cao, "Application of support vector machines in financial time series forecasting," Omega, pp. 309-317, August 2001.
- [3] Y. S. Bu-Mostafa and A. F. Atiya, "Introduction to financial forecasting", Appl. Intell., vol. 6, no. 3, pp. 205-213, 1996
- [4] Roondiwala, M., Patel, H. and Varma, S., 2017."Predicting stock prices using LSTM" International Journal of Science and Research (IJSR), 6(4),pp.1754-1756.
- [5] PENG Yan, LIU Yuhong, ZHANG Rongfen, 2019. "Modeling and analysis of stock price forecast based on LSTM". Computer Engineering and Applications, 55 (11), pp.209-212.
- [6] Deng, Fengxin and Wang, Hongliang, 2018. "Application of LSTM neural networks in stock price trend prediction - a study based on individual stock data from the US and Hong Kong stock markets". Collection, 14.
- [7] J. Chawalit, C. Rujira and T. Laksmeay, "Stock Price Prediction With Long Short Term Memory Recurrent Neural Network," IEEE.
- [8] R. Nandakumar, U. K. R and Y. V. Lokeshwari, "Stock Price Prediction Using Long Short Term Memory," International Research Journal of Engineering and Technology (IRJET), Chennai, Tamil, 2018.
- [9] C. Olah, "Understanding LSTM Networks," 15 08 2015. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed 29 08 2019].

[10]<https://www.geeksforgeeks.org/introduction-to-pandas-in-python>

[11]<https://www.geeksforgeeks.org/difference-between-tensorflow-and-keras>

[12]<https://www.geeksforgeeks.org/python-introduction-matplotlib/>

[13] <https://finance.yahoo.com/quote/NFLX/history>

[14] Ma, Q., 2020, "Comparison of ARIMA, ANN and LSTM for stock price prediction". In E3S Web of Conferences (Vol. 218). EDP Sciences.

[15] Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A. and Salwana, E., 2020, "Deep learning for stock market prediction". Entropy, 22(8), p.840.

[16] Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. , "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques". Expert Syst. Appl. 2015, 42, 259–268.

[17] Ruobing Zhang, "LSTM-based Stock Prediction Modeling and Analysis". Advances in Economics, Business and Management Research, volume 211

[18] S. Dinesh , A.M.S. Rama Raju, S. Rahul, O. Naga Sandeep, Mr. N D S S Kiran Relangi" STOCK PRICE PREDICTION USING LSTM". Mukta Shabd Journal, Volume X, Issue VI, june/2021