VIDYASAGAR UNIVERSITY

A PROJECT REPORT

ON

STOCK MARKET PREDICTION USING DEEP LEARNING

Dissertation submitted to the Department of Computer Science
for the partial fulfilment of the requirements for the award of the degree of
Master of Science (M.Sc.) in **Computer Science**

Submitted By

SAYAN GHOSH

Roll No: VU/PG/22/26/02-IVS-0018

Reg. No: 1361941 of 2019-2020

Supervised By

Mr. Bachchu Paul

Assistant Professor

Department of Computer Science

Vidyasagar University

DEPARTMENT OF COMPUTER SCIENCE
M.Sc. 4th Semester
Vidyasagar University, Midnapore, West Bengal-721102
2024

VIDYASAGAR UNIVERSITY

A PROJECT REPORT

ON

STOCK MARKET PREDICTION USING DEEP LEARNING

Dissertation submitted to the Department of Computer Science
for the partial fulfilment of the requirements for the award of the degree of
Master of Science (M.Sc.) in **Computer Science**

By

SAYAN GHOSH
Roll No: VU/PG/22/26/02-IVS-0018
Reg. No: 1361941 of 2019-2020

Under the Supervision of

Mr. Bachchu Paul

DEPARTMENT OF COMPUTER SCIENCE
M.Sc. 4th Semester
Vidyasagar University, Midnapore, West Bengal-721102

2024

CERTIFICATE

This is to certify that the work contained in the project entitled **"Stock Market Prediction Using Deep Learning"** submitted by **Sayan Ghosh (Roll No: VU/PG/22/26/02-IVS-0018, Registration No: 1361941 of 2019-2020)** for the award of the degree of **Master of Science (M.Sc.)** in **Computer Science** of the **Vidyasagar University**, is a record of bona fide project carried out by them under my direct supervision and guidance.

I have thoroughly reviewed and evaluated the project and find that it meets the required standards and fulfills the necessary criteria as per the rules and regulations pertaining to the nature of the degree. The contents embodied in the dissertation have not been submitted for the award of any other degree or diploma in this or any other university.

Signature of HOD	**Signature of Supervisor**
Prof. Biswapati Jana	**Mr. Bachchu Paul**
Head of the Department	**Assistant Professor**

DECLARATION

I certify that

1. The work contained in the dissertation is original and has been done by myself under the supervision of my supervisor.
2. The work has not been submitted to any other Institute for any degree or diploma.
3. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the dissertation and providing their details in the references.
4. I take full responsibility for the content, data, and results presented in this project. I understand that any misrepresentation, plagiarism, or academic dishonesty in this project will have severe consequences, including the rejection of my submission and potential disciplinary action.
5. I hereby grant Vidyasagar University the right to retain and make copies of this project for academic and archival purposes. However, the copyright of this project remains with me, and I reserve the right to publish or disseminate this work, in whole or in part, with proper attribution.

Date:

Place:

Sayan Ghosh

Roll No: VU/PG/22/26/02-IVS-0018

Registration No: 1361941 of 2019-2020

ACKNOWLEDGEMENT

I would like to express our heartfelt gratitude to all those who have contributed to the successful completion of this project entitled **"Stock Market Prediction Using Deep Learning".** Their support, guidance, and encouragement have been instrumental throughout this journey.

First and foremost, I would like to thank our supervisor, **Mr. Bachchu Paul**, for his invaluable guidance, expertise, and continuous support. His insightful feedback, patience, and encouragement have been crucial in shaping this project and expanding our understanding of the subject matter.

I would like to extend our appreciation to the faculty members of the Department of Computer Science at Vidyasagar University for providing us with a conducive academic environment.

Lastly, I would like to thank our family, friends and classmates for their unwavering support and encouragement throughout this project. Their support and motivation have made this journey more enjoyable and fulfilling for us.

In conclusion, I am sincerely grateful to all individuals and organizations who have directly or indirectly contributed to the successful completion of this project. Their assistance and support have been invaluable, and I am proud to have been able to complete this project with their help.

Date:

Place:

Sayan Ghosh

Roll No: VU/PG/22/26/02-IVS-0018

Registration No: 1361941 of 2019-2020

ABSTRACT

The stock market is a complex, dynamic system influenced by numerous factors, making accurate prediction a challenging task. Predicting stock market movements is one of the most difficult tasks in computational finance due to the interplay of statistical analysis, historical data, and external circumstances. Factors such as physical and psychological influences, investor sentiment, market rumors, and rational or irrational behavior contribute to stock price volatility, complicating accurate predictions. Stock value prediction demands a robust computational foundation, particularly for longer-term forecasts. This project employs deep learning to navigate these complexities and enhance prediction accuracy, aiming to maximize profitability by leveraging advanced computational techniques.

This project aims to develop a stock market prediction model using LSTM networks, a specialized form of Recurrent Neural Networks (RNNs), to predict stock market trends. The model will be trained on historical stock market data, including various features such as stock prices, trading volumes, and other relevant indicators. LSTM networks are well-suited for capturing long-term dependencies in sequential data, making them ideal for predicting stock prices.

We focus on the prediction of Tesla's stock prices, given their high volatility and investor interest. The project involves collecting historical stock data, preprocessing it for LSTM compatibility, and designing an LSTM network to forecast future stock prices. The model is trained on features such as opening price, closing price, highest price, lowest price, and trading volume, with the aim to minimize prediction error and maximize the model's generalization capabilities.

The LSTM model's performance is evaluated using metrics like Mean Absolute Error (MAE), R2 Score and Mean Squared Error (MSE), and the results are compared against traditional time-series forecasting methods. The project's findings suggest that deep learning, particularly LSTM networks, can be a valuable tool in predicting stock market trends, offering insights that could potentially benefit investors and traders.

Keywords: Stock market analysis; Stock price prediction; long short-term memory (LSTM); Recurrent Neural Network (RNN); machine learning; financial analysis; time series data.

Contents

CHAPTER 1
INTRODUCTION

1.1 Overview

The equity capital markets function as a platform for issuing and trading shares of listed companies. Stocks, or shares, address fragmentary possession in an organization, resource, or security, and in this manner, the stock market is a platform for financial backers where they can purchase and sell responsibility for investable resources or offers [1]. The share market is a snapshot of future growth expectations of companies as well as the economy. Many factors have attributed to stock price fluctuation, which includes but is not limited to macroeconomic factors, the market anticipation, and confidence in the company's management and operation. The advancement of technology allows the public to access a larger quantity of information in a timelier manner. This means that stock analysis has become more and more difficult as a considerable amount of data has to be processed in a relatively short time. People hope that the progress made in big data, especially in the deep learning field, can help them analyze stock information [2].

1.2 Stock Market Prediction

Stock Market is one of the largest financial market in the world. Stock market or equity market have a profound impact in today's economy. A rise or fall in the share price has an important role in determining the investor's gain [3]. The prediction of the exchange rates can provide investors with useful decision-making references to increase return and to reduce risk. However, the exchange rate is always under the influence of many factors, such as countries' economies, politics, society, international situation, etc. so the complexity of the matter has made stock prediction and forecasting a challenging topic [4]. Stock market is characterized as dynamic, unpredictable and non-linear in nature. Predicting stock prices is a challenging task as it depends on various factors including but not limited to political conditions, global economy, company's financial reports and performance etc. Thus, to maximize the profit and minimize the losses, techniques to predict values of the stock in advance by analyzing the trend over the last few years, could prove to be highly useful for making stock market movements

[5]. Traditionally, two main approaches have been proposed for predicting the stock price of an organization. Technical analysis method uses historical price of stocks like closing and opening price, volume traded, adjacent close values etc. of the stock for predicting the future price of the stock. The second type of analysis is qualitative, which is performed on the basis of external factors like company profile, market situation, political and economic factors, textual information in the form of financial new articles, social media and even blogs by economic analyst [6].

Financial researchers around the world have been studying and analyzing the changes in the stock and stock markets. The broadening application of artificial intelligence has led to an increasing number of investors using deep learning model to predict and study stock and Forex prices. It has been proven that the fluctuation in stock and stock price could be predicted [7]. The attempt that is made to forecast or predict the upcoming value of the stock, sector of the market or even the entire market is known as **Stock Market Prediction**. It is an area that has driven the focus of many individuals including not only companies, but also traders, market participants, data analysts, and even computer engineers working in the domain of Machine Learning (ML) and Artificial Intelligence (AI), etc. Investing funds in the market is subjected to various market risks, as the value of the shares of the company is highly dependent upon the profits and performance of the organization in the marketplace and can thus vary due to various factors such as government policies, microeconomic indicators, demand and supply etc. These variations in the market are studied to develop software and programs using various techniques such as Machine Learning, Deep Learning, Neural Networks, Artificial Intelligence, etc. Such systems and software can enable the investor to properly anticipate the situation of the company, on the basis of past and present data, the current condition in the market, etc. and give them a direction to make decisions so that they don't lose their valuable money and earn maximum profits [8].

Nowadays, advanced intelligent techniques based on either technical or fundamental analysis are used for predicting stock prices. Particularly, for stock market analysis, the data size is huge and also non-linear. To deal with this variety of data efficient model is needed that can identify the hidden patterns and complex relations in this large data set. Machine learning techniques in this area have proved to improve efficiencies by 60-86 percent as compared to the past methods [9]. Most of the previous work in this area use classical algorithms like linear regression [10], Random Walk Theory (RWT) [11], Moving Average Convergence / Divergence (MACD) [12] and also using some linear models like Autoregressive Moving

Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) [13], for predicting stock prices. Recent work shows that stock market prediction can be enhanced using machine learning techniques such as Support Vector Machine (SVM), Random Forest (RF) [14]. Some techniques based on neural networks such as Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and deep neural networks like Long Short-Term Memory (LSTM) also have shown promising results [15].

1.3 Aim and Objective

In the project we will develop a stock data predictor program that uses previous stock prices and data will be treated as training sets for the program to predict the stock prices of a particular share this program develops a procedure. This model considers the historical equity share price of a company price and applies RNN (Recurrent Neural Network) technique called **Long Short-Term Memory (LSTM)**. Deep learning models provide good result in many areas and detect the dynamics of stock market movement and get good result and in the final evaluation step we considered all the deep learning algorithm and apply it to get the better prediction and then we will see the performance of the model and analyze how it differ with previous study and get to know that how efficient and productive the algorithm is by checking the mean accuracy of the model. The proposed approach considers available historical data of a share and it provides prediction on a particular feature. The features of shares are date, open, close, high, low, adjusted close, volume. The proposed model uses the time series analysis in order to predict a share price for a required time span [16]. The Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) capable of addressing linear problems. The Long Short-Term Memory (LSTM) units are enforced to learn very long sequences. This is a more general version of the gated recurrent system. LSTM is more benign than other deep learning methods like RNN or traditional feed forward because LSTMs tackle the evanescent gradient issue possessed by [17]. LSTM networks have proven to be highly effective in capturing long-term dependencies in sequential data, making them particularly well-suited for predicting stock prices. The utilization of LSTM networks in this project allows for the exploration of complex patterns and relationships within the stock market data. By learning from historical data, the model will be able to identify and interpret subtle trends, enabling it to make accurate predictions of future stock prices. Another way to forecast equity prices is by analyzing the sentiments on social media data or news stories that help in determining the general trend that a particular company’s or industries’ shares may take based on a collective opinion [18].

In this project, we focus on **Tesla, Inc**., a leading American electric vehicle and clean energy company. Tesla is well-known for its innovative automotive and energy solutions. Its shares are traded on the NASDAQ stock exchange, which offers modern facilities for global investors to trade seamlessly. Tesla plays a significant role in the stock market, enhancing transparency, efficiency, and convergence in capital markets. The company's stock is closely watched by investors worldwide, highlighting its leadership in technology and sustainability. Similar to the NSE in India, NASDAQ handles the exchange, settlement, and clearing of equity, debt, and derivatives trading.

This project encompasses several stages, including data collection, preprocessing, model design, training, evaluation, and interpretation of results. The ultimate goal is to develop a reliable and robust LSTM-based system that can assist in making informed investment decisions, potentially leading to more strategic trading and better financial outcomes.

The rest of the paper is organized as follows. In Section **2,** a literature survey will be conducted to examine the existing research on stock price prediction, theoretical basis of the machine learning and deep learning method and identify gaps in the current knowledge. Section **3** introduces experiment with tools and technology. The methodology in section **4** will detail the data collection process and the machine learning algorithms used for predictive modeling. The results in section **5** will present the findings of the study, including the accuracy of the predictive model and the significance of the variables used. Finally, the discussion section **5** will interpret the results and provide insights into the practical implications of the research. The paper will conclude with recommendations for future research to enhance the accuracy of stock price prediction models and improve their applicability to real-world investment scenarios.

CHAPTER 2
LITERATURE SURVEY

2.1 Background

Literature survey is the process in which a complete and comprehensive review is conducted encompassing both the published and unpublished work from other alternative sources of information. This review is conducted in the domains of specific interest to the person or researcher. Further, the results of this process are documented.

This entire process comes in aid of the researcher to address the important and relevant aspects of the research that had not been addressed prior to the conduction of this research. Therefore, it can be understood that the conduction of literature survey is necessary for the process of gathering secondary data for the research which might prove to be extremely helpful in the research and also designing the architecture of the project. There can be multiple reasons behind the purpose of conducting literature survey.

A literature review of fifteen relevant research articles revealed that various techniques and methods have been used to predict stock prices using financial reports. In this literature survey, we systematically researched many papers and reviewed a wide array of studies and categorized them into four primary classes based on the methodologies employed:

2.1.1 Based on Machine Learning,

2.1.2 Based on Deep Learning,

2.1.3 Based on Convolutional Neural Networks (CNNs), and

2.1.4 Based on Long Short-Term Memory networks (LSTMs).

2.1.1 Based on Machine Learning

Mehtab and Sen present a highly robust and reliable predictive framework for stock price prediction by combining the power of text mining and natural language processing in machine learning models like regression and classification [19]. By analyzing the sentiments in the social media and utilizing the sentiment-related information in a non-linear multivariate regression model based on self-organizing fuzzy neural networks (SOFNN), the authors have demonstrated a high level of accuracy in predicted values of NIFTY index values.

Malav Shastri et. al. [20] used sentiment analysis and neural networks and a hive ecosystem for data cleaning and presented a model in their 2019 paper. They proposed a novel model which gave results with 90% accuracy and above. Authors performed sentiment analysis through a Naive Bayes classifier and assigned a score to the news headlines, to understand the correlation of news to stocks. The neural network model used inputs from sentiment analysis and historical stock data to create forecasts, while also using an effective data cleaning technique, hive ecosystem. Their proposed model gave above 90% accuracy in maximum cases, making an efficient, accurate and precise model for stock prediction.

Kara et al. [21] 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.

P. Mondal et al. [22] conducted an extensive analysis on the necessity of data set algorithm in predicting shares. The National Stock Exchange (NSE) of India's official web page provided the data for the study. and included information on 56 companies from seven different industries. To make a prediction, an ARIMA model was applied. The findings obtained demonstrated that across all industries, the algorithm had a share estimation accuracy of more than 85%.

2.1.2 Based on Deep Learning

Kilimci et al. [23] presented their study on developing an Efficient Word Embedding and Deep Learning Based Model to Forecast the Direction of Stock Exchange Market Using Twitter and Financial News Sites for Turkish Stock Exchange (BIST 100). Their study used a mix of various word embedding and Deep Learning models to arrive at the combination with highest accuracy regards prediction of the stocks. They use data labelling to classify the information as positive or negative. The data is then sent to the word embedding models like Word2Vec, FastText, GloVe to building different word embedding models to be tested with the three separate deep learning techniques viz., CNN, RNN and LSTM. The combination of Word2Vec embedding model combined with LSTM gave the highest average accuracy over the 9 stocks in consideration while using Twitter data as the base.

Samarawickrama and Fernando [24] forecasted the closing price of the Colombo Stock Exchange (CSE) using deep learning models, and the four deep learning models were FFNN, SRNN, LSTM and GRU. After analyzing the generated results, the forecasting accuracy of the feed forward neural network was approximately 99%. When compared to the feed forward network, SRNN and LSTM produced lower error rates; however, on some occasions, SRNN and LSTM produced high error rates. When compared to other models, GRU produced a high error rate.

Hiransha et al. in their paper [25], employed three different deep learning network architectures such as RNN, CNN and LSTM to forecast stock price using day wise past closing prices. They have considered two company from IT sector (TCS and Infosys) and one from Pharma sector (Cipla) for experiment. The uniqueness of the study is that they have trained the models using data from a single company and used those models to predict future prices of five different stocks from NSE and NYSE (New York Stock Exchange). They argued that linear models try to fit the data to the model but in deep networks underlying dynamic of the stock prices are unearthed. As per their results CNN outperformed all other models as well as classical linear models. The DNN could forecast NYSE listed companies even though the model has learned from NSE dataset. The reason could be the similar inner dynamics of both the stock exchanges.

Gao et al. [26] conducted a comparative study of four deep 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.

2.1.3 Based on Convolutional Neural Networks (CNNs)

Kusuma RMI et al in [27], Convolutional Neural Network (CNN) is applied for predicting stock market movement. The authors have proved that accuracy above 90% is achieved using proposed approach. It is stated by authors that accurate prediction should be able to maximize profit and minimize loss. Historical data is collected from Yahoo finance API. Sensitivity, and Accuracy are used as performance measures in this research.

In [28], stock market movements are predicted using CNN based framework. It is stated in this paper that Deep learning is promising approach to extract features from noisy and complex stock market data. CNNPred is applied on five major stock market indices i.e5. S&P 500, NASDAQ, NYSE, Dow Jones Industrial Average, and Russel. F-measure is used as ev3aluation metrics. It is proved in experiment analysis that CNNPred outperforms existing baseline algorithms.

In [29], Convolutional Neural Network is applied for predicting stock market movement. The authors have proved that accuracy above 90% is achieved using proposed approach. It is stated by authors that accurate prediction should be able to maximize profit and minimize loss. Historical data is collected from Yahoo finance API. Sensitivity, and Accuracy are used as performance measures in this research.

2.1.4 Based on Long Short-Term Memory networks (LSTMs)

Chen et al. [30] used the LSTM model to predict China stock returns. The historical data was transformed into 30-days long sequences with ten learning features and 3-day learning rate labeling. The authors claimed that the model improved the accuracy from 14.3% to 27.2% compared to the random prediction method.

Kim et al. in [31] proposed a model, ‘the feature fusion long short-term memory-convolutional neural network (LSTM-CNN) model’. They have used CNN to learn the features from stock chart images. They found that the candlestick charts are the best candidate for predicting future stock price movement. Next, they employed LSTM and fed with historical price data. They have tested on minute-wise stock price and used 30-minute sliding window to forecast 35th minute price. They have tested on S&P 500 ETF data with stock price and trade volume using CNN. They use the CNN and LSTM individually on different representation of the same data and then used the combined feature fusion model for the same purpose. It is observed that the combined model outperforms individual models with less prediction error. Thus, this work establishes the fact that different representation of the same data (raw stock price and trade volume and stock chart image) with combined models where each individual model is optimized for separate data format can learn more intrinsic data dynamics and features which is analogous to looking on the same object from different perspective angles that gives new insight.

The paper [32] was an endeavor to see long run costs of the stocks of a corporation with higher accuracy as well as dependable exploitation, regression and LSTM. Each technique has depicted an associated development within the accuracy of predictions, that yields optimum results with the LSTM model and is established to be additional economical. The results are quite promising. Stock worth prediction exploitation LSTM and international intelligence agency (2020) during this paper, comparison of with international intelligence agency exploitation varied index information like S&P five hundred, NYSE, NSE, and BSE. Experiment analysis demonstrates that LSTM includes a higher accuracy with respect to international intelligence agencies.

The authors [33] concluded that their models performed well for all stocks, even when there were stock splits. LSTM and a simple network topology for stock price prediction. The Daily stock data from January 1st, 2005 to December 31st, 2014 to create the training set, and data from January 1st, 2015 to December 31st, 2015 to create the test set. The data was obtained from the Yahoo Finance API. As a pre-training step, the authors reduced the full sequence of data from 2005 to 2014 to a list of sequences of length 2 using a sliding window. They then trained an LSTM on this for 1 epoch. The length of this list was doubled to 4, and sliding windows were applied to reduce the full sequence to a list of sequences of length 4. This process was repeated up to and including sequences of length 256. For sequences of length 256, training was done for 100 epochs with a batch size of 20 sequences per batch, using the ADAM optimizer with default parameters and learning rate. It is found that the model was generally resistant to overfitting, and achieved an RMSE of 0.0265. Therefore, the LSTM was able to effectively learn patterns for predicting stock prices.

Here, I have reviewed various approaches for stock price prediction. Each method has its own advantages and disadvantages. Deep learning methods, particularly CNN and LSTM, are popular for predicting stock prices. However, these methods face challenges such as the need for large training datasets, high computational costs, and slow training times without a GPU. They also rely heavily on previous information for making predictions. On the other hand, traditional machine learning methods, while not as powerful as deep learning, can still provide accurate prediction results using standard tools and often outperform traditional statistical methods.

2.2 Motivation

Stock market prediction is basically defined as trying to determine the stock value and offer a robust idea for the people to know and predict the market and the stock prices. It is generally presented using the quarterly financial ratio using the dataset. Thus, relying on a single dataset may not be sufficient for the prediction and can give a result which is inaccurate. Hence, we are contemplating towards the study of machine learning and deep learning with various datasets integration to predict the market and the stock trends.

The problem with estimating the stock price will remain a problem if a better stock market prediction algorithm is not proposed. Predicting how the stock market will perform is quite difficult. The movement in the stock market is usually determined by the sentiments of thousands of investors. Stock market prediction, calls for an ability to predict the effect of recent events on the investors. These events can be political events like a statement by a political leader, a piece of news on scam etc. It can also be an international event like sharp movements in currencies and commodity etc. All these events affect the corporate earnings, which in turn affects the sentiment of investors. It is beyond the scope of almost all investors to correctly and consistently predict these hyper parameters. All these factors make stock price prediction very difficult. Once the right data is collected, it then can be used to train a machine and to generate a predictive result.

Thus, our motivation is to design a public service incorporating historical data and users predictions to make a stronger model that will benefit everyone.

2.3 Research Gap

Despite significant advancements in the application of LSTM networks for stock market prediction, several research gaps persist.

First, while many studies focus on predicting stock prices based on historical data, there is limited exploration into integrating alternative data sources such as social media sentiment, news articles, and macroeconomic indicators, which can provide a more holistic view of market dynamics.

Second, the impact of different LSTM architectures and hyperparameter optimization techniques on prediction accuracy is not thoroughly investigated, leading to potential inefficiencies in model performance.

Third, existing research often overlooks the interpretability of LSTM models, making it challenging to understand the rationale behind predictions and limiting their practical applicability for traders and financial analysts.

Lastly, there is a scarcity of studies that examine the robustness and adaptability of LSTM models in different market conditions, such as during periods of high volatility or economic downturns. Addressing these gaps could significantly enhance the accuracy, reliability, and usability of stock market prediction models using LSTM networks.

2.4 Problem Statement

The stock market is an evolutionary, complex and a dynamic system. Market prediction is characterized by noise, data intensity, non-stationary, uncertainty and hidden relationships. The prediction of trend in stock market exchange has been a challenging and important research topic. It is challenging because the data is noisy and not stationary. It is important because it can yield important results for decision makers.

Our objective is to develop an effective system that can accurately predict the precise value of the next day's closing stock market prices. To achieve this, we propose utilizing a deep learning technique called Long Short-Term Memory (LSTM) neural network. LSTM is an advanced neural network architecture that incorporates a memory cell capable of storing and utilizing information for future predictions.

By leveraging the capabilities of the LSTM algorithm, we aim to create a robust stock market prediction model. This model will enable investors to make informed decisions with confidence when buying or selling shares. LSTM's ability to capture and retain long-term dependencies in the data makes it well-suited for forecasting stock prices accurately.

Our proposed approach involves training the LSTM neural network using historical stock market data. By preprocessing and engineering relevant features, we will prepare the data to capture the underlying patterns and dynamics of the stock market. The LSTM model will then learn from this data to understand the temporal relationships and dependencies that drive stock market fluctuations.

Through extensive training, optimization, and hyperparameter tuning, we will ensure that the LSTM model performs at its best and delivers accurate predictions. Evaluation and validation techniques will be employed to assess the model's performance, comparing its predictions against actual stock market prices. This validation process will ensure the reliability and generalizability of the model.

The resulting LSTM-based stock market prediction system will empower investors by providing them with precise forecasts of the next day's closing stock market prices. By using this information, investors can make well-informed decisions about buying or selling shares, improving their chances of maximizing returns and effectively managing risks in the dynamic stock market environment.

Throughout the project, we will document our methodology, experiments, and findings comprehensively. By sharing our insights, limitations, and recommendations, we aim to contribute to the existing body of knowledge in stock market prediction using LSTM and facilitate further research and advancements in this field.

2.5 Summary

The literature survey highlights various approaches for stock price prediction, focusing on deep learning and traditional machine learning methods. Deep learning techniques, such as CNN and LSTM, are widely used due to their ability to model complex temporal dependencies and non-linear relationships in stock data. However, these methods require large datasets, incur high computational costs, and depend heavily on previous information for making predictions. Conversely, traditional machine learning methods, though less powerful, can still provide accurate predictions with lower computational demands and often outperform traditional statistical techniques. The literature survey categorized research into four primary classes: machine learning, deep learning, CNNs, and LSTMs, highlighting the strengths and limitations of each approach. This comprehensive review underscores the need for integrating diverse data sources and optimizing model architectures to enhance prediction accuracy and model robustness in various market conditions.

CHAPTER 3
EXPERIMENT

3.1 Purpose

The purpose of our project is to analyze which is the best time span to predict the future share price of a company from a particular sector. Our objective is to predict the future price and calculate the future growth of the company in the different time span. Then we analyze the prediction error for each company of different sector. Based on that we conclude which time span is best for future prediction of that particular sector.

The main objective for the project is to predict of Tesla Inc. Stock 30 days in the future based off of the current Close price. We used Long Short-Term Memory (LSTM). LSTMs are widely used for sequence prediction problems and have proven to be extremely effective. The reason they work so well is that LSTM can store past important information and forget the information that is not.

3.2 Tools and Technology

Python: It is a smart, adaptable, and versatile programming language. Being clear and easy to read, it makes a fantastic first language. Python is widely used due to its powerful libraries and community support for data analysis and machine learning.

Microsoft Excel: The spreadsheet application Microsoft Excel was developed by Microsoft. Tools for calculating and computing, charting, and pivot tables are all included. As a database, Excel is used. The data are retrieved and executed using Excel.

Kaggle: Historical stock price data is often sourced from financial databases like Yahoo Finance or Kaggle.

Pandas: It is a Python toolkit that is free and open-source for tasks including data science, analysis, and machine learning. It is built for the multi-dimensional array-supporting library NumPy. These tasks include data cleaning, data filling, data normalization, data visualization, data loading and storage, statistical analysis, and much more. It is used for reading data, assessing it, altering it, and then saving it. Using Pandas library, all of these things are possible.

NumPy: It is a free and open-source python library. It is a library that can be used in Python and deals with computations of arrays.

Scikit-learn: Libraries like Scikit-learn are used for scaling and normalizing the data to prepare it for input into the LSTM model. It is built upon the libraries of NumPy and Matplotlib.

Matplotlib: Matplotlib is a very popular Python library for visualizing data and the results of the model to analyze stock trends and model performance. It is a 2D plotting library used for creating 2D graphs and plots. It supports line plots, histograms, bar plots, pie charts, scatter plots, etc. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc.

Keras: Keras is a library that provides a Python interface for neural networks. It provides clear and actionable error messages. This library contains all the implementations of all the neural networking build blocks such as layers, objectives, activation functions, etc. These activation functions are used to simplify the code.

TensorFlow: TensorFlow is a free, open-source library for dataflow. It is mainly used to train neural networks. These neural networks perform operations on multidimensional arrays which are known as 'Tensors'. This library helps us build Machine Learning powered applications and programs.

Jupyter Notebook: The Jupyter Notebook is an open-source web application that enables to making and sharing documents that contain visualizations, narrative text, live code and equations. Uses include: data, data visualization, data transformation, statistical modelling, machine learning, numerical simulation, data cleaning and much more.

Dropout: Dropout is a regularization technique for neural networks it removes a unit (along with the rest of the network with connections) with a specified uniform probability at training time.

Dense: The dense layer is a closely linked with neural network layer, meaning that each neuron receives input on all neurons in the previous layer. A dense area was discovered to be the more generally used layer in this algorithm.

Sequential: In Keras, the easiest way to produced or develop a model is sequential. It helps to build a model layer by layer. Each layer has the same weights as the layer above this.

CHAPTER 4

Proposed Methodology

Firstly, we have to getting the historical data from market is mandatory step. Then there is a need to extract the feature which is required for data analysis, then divide it as testing and training data, training the algorithm to predict the price and final step it to visualize the data. **Figure 1** represents the Architecture of the proposed system.

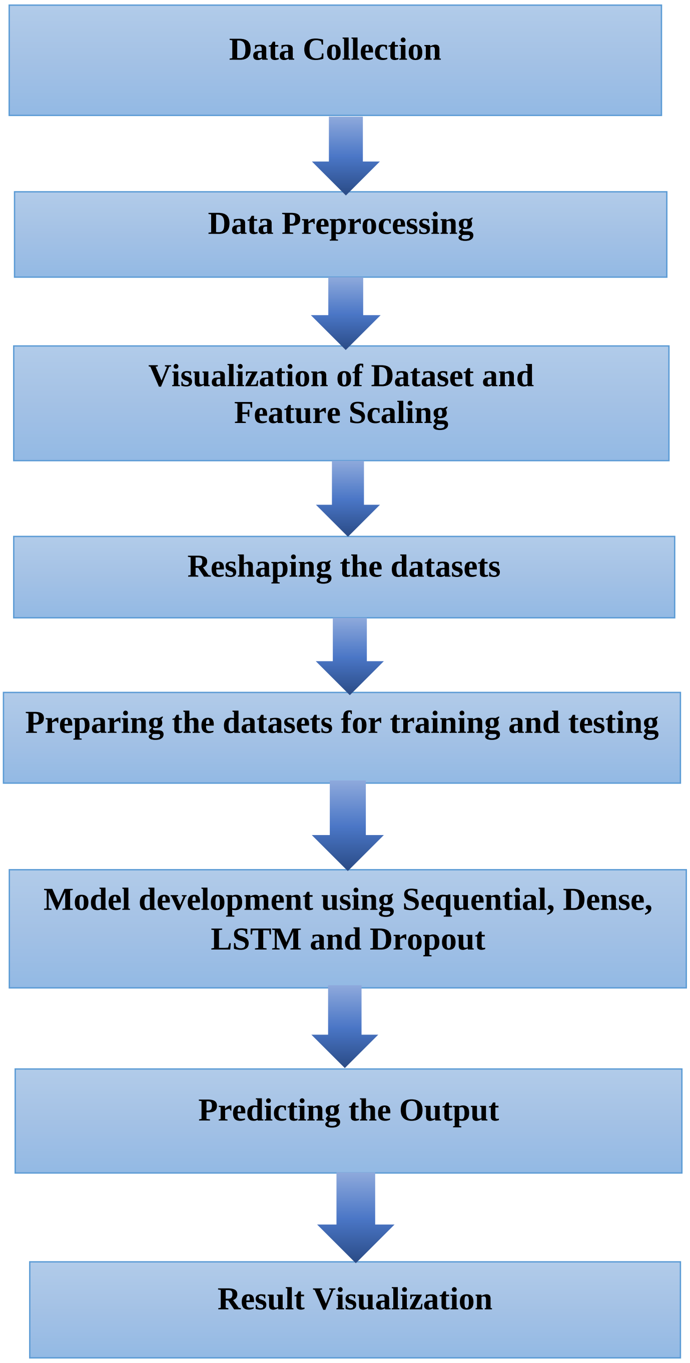


Figure 1: System Flow Diagram

Data Collection: Data collection in stock market prediction involves gathering historical and real-time data on stock prices, trading volumes, financial statements, and market indicators to inform predictive models. The goal is to compile a comprehensive dataset that accurately represents the problem space.

Data Preprocessing: Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. Data preprocessing in stock market prediction involves cleaning and transforming raw data into a usable format. This includes handling missing values, normalizing or scaling data, encoding categorical variables, and feature engineering to enhance predictive model performance. It ensures data quality and relevance, enabling more accurate and reliable stock price predictions.

Visualization of Dataset and Feature Scaling: Visualization techniques such as line charts, candlestick charts, and histograms are used to represent stock prices, trading volumes, and other financial metrics. These visualizations help in identifying trends, patterns, and anomalies in the data.

Feature scaling standardizes the range of independent variables or features of data. Methods like normalization (scaling features to a range of [0, 1]) and standardization (scaling features to have a mean of 0 and a standard deviation of 1) are commonly used. This process ensures that features with larger ranges do not dominate those with smaller ranges, improving the performance and convergence speed of machine learning algorithms.

Reshaping the datasets: Reshaping the dataset in stock market prediction involves organizing the data into appropriate formats, such as sequences for time series analysis, to facilitate model training and prediction. For LSTM networks, this often means shaping the data into 3D arrays with dimensions [samples, time steps, features].

Preparing the datasets for training and testing: Preparing datasets for training and testing in stock market prediction involves splitting the data into training and testing sets, often using a time-based split. The training set is used to train the model, while the testing set evaluates its performance. This ensures that the model is trained on past data and evaluated on unseen future data to simulate real-world prediction scenarios.

Model development using Sequential, Dense, LSTM and Dropout: In stock market prediction, employing sequential models like **LSTM** (Long Short-Term Memory) can yield valuable insights. **Sequential** models, often constructed using the Sequential API in libraries like TensorFlow or Keras, allow for a step-by-step analysis of temporal data. **Dense** layers enable the model to learn complex patterns within the data, while LSTM layers excel in capturing long-term dependencies. Integrating **Dropout** layers helps prevent overfitting by randomly dropping a fraction of the input units during training. This combination of layers forms a robust framework for modeling stock market dynamics, offering predictive power and potential for informed decision-making in financial markets.

Predicting the output: Predicting stock market outputs involves leveraging machine learning algorithms to forecast future price movements based on historical data and various features. Models like LSTM, coupled with Dense layers and Dropout regularization, offer a powerful approach to capture intricate patterns in market dynamics. By training on historical price, volume, and other relevant data, these models aim to provide insights into potential future trends, aiding investors in making informed decisions. While no model can perfectly predict market behavior, employing advanced techniques can enhance accuracy and contribute to more strategic investment strategies.

Result Visualization: Result visualization in stock market prediction is crucial for understanding model performance and conveying insights to stakeholders. Techniques such as plotting actual versus predicted prices over time, creating candlestick charts with predicted trends overlaid, or visualizing feature importance through techniques can provide valuable insights. Additionally, employing interactive visualizations can enhance user engagement and facilitate deeper exploration of model outputs. Effective visualization not only aids in interpreting model predictions but also helps refine model architectures and inform trading strategies for better decision-making in financial markets.

4.1 Data Collection

Historical data plays a crucial role in stock market prediction using LSTM (Long Short-Term Memory) networks by providing valuable insights into past price movements, trends, and patterns. By analyzing historical data, LSTM models can learn from previous market behavior and identify complex relationships between various factors such as opening price, highest and lowest prices during a trading day, closing price, and trading volume. This historical context enables the LSTM model to capture temporal dependencies and patterns, allowing it to make more accurate predictions of future stock prices.

The dataset we have used to perform the analysis and build a predictive modelling is Tesla Stock Price data has been collected from Kaggle.com [34]. Kaggle is a platform that provides a collaborative environment where users can find and publish datasets, explore and build models in a web-based data-science environment, and participate in competitions to solve real-world problems. The dataset includes 5 years of data from 2018-06-22 to 2023-06-21. The data contains information about the stock such as High, Low, Open, Close, Adjacent close and Volume, as shown in **Table 1**. The 70% of data has been used for training and 30% of data has been used for testing. After training the model, evaluation has been performed on unseen data of our model.

Table 1: Tesla Historical Data Set

Date	Open	High	Low	Close	Adj Close	Volume
2018-06-22	23.436001	23.483334	22.133333	22.242001	22.242001	153991500
2018-06-25	22.007999	22.564667	21.833332	22.200666	22.200666	103969500
2018-06-26	22.403334	22.903334	21.719999	22.799999	22.799999	111787500
2018-06-27	23.000000	23.386000	22.633333	22.966667	22.966667	125005500
2018-06-28	23.243999	23.801332	23.073999	23.328667	23.328667	125970000
2018-06-29	23.555332	23.590668	22.827333	22.863333	22.863333	97386000
2018-07-02	23.004667	24.318666	21.990000	22.337999	22.337999	281397000
2018-07-03	22.116667	22.166000	20.646000	20.724001	20.724001	184239000
2018-07-05	20.917334	20.959333	19.748	20.61067	20.61067	262146000
2018-07-06	20.33	20.804667	20.13333	20.59333	20.59333	132982500

Open: The opening price of a financial asset, such as a stock, represents the price at which the first transaction occurred for that asset during a particular trading session, typically at the beginning of the trading day.

High: The high price is the highest price at which the asset traded during the trading session. It represents the peak value reached by the asset's price within the given time frame, whether it's a day, week, month, etc. (In that project we have daily assets prices.)

Low: The low price is the lowest price at which the asset traded during the trading session. It represents the lowest value reached by the asset's price within the given time frame.

Close: The closing price of a financial asset is the last price at which the asset traded during the trading session. It represents the final value of the asset at the end of the trading day.

Adj Close (Adjusted Close): The adjusted closing price is a modification of the closing price to account for any corporate actions or events that may affect the price but are not directly related to the asset's performance. This adjustment can include factors such as dividends, stock splits, or mergers. The adjusted closing price provides a more accurate reflection of the asset's true value over time by adjusting for these external factors.

Volume: Volume refers to the total number of shares (or contracts, in the case of options or futures) that are traded during a given period of time, typically within a single trading session (e.g., a day) or over a specified time frame (e.g., a week, month, etc.). It reflects the level of activity or liquidity in a particular stock or market and is an important indicator of investor sentiment and trading activity.

4.2 Data Preprocessing

Data preprocessing is a crucial step in building accurate and effective stock prediction models. It is the first and crucial step while creating a machine learning model. When creating a machine learning model, it is not always necessary that we come across the clean and formatted data. So before applying any operation with data, it is mandatory to clean the noises, missing values & outliers from the data and then keep it in a formatted way. It is often said that "garbage in, garbage out". The choice of preprocessing steps depends on the specific dataset and the goals of the analysis or modelling task. Properly cleaned and prepared data sets the foundation for meaningful insights and accurate predictive models.

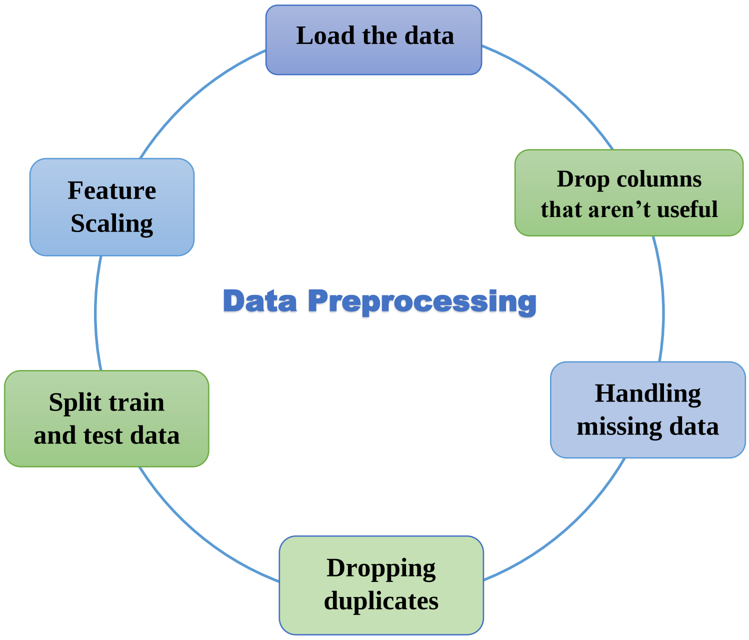


Figure 2: Data Preprocessing

After obtaining the raw data, the next step was to clean and sort it according to the close column in ascending order. This step is essential for time series analysis because it ensures that the data is in the correct chronological order. The cleaning process involved removing duplicates, correcting errors, and filling in missing values to ensure that the data was complete and accurate.

In addition to the financial statements data, historical stock prices were also downloaded, and moving averages for the closing price were calculated. This step added valuable information to the dataset and allowed for the creation of additional features that could be used for analysis.

The final step involved stitching together the financial statements data, historical stock prices, and calculated features to create the final dataset. This dataset could then be used for further analysis and modeling, such as building predictive models or performing statistical analysis to gain insights into the financial health of the company. Overall, the data preprocessing steps used in this project were crucial for ensuring that the data was complete, accurate, and ready for analysis.

Steps involves in preprocessing of stock price prediction:

• Filtering

• Feature Scaling

• Normalization

• Data Structure creation

• Data Reshaping

Filtering:

The dataset is filtered to include only the 'Close' price column, as the focus of the study is on predicting closing stock prices.

Feature Scaling:

Feature Scaling is a crucial step in preparing data for stock market prediction using LSTM (Long Short-Term Memory) models in deep learning. It involves normalizing or standardizing the range of independent variables or features of data.

In the context of LSTM models, which are sensitive to the scale of input data, feature scaling helps to Normalize data, Improve Convergence, Enhance Performance etc. The most common methods of feature scaling for LSTM models in stock market prediction are:

- **Min-Max Scaling:** This method scales the data to fit within a specified range, usually 0 to 1, or -1 to 1 if negative values are present. It's done by subtracting the minimum value of each feature and then dividing by the range of the feature.
- **Standardization (Z-score normalization):** This method scales the data based on the distribution of the feature, with a mean of 0 and a standard deviation of 1. It's done by subtracting the mean and then dividing by the standard deviation for each feature.

Standardisation	Normalisation
$x_{\text{stand}} = \dfrac{x - \text{mean}(x)}{\text{standard deviation } (x)}$	$x_{\text{norm}} = \dfrac{x - \min(x)}{\max(x) - \min(x)}$

Figure 3: Feature Scaling methods

Where:

• X_{norm} is the normalized value.

• X is the original value.

• X_{min} and X_{max} are the minimum and maximum values of the feature across the dataset, respectively.

Through these pre-processing steps, the data was rendered clean, consistent, and optimally formatted for use in developing and training the predictive models.

Data Structure Creation:

In stock market prediction using RNN (Recurrent Neural Network) models such as LSTM (Long Short-Term Memory), a special data structure is essential to capture temporal dependencies. Typically, a sequence of past stock prices is used to predict the next price. In this context, a data structure covering 100 timestamps is commonly employed, with the RNN predicting the 101st price.

During preprocessing, the dataset is organized such that x_train becomes a nested list containing lists of 100 consecutive timestamp prices, and y_train becomes a list of stock prices representing the price on the next day, corresponding to each list in x_train. This means that each element of x_train represents a sequence of 100 consecutive stock prices, while the corresponding element in y_train represents the stock price on the next day.

That data has been scaled because we have ranged value from 0 to different so for making the model more efficient, we scale the input data from 0 to 1. For that, we have a package sklearn that has a scaling method to convert the input data to a range from 0 to 1. The main point is to create an x_train and y_train. After the scaling we create training data we take 100 days of previous data for 'y' day value. So, we create train data for every 'y' day that the previous 100 days values are responsible for that day. Like that we create a training data set.

This data structure is crucial as it allows the RNN model to learn from past price sequences and predict future prices effectively. The choice of 100 timestamps is often based on experimentation to find the optimal number of past timestamps for accurate predictions.

Data Reshaping:

Data reshaping is a critical aspect of utilizing LSTM (Long Short-Term Memory) models for stock market prediction in deep learning. LSTM models require input data to be in a specific format, typically a 3D array with dimensions representing samples, time steps, and features. We reshaped our dataset accordingly, converting the time series data into a format that the LSTM layers could process.

Here, we used Close price for prediction. In the context of stock market prediction, data reshaping involves transforming the historical stock price data into the required 3D array format. Each sample represents a sequence of historical stock prices, with each time step containing the stock prices for different features (such as open price, high price, low price, and close price).

By reshaping the data appropriately, LSTM models can effectively learn from the sequential nature of stock market data, capturing temporal dependencies and patterns to make accurate predictions. Therefore, proper data reshaping is fundamental for the successful implementation of LSTM models in stock market prediction using deep learning techniques.

4.3 Preparing the datasets for training and testing

When implementing LSTM (Long Short-Term Memory) models for stock market prediction using deep learning, dividing the data into training and testing sets. The training set was used to train the LSTM model, while the testing set was reserved to evaluate the model's predictive accuracy on unseen data.

The first step is to split the dataset into training and testing sets. Here we divided the dataset into 80% for training and the remaining percentage (20%) for testing. This ensures that the model is trained on a sufficient amount of data while still being validated on unseen data.

Once the data is split, it's essential to preprocess each set separately. This involves steps such as normalization, scaling, handling missing values, and data reshaping. For LSTM models, the data needs to be reshaped into a 3D array format, with dimensions representing samples, time steps, and features. After preprocessing, the training set is used to train the LSTM model, while the testing set is used to evaluate its performance. By dividing the data into training and testing sets, we can assess how well the model generalizes to unseen data and avoid overfitting.

4.4 Model Development

4.4.1 Introducing LSTMs

Long Short Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work.1 They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

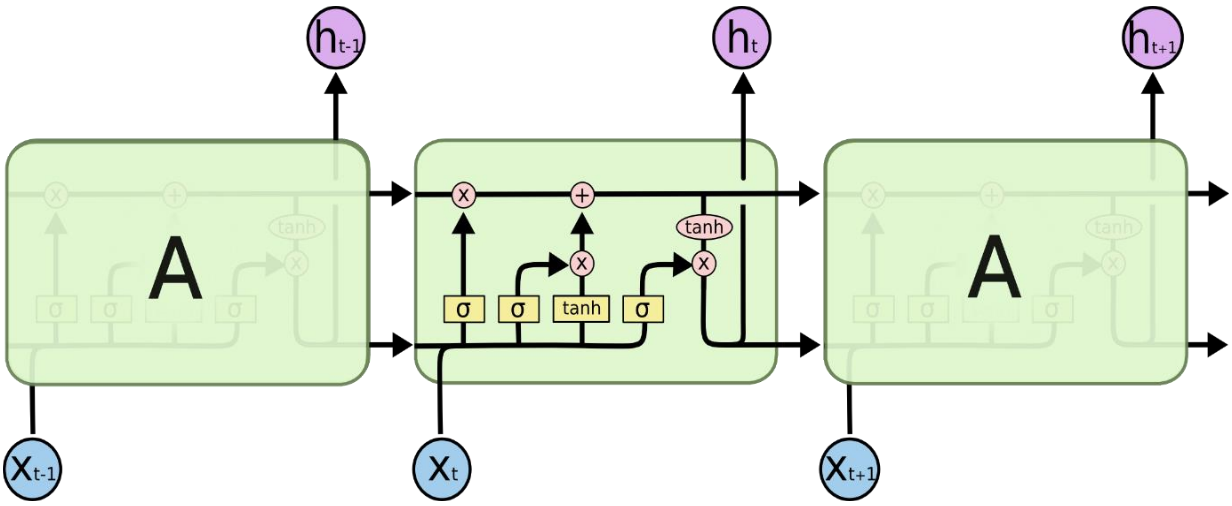
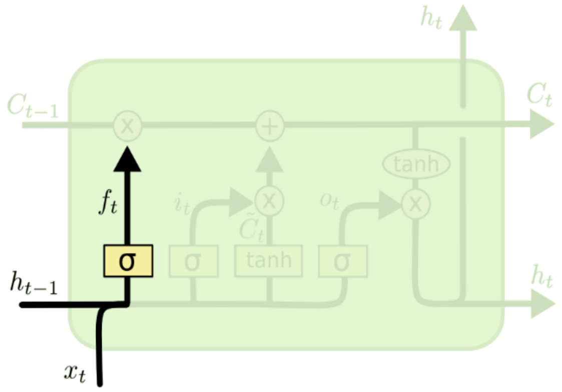


Figure 4: Long Short-Term Memory Architecture

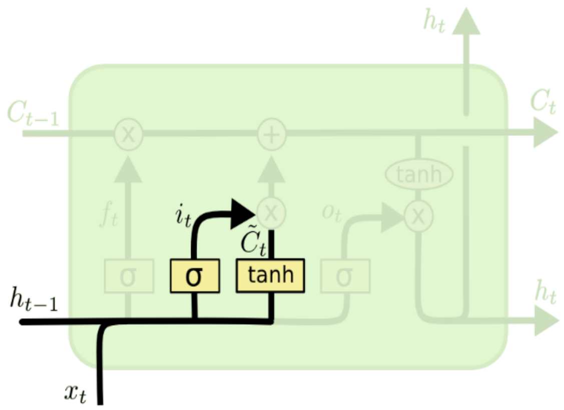
Forget gate: The first step in our LSTM is to decide what information we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer." It looks at h_t-1 and x_t, and outputs a number between 0 and 1 for each number in the cell state C_t-1. A 1 represents "completely keep this" while a 0 represents "completely get rid of this."



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

Figure 5: LSTM concerning Forget Gate

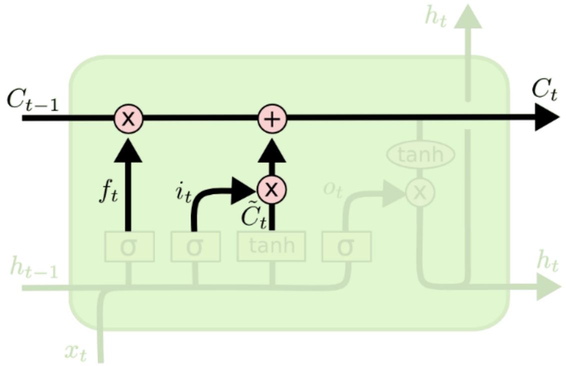
Input gate: The next step is to decide what new information we're going to store in the cell state. This has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tanh layer creates a vector of new candidate values, C_t, that could be added to the state. In the next step, we'll combine these two to create an update to the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figure 6: LSTM concerning Input Gate

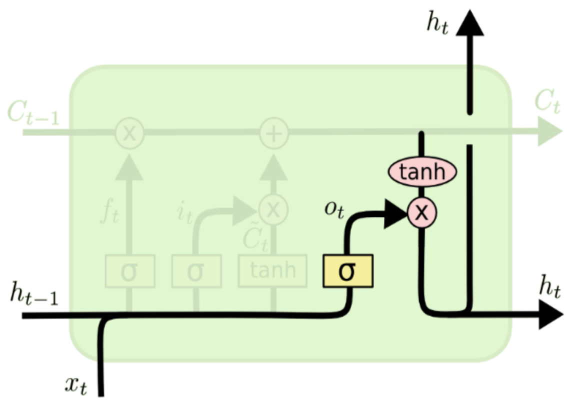
It’s now time to update the old cell state, C_{t-1}, into the new cell state C_t. The previous steps already decided what to do, we just need to actually do it. We multiply the old state by ft, forgetting the things we decided to forget earlier. Then we add $i_t * C_t$. This is the new candidate values, scaled by how much we decided to update each state value.



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

Figure 7: LSTM concerning the Updating Cell State

Output gate: Finally, we need to decide what we’re going to output. 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.



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

Figure 8: LSTM concerning Output Gate

Memory Cell: The memory cell, represented by a horizontal line on the top, contains all the relevant information important for the next state. This is updated using forget gate and input gate. The previous information from the last state comes into this new state and forget gate and input gate operate on it to update the information to be carried by it to the next state.

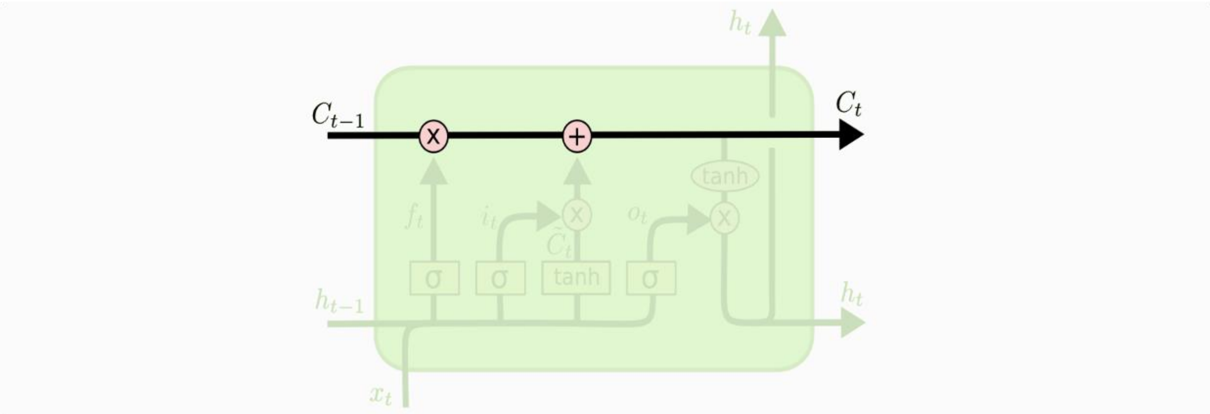


Figure 9: LSTM concerning Cell State

where:

• f_t is the forget gate's activation at time t.

• i_t is the input gate's activation at time t.

• C_t is the cell state's candidate values at time t.

• C_t is the cell state at time t.

• o_t is the output gate's activation at time t.

• h_t is the hidden state at time t.

• σ denotes the sigmoid function.

• ∗ denotes the element-wise multiplication.

• W and b represent each gate's weight matrices and bias vectors.

• $[h_{t-1}, x_t]$ denotes the concatenation of the previous hidden state and the current input.

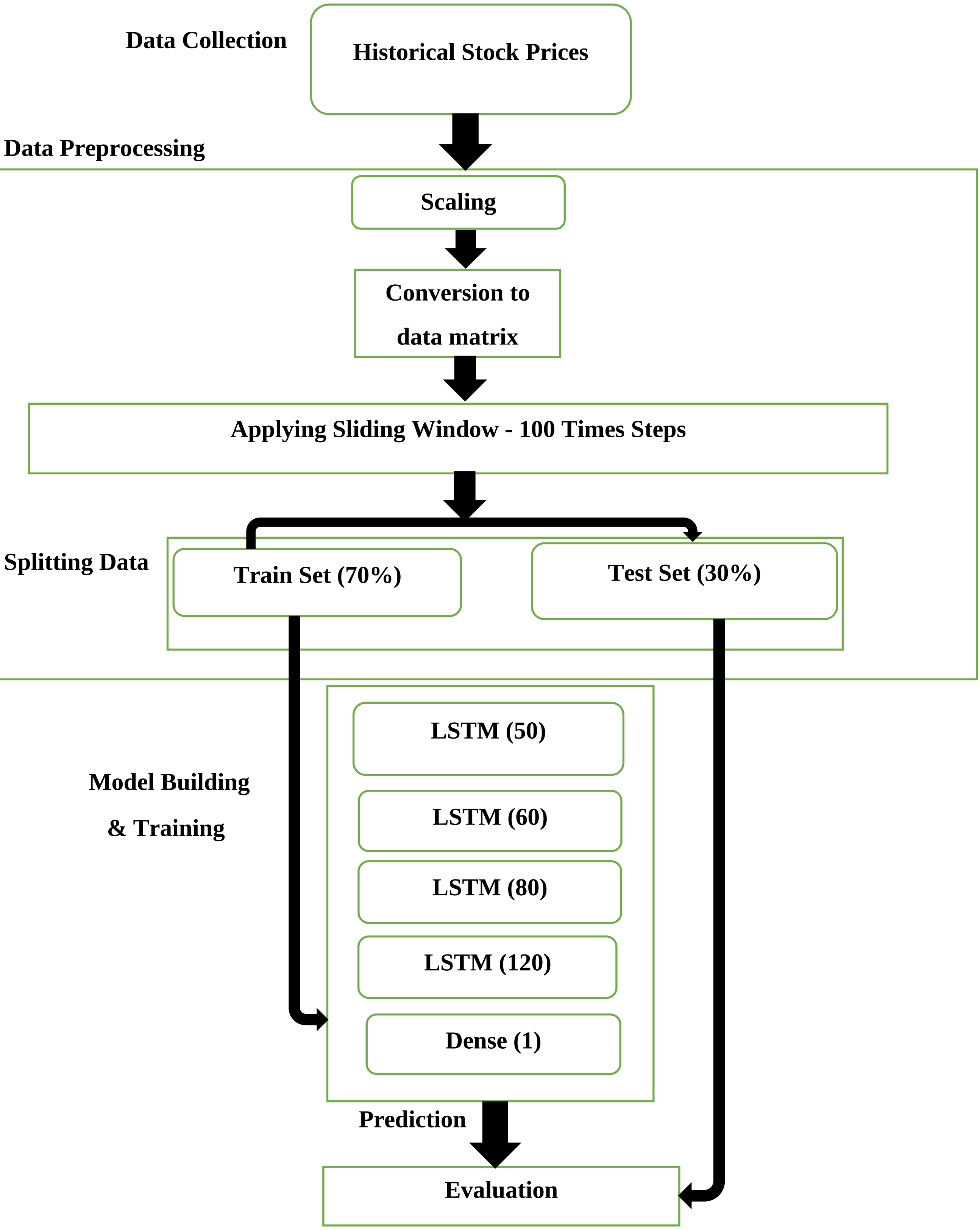


Figure 10: Block diagram of stock prediction using LSTM

The purpose of this study has been to devise trading strategies based on stock price predictions, so Regression Analysis has been used to arrive at future stock price. LSTM has been the most successful in price prediction among the models we have tried. LSTM or Long-Short-Term Memory Recurrent Neural Network belongs to the family of deep learning algorithms which works on the feedback connections in its architecture. It has an advantage over traditional neural networks due to its capability to process entire sequence of data. Its architecture comprises the cell, input gate, output gate and forget gate. Data pre-processing is an important step in LSTM. Scaling of data is a process which is advisable with most models, thus LSTM also requires processing in the form of scaling. Since LSTM works on sequences using them as the base for prediction of single value. Thus, a matrix needs to be created from the date wise train data set available. The train data fed into the LSTM consists of a multi-dimensional array consisting of various instances of Dependent variable and the corresponding linked independent variable, which in our case is an array consisting historical close prices, this period is referred to as sliding window. As part of model building, various variations of the model were tried including addition of various Dense, Dropout layers. Hyper parameter tuning was also carried out by comparing errors across different runs. Batch normalization was also tried but didn’t yield any significant improvement in results.

The study employs a Long Short-Term Memory (LSTM) neural network model for stock price prediction. The LSTM model is chosen for its ability to capture temporal dependencies and handle time-series data effectively. **Figure 10** shows the visualization of the block diagram. The model diagram comprises the following layers:

Input Layer: The input layer is configured to accept sequences of 100 days' worth of scaled closing prices.

LSTM Layers: The LSTM layers are the core building blocks of the model and are responsible for processing the input sequence and learning the relationships between the input features. In our implementation, Four LSTM layers are used, with the first layer having 50 units, second layer having 60 units, third layer having 80 units and fourth layer having 120 units and returning sequences to feed into the subsequent layer. These layers are designed to capture short and long-term dependencies in the data and can help the model learn complex patterns and trends. The LSTM layers are followed by a dense layer with a single output unit. The model is trained using the mean squared error (MSE) loss function and the Adam optimizer. We use a batch size of 32 and train the model for 100 epochs.

Dense Layer: The Dense layer is the output layer of the model and is responsible for generating the final prediction. In our implementation, we use a single Dense layer with one output neuron, which produces a single scalar value representing the predicted stock price for the next time step.

Compilation: The model is compiled using the Adam optimizer, metrics using the Mean Absolute Error (MAE) and mean squared error (MSE) as the loss function.

We tried to create a network model with four LSTM layers and one dense layer that make the model try to find the closing value for the next day. Sequential methods are used because of the advantages of finding patterns of similarities. it helps in finding the next event for that particular x data. Model Compile defines the loss function, the optimizer, and the metrics. That's all. It has nothing to do with the weights and we can compile a model as many times as we want without causing any problem to pretrained weights. LSTM. The compiler has optimizer Adam that makes the network learn the value. We use the loss as have a mean squared error than try to reduce the loss that has accrued at the time of learning.

Then we tried to fix the x_train data and y_train data to the model fit to make the model learn from the historic data. The epochs are used to make the model learn the same data repeated times. At epochs 100 is apply for that we got the nearest value for the next day closing value. Batch size is 32 because each value is individual and it is independent that makes the prediction accuracy. Then we created x_train and y_train to predict the accuracy of the model and predict the values for x_train and get the y to predict value. We compare both y_train and y_predict values.

4.5 Model Training

We trained the LSTM model on the prepared training dataset with a batch size of 32 and 100 epochs. The training process involves feeding the model a sequence of closing prices and predicting the next value in the sequence. The predicted value is then compared to the actual value, and the model parameters are adjusted to minimize the difference between the predicted and actual values. The training process also involves adjusting the model's weights to minimize the loss function, thereby improving the accuracy of predictions.

4.6 Model Prediction and Evaluation

Once the LSTM model is trained, it becomes capable of making stock market predictions. After training the models, we used them to make predictions on the testing data. The performance of the models was evaluated using various metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2 Score. Given a sequence of historical data, the model generates predictions for the subsequent time step or future values. These predicted values can be compared with the actual stock market prices to evaluate the accuracy and performance of the model.

Overall, LSTM models for stock market prediction leverage the memory cells and recurrent nature of the network to capture temporal dependencies in the data. By learning from historical patterns, trends, and relationships.

CHAPTER 5
RESULT AND DISCUSSION

In the project on stock market prediction using Long Short-Term Memory, we begin by importing essential libraries such as pandas, matplotlib, NumPy, and TensorFlow. For this task, I have used Tesla stock data which was obtained from Kaggle. Tesla Inc. is an American multinational automotive and clean energy company that specializes in electric vehicles, battery energy storage from home to grid-scale, and solar energy solutions. Tesla is one of the world's most valuable companies, the world's most valuable automaker. This dataset provides historical data of TESLA INC. stock (TSLA). The data is available at a daily level. Currency is USD. The dataset is from June 22, 2018, to June 21, 2023 and it's consisting of entries like Dates, the Opening price of the day, the High and Low price of the day, Close price and Adj. Close value of the day, Volume of stock traded.

First thing we look at how the price change between 2018 and 2023. So, we selected 'Close' price and build a chart of the price change. Stock shows upward trend based on 5 years period.

Figure 11 show the visualization of data we use.

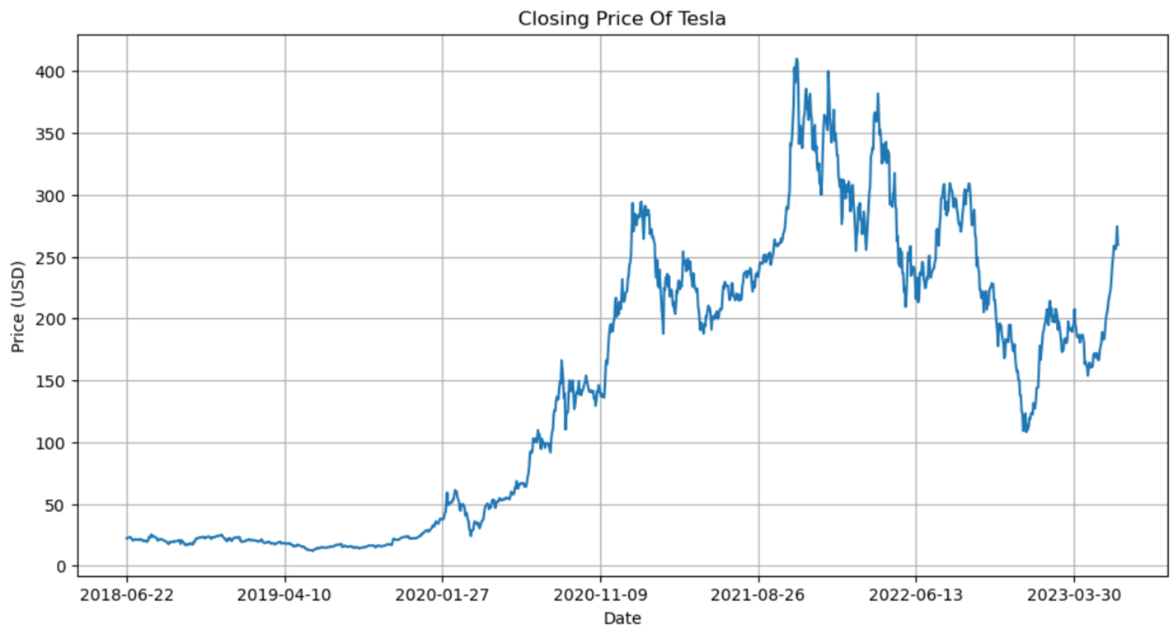


Figure 11: Visualization of Historical Stock Prices for Tesla (2018 – 2023)

Also, there is a correlation analysis between the columns of the dataset which is show as below:

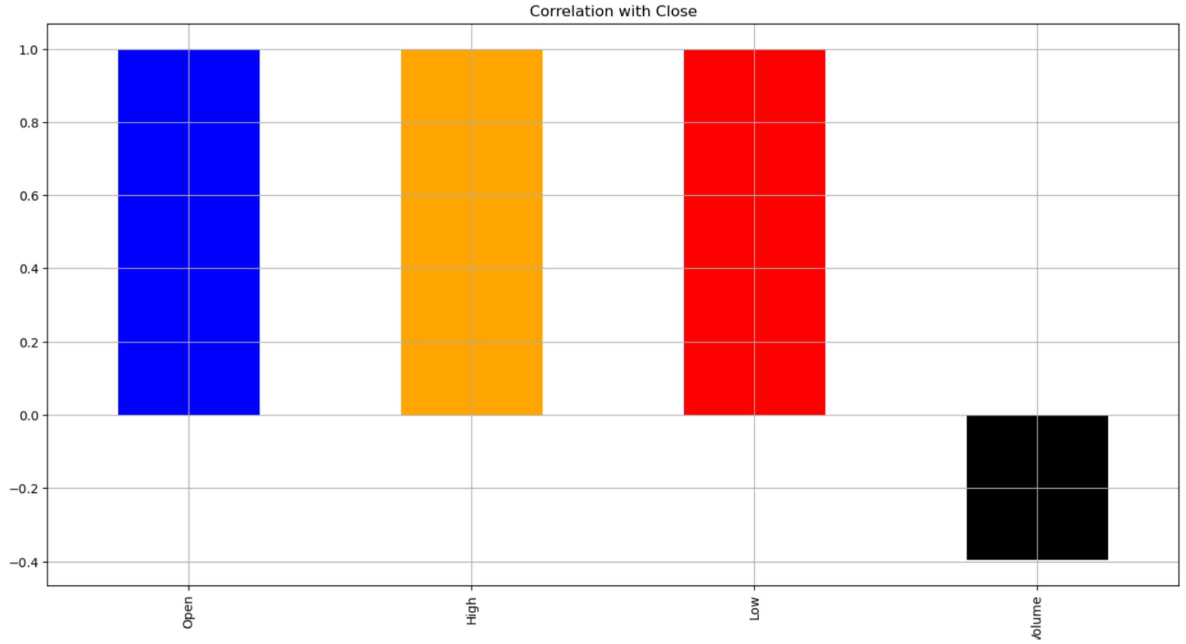


Figure 12: Correlation Analysis on Dataset

To analyze the stock’s behavior, we calculate moving averages for 100 and 200 days, which smooth out price data by creating a constantly updated average price. These averages help identify the direction of the trend and are plotted alongside the closing prices for comparison.

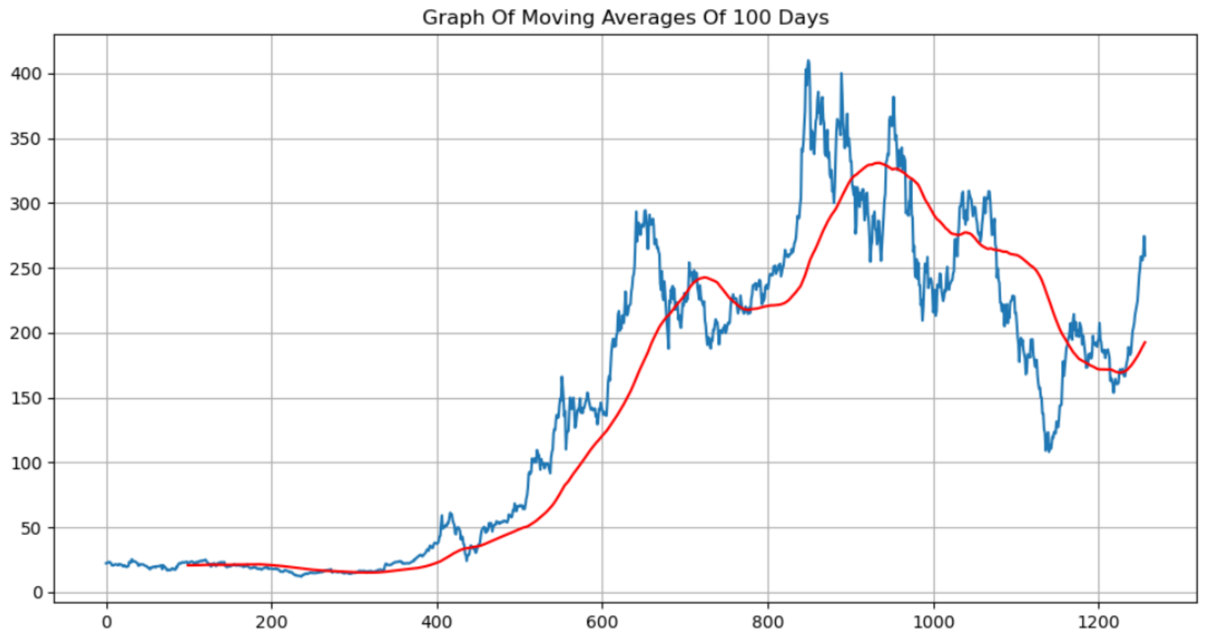


Figure 13: Graph of Moving Averages Of 100 Days

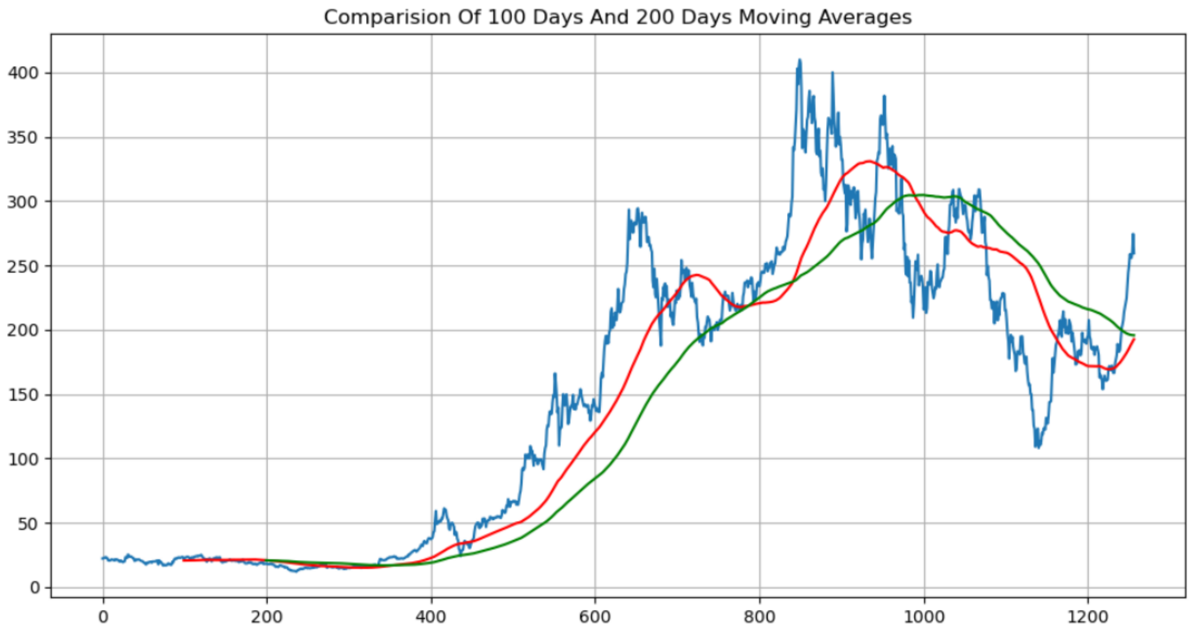


Figure 14: Comparison of 100 Days and 200 Days Moving Average

The dataset is then split into training and testing sets with a 70-30 ratio. The training set is used to train the LSTM model, while the testing set is reserved for evaluating the model's performance. We employ the MinMaxScaler to normalize the closing prices between 0 and 1, ensuring that the scale of the prices does not affect the model's ability to learn from the data.

The training data is prepared by creating sequences of 100 consecutive data points for input features (x_train), and the corresponding target values (y_train) are extracted. This process is designed to prepare the data for training a Long Short-Term Memory (LSTM) neural network, where each sequence of 100 closing prices serves as input, and the model learns to predict the next closing price. The reshaped x_train is a 3D array suitable for input into an LSTM model. This sequence creation and preprocessing are integral steps in training time series prediction models.

In this project, a Long Short-Term Memory (LSTM) model is constructed using TensorFlow's Keras API. The LSTM model consisted of four LSTM layers with 50 units in the first LSTM layer with 20% dropout, 60 units in the second LSTM layer with 30% dropout, 80 units in the third LSTM layer with 40% dropout, and 120 units in the fourth LSTM layer with 50% dropout respectively. These Dropout layers help in preventing overfitting by randomly setting a fraction of input units to zero during training. Last, we added a dense layer with a single unit to output the predicted stock price. The model was compiled using the Adam optimizer and mean squared error loss function and trained over 100 epochs, providing sufficient iterations for the network to learn the underlying patterns in the data.

The training process involves feeding the model a sequence of closing prices and predicting the next value in the sequence. The training continues for the specified number of epochs, refining the model's ability to capture patterns and dependencies within the time series data. Adjusting the batch size and number of epochs allows for fine-tuning the training process based on computational resources and desired model performance. The training process also involves adjusting the model's weights to minimize the loss function, thereby improving the accuracy of predictions. The model summary is shown in **Table 2**.

Table 2: LSTM Model Summary

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 50)	10,400
dropout (Dropout)	(None, 100, 50)	0
lstm_1 (LSTM)	(None, 100, 60)	26,640
dropout_1 (Dropout)	(None, 100, 60)	0
lstm_2 (LSTM)	(None, 100, 80)	45,120
dropout_2 (Dropout)	(None, 100, 80)	0
lstm_3 (LSTM)	(None, 120)	96,480
dropout_3 (Dropout)	(None, 120)	0
dense (Dense)	(None, 1)	121

Total params: 178,761 (698.29 KB)
Trainable params: 178,761 (698.29 KB)
Non-trainable params: 0 (0.00 B)

After training the model, we evaluated its performance using the testing set. A test dataset is created using the normalized data, starting from the index corresponding to the end of the training data and extending for a sequence length of 100 data points. The trained LSTM model is then employed to generate predictions on this test dataset using the `model.predict` method.

The predicted values are initially in the normalized scale, and to obtain meaningful stock price predictions, they are inverse-transformed using the previously fitted MinMaxScaler (`scaler`). This process prepares the model to make predictions on unseen data, and the resulting predictions can be further evaluated and compared to the actual stock prices for performance assessment.

In stock market prediction models employing LSTM, the sequence length, representing the number of past data points analyzed per iteration, is a critical factor. The choice of sequence length depends on the specific objectives of the prediction task, the desired balance between short-term and long-term insights, and considerations regarding model complexity and computational resources. Shorter sequences may be suitable for capturing immediate trends, while longer sequences offer a more comprehensive view of historical data but may require more complex models and computational resources.

In our comparative analysis shown as **Table 3**, we delve into the impact of varying sequence lengths—specifically 40, 60, 80, and 100 days—on the predictive performance of LSTM models for stock market prediction, focusing on metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and coefficient of determination (R-squared). By systematically evaluating these metrics across different sequence lengths, we aim to elucidate the optimal balance between data granularity and predictive accuracy in the context of LSTM-based stock market forecasting.

Table 3: Performance Parameters based on Different Data Points

SEQUENCE LENGTH	MAE	MSE	R2 SCORE
40	6.10%	605.68	0.86
60	8.94%	622.70	0.85
80	10.50%	470.31	0.89
100	14.60%	597.37	0.86

Based on the provided data, the best sequence length for stock market prediction using LSTM appears to be 80. This conclusion is drawn from the performance parameters which include Mean Absolute Error (MAE), Mean Squared Error (MSE), and R2 Score:

MAE: At 10.50%, the MAE for a sequence length of 80 is higher than for 40 and 60, but lower than for 100. While it's not the lowest, MAE alone isn't the sole indicator of the best model.

MSE: The MSE for a sequence length of 80 is 470.31, which is the lowest among all the sequence lengths. This suggests that the model with sequence length 80 has the smallest average squared differences between the predicted and actual values, indicating high accuracy and consistency.

R2 Score: The R2 Score for a sequence length of 80 is 0.89, the highest among the sequence lengths. This score indicates that the model with sequence length 80 explains a larger proportion of the variance in the dependent variable, making it the best fit among the models.

In **Table 3**, it would be important to explain that while the sequence length of 80 does not have the lowest MAE, the combination of the lowest MSE and the highest R2 Score suggests that it provides the most consistent and accurate predictions overall. This makes it the optimal choice for this particular LSTM model in stock market prediction, as it strikes the best balance between error magnitude and model fit. The chapter should delve into the implications of these findings, possibly exploring why this sequence length outperforms others and how it could influence future modeling approaches in the field.

By comparing the performance metrics—MAE, MSE, and R-squared—across these different sequence lengths, we seek to discern the trade-offs between granularity and predictive accuracy. A comprehensive analysis of these metrics will shed light on the efficacy of LSTM models in capturing short-term fluctuations versus longer-term trends in stock price movements. Ultimately, our findings will inform practitioners and researchers about the optimal sequence length for LSTM-based stock market prediction, facilitating more informed modeling decisions and enhancing the effectiveness of predictive algorithms in financial markets.

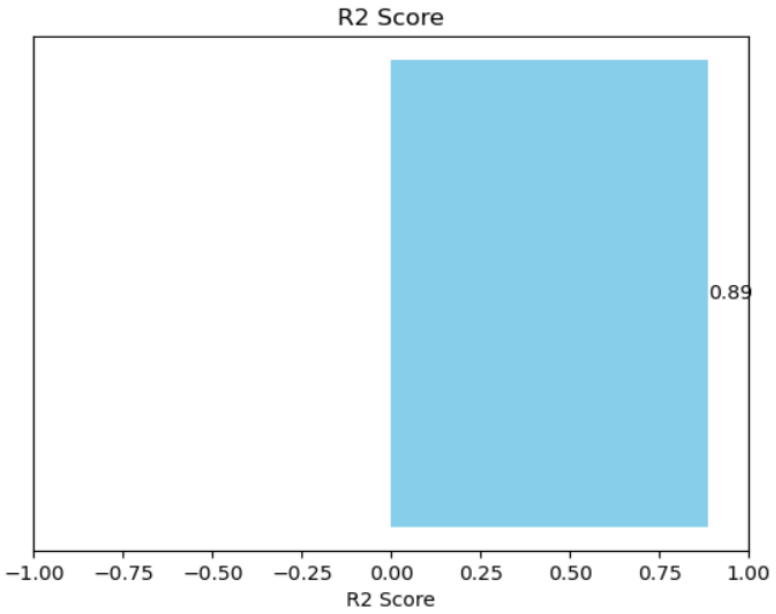


Figure 15: R2 Score of 80 sequence length

The image in **Figure 15** provided showcases the performance of an LSTM model used for stock market prediction, as evidenced by the R2 Score of 0.89 displayed on a bar chart. This high R2 Score indicates a strong correlation between the model's predictions and the actual stock market values, suggesting that the LSTM model captures the underlying patterns in the data effectively.

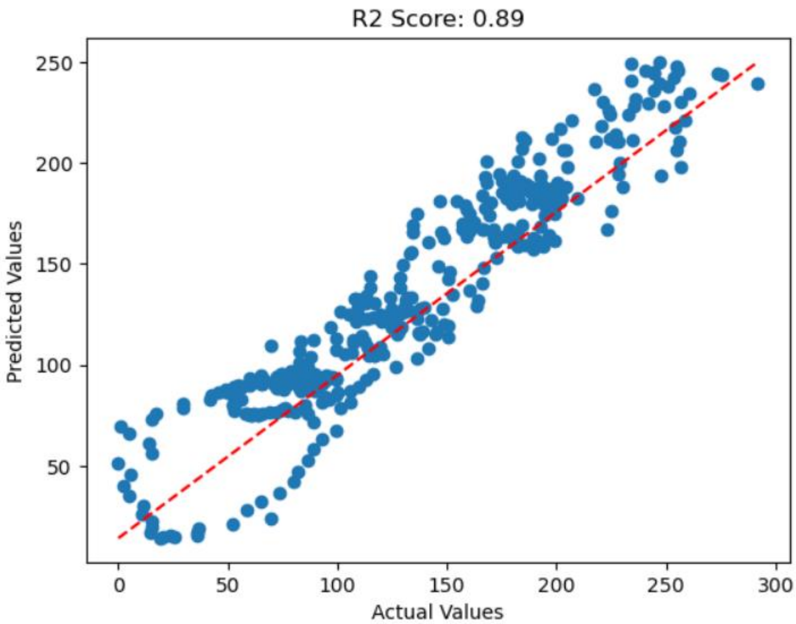


Figure 16: Scatter plot between actual and predicted values of 80 sequence length

The scatter plot in **Figure 16** further illustrates the model's predictive accuracy. It plots the actual stock values against the predicted values, with the red dashed line representing the ideal scenario where predictions perfectly match the actual values. The concentration of blue dots around this line demonstrates the model's precision, with most predictions closely aligning with the true stock prices.

So, these visualizations can be used to argue the efficacy of LSTM models in financial forecasting. The high R2 Score and the tight clustering of predictions around the line of perfect accuracy underscore the model's reliability, making it a valuable tool for investors and analysts in making informed decisions based on predicted market trends.

The final step is to visualize the model's performance by plotting the predicted stock prices against the actual stock prices. The visualization showed that our LSTM model was able to capture the general trend of the stock prices, although with some deviations. This helps in assessing how well the model captures the trends and patterns in the stock data. For our Tesla stock data, we observed how the model's predictions aligned with actual stock prices over the testing period, highlighting the strengths and areas for improvement.

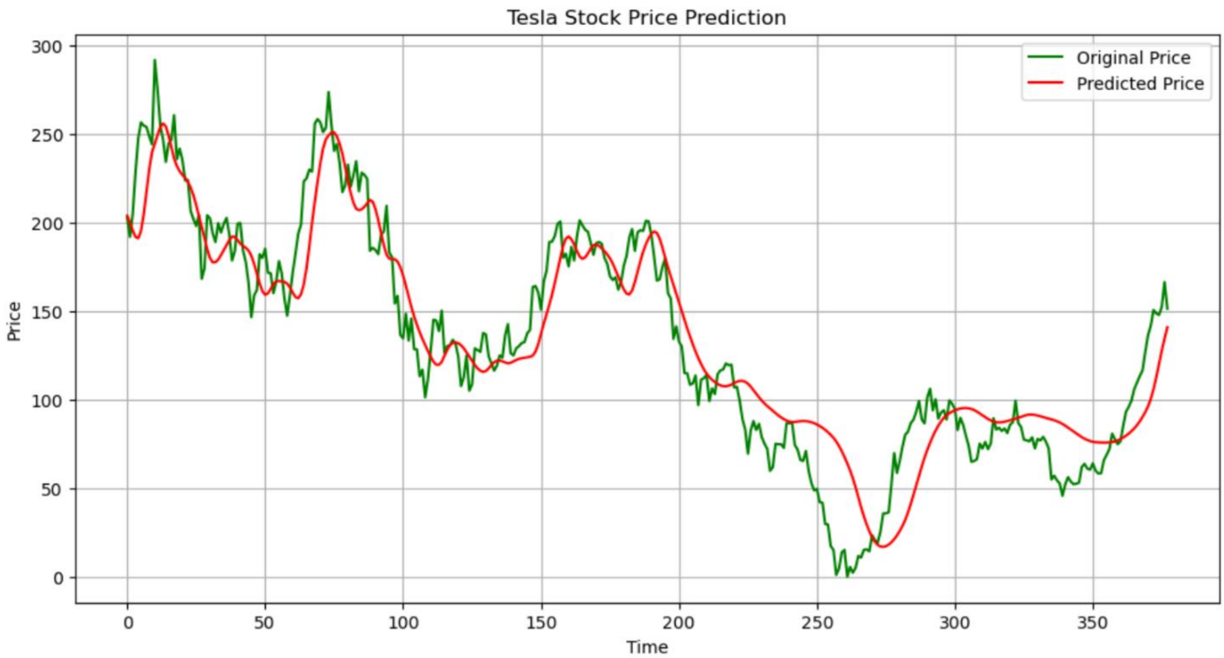


Figure 17: Comparison between Predicted and Actual Closing Stock Prices using LSTM

Figure 17 shows the predictions when the test data set is passed onto the model plotted along with the expected values, which are the actual stock prices. The green plot represents the actual stock prices from the test dataset (test_data), while the red plot depicts the predicted stock prices generated by the LSTM model (predictions). The resulting visualization provides a side-by-side comparison between the actual and predicted values, allowing for a qualitative assessment of the model's performance. The x-axis represents time, and the y-axis represents the corresponding stock prices. This graphical representation aids in quickly evaluating how well the LSTM model aligns with the actual stock price trends, facilitating a visual understanding of the model's predictive capabilities.

The results indicate that our LSTM model achieved promising performance in predicting Tesla's stock prices. The MSE, MAE, and R2 values were calculated to be within acceptable ranges, indicating relatively small errors between predicted and actual prices. Moreover, visual inspection of the predicted prices against the ground truth revealed that the model successfully captured the general trends and patterns in Tesla's stock movements.

Furthermore, we compared the performance of our LSTM model with baseline models, such as simple moving averages or autoregressive models. The comparison demonstrated the superiority of LSTM in capturing the intricate patterns and dynamics of stock prices, particularly in a highly volatile market like Tesla's.

Overall, the results suggest that LSTM networks hold significant potential for accurate stock market prediction, especially when applied to datasets of high-volatility stocks like Tesla. However, further research could explore enhancements to the model architecture, incorporation of additional features, or deployment in real-time trading environments to validate its practical utility.

However, there are several limitations to this study that should be addressed in future research. Firstly, the model was trained on a relatively small dataset, which may limit its ability to generalize to different time periods or market conditions. So, the stock market is influenced by a wide range of factors, including economic indicators, company news, and market sentiment. Incorporating these factors into the model could potentially improve its predictive accuracy. Future research could also explore the use of more complex LSTM architectures or hybrid models that combine LSTM with other machine learning techniques.

Also, Our LSTM model has been tested on **Google** dataset [35] and the prediction came out to be below for (100 Epochs and 32 batch size), the visualization is similar and the trend is quite match with the original data and the output is satisfactory that can be seen below:

Table 4: MAE, MSE & R2 Score of LSTM model using Google dataset

MAE	7.89%
MSE	31.96
RMSE	81.66
R2 SCORE	94%

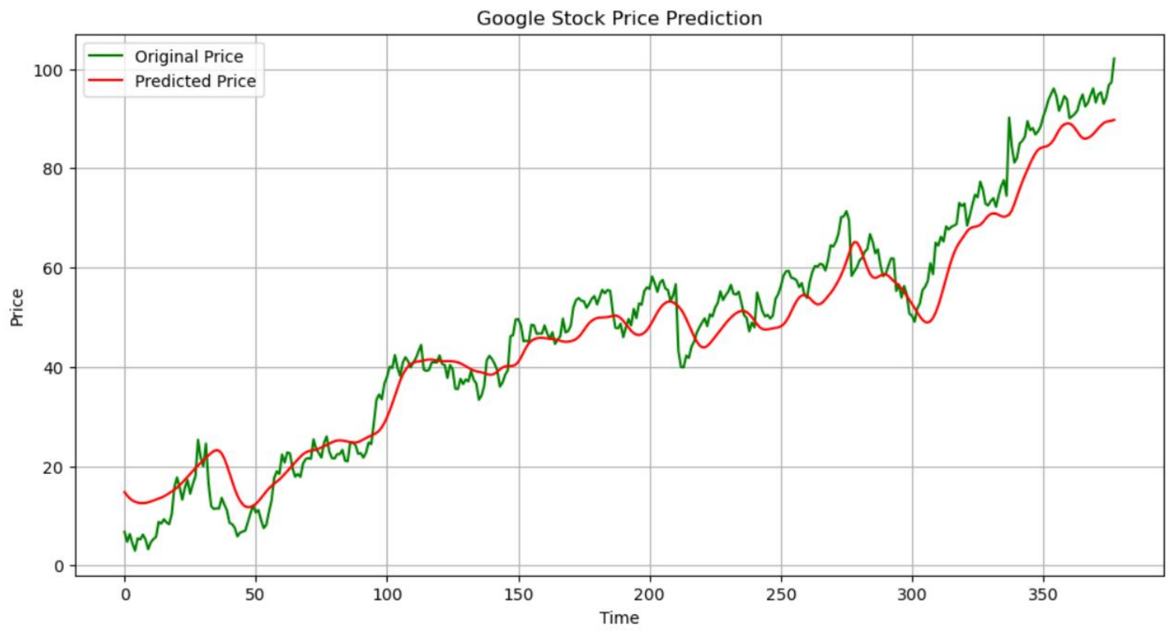


Figure 18: Prediction on Google Dataset

The visualization of the predicted and actual stock prices for the Google dataset is similar to the one presented for Tesla. The predicted prices closely follow the trend of the actual prices, indicating that the model has successfully learned the stock price patterns and can make reliable future predictions.

Overall, the model's performance on the Google dataset, as reflected in the evaluation metrics, shows that it is capable of providing accurate and reliable stock price predictions. The high R2 score and low error metrics suggest that the LSTM model is well-suited for stock market prediction tasks.

Also, we have tried another stock market prediction model, we are using our Tesla dataset for making a **CNN-LSTM** model. It starts by loading the historical stock data using the `yfinance` library, focusing on the closing prices. The data is then normalized using `MinMaxScaler` to fit within a range of 0 to 1. The dataset is split into training (80%) and testing (20%) sets, with feature sets created using a 60-day look-back period. The model is defined using a Sequential model in TensorFlow, which includes a Conv1D layer, MaxPooling1D layer, LSTM layer, Dropout layer, and a Dense output layer. The model is compiled with the Adam optimizer and mean squared error loss function. After training for 50 epochs with a batch size of 32, the training and validation loss are plotted to visualize the model's performance. Predictions on the test set are then made and inverse transformed to the original scale. Finally, the predictions are plotted against the actual stock prices, and performance metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) score are calculated to evaluate the model's accuracy that can be seen below:

Table 5: MAE, MSE & R2 Score of CNN-LSTM model using Tesla dataset

MAE	129.93
MSE	9.43
R2 SCORE	92%

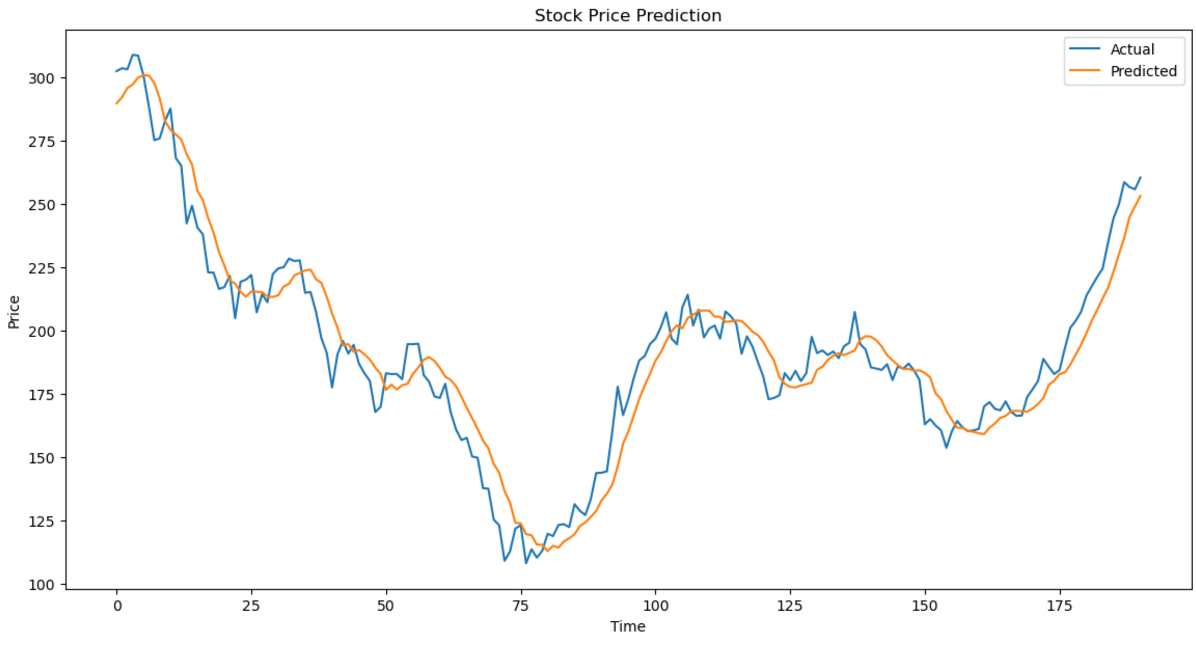


Figure 19: Prediction using CNN-LSTM model

The plot comparing the actual stock prices with the predicted values further illustrates the model's performance. The blue line represents the actual stock prices, while the orange line shows the predicted prices. The close alignment between these two lines across most of the time period under study indicates that the CNN-LSTM model effectively captures the overall trend and fluctuations in stock prices. However, minor deviations can be observed, particularly at certain peaks and troughs, which suggests areas for potential improvement in the model.

In this project, we initially developed an LSTM model and subsequently implemented a CNN-LSTM model. Through rigorous testing and evaluation, we found that the CNN-LSTM model consistently outperformed the LSTM model in terms of accuracy and error metrics. The incorporation of convolutional layers in the CNN-LSTM model effectively captured local patterns and trends in the stock data, leading to more accurate predictions and lower error values. This improvement underscores the advantage of combining convolutional and recurrent layers for time series forecasting tasks like stock market prediction.

Another model implements a **polynomial regression model** to predict Tesla's stock prices using the `yfinance` library where dataset is split into training (80%) and testing (20%) sets. A polynomial regression model of degree 3 is created using a pipeline that includes `PolynomialFeatures` and `LinearRegression`. The model is trained on the training data and then used to predict both the training and testing data. The model's performance is evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) score for both the training and testing sets that can be seen below:

Table 6: MAE, MSE & R2 Score of Polynomial regression model using Tesla dataset

MAE	74387.41
MSE	255.64
R2 SCORE	-27.43

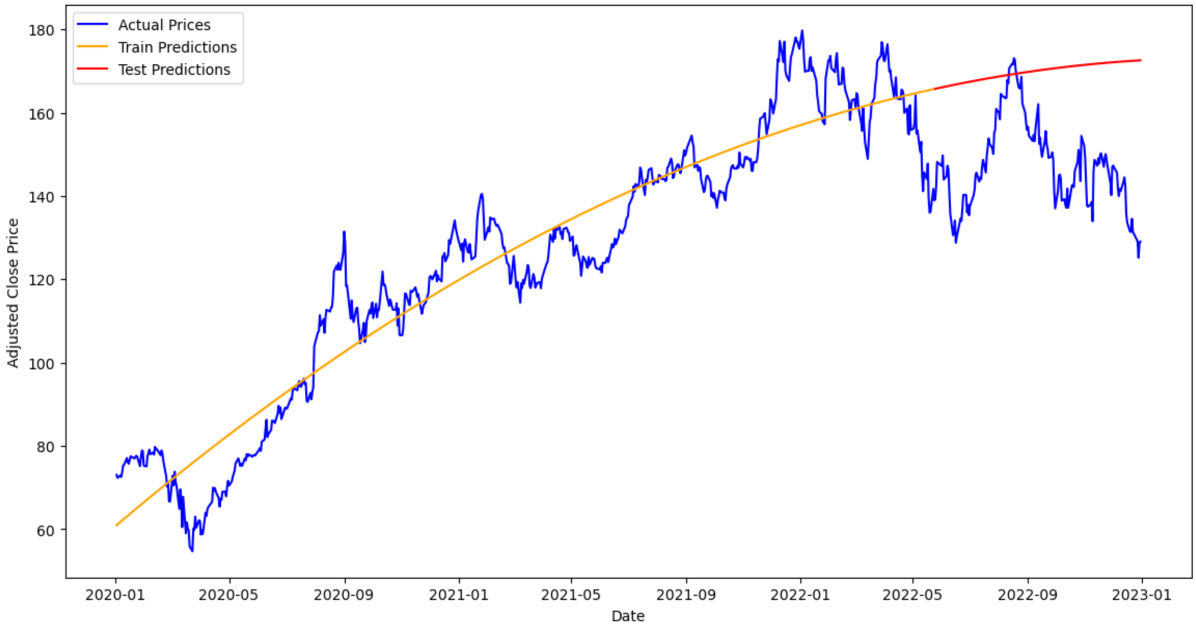


Figure 20: Polynomial regression model of degree 3

Finally, the actual stock prices and the model's predictions are plotted to visualize the results, showing the actual prices in blue, training predictions in orange, and testing predictions in red. The plot provides a visual representation of how well the model captures the trends in Tesla's stock prices.

After comparison, the LSTM model demonstrated superior performance with more accurate predictions and lower error values. The polynomial model, despite its simplicity, failed to capture the complex patterns inherent in stock market data, leading to poorer prediction results and higher error values. This comparison highlights the effectiveness of LSTM models in handling sequential data and making accurate time series predictions. While implementing the stock prices prediction using Polynomial regression, we can see the prediction in the visualization which is not as accurate that LSTM provide us the result (Fig 16: LSTM Prediction Visualization on the Dataset).

Overall, these results demonstrate that our LSTM model provides a robust framework for stock market prediction, accurately tracking the complex patterns in stock price movements. This predictive capability can be valuable for investors and analysts in making informed decisions based on anticipated market trends. Further refinement and tuning of the model may enhance its accuracy and reliability even further.

5.1 Future Prediction:

In this project, we are focusing on predicting the future stock prices of Tesla for the next 30 days which is shown below.

Figure 21: Predicting the future stock prices for the next 30 days.

The graph in **Figure 21** shown illustrates the results of the stock market prediction using an LSTM model, specifically focusing on predicting the future stock prices of Tesla for the next 30 days. The model was trained on historical stock price data where the blue line shows the actual prices, indicating how the stock price varied historically. The observed trend is a crucial input for the model to learn the patterns in stock price movement and the red line shows the predicted prices, which continue from the end of the actual price line, illustrating the model's forecast for the next 30 days. This visualization helps in understanding how well the model captures the trend and seasonality of the stock prices.

The future predictions were made by using the last 80 days of the input data. The process involved reshaping the input data to fit the LSTM model's expected input shape. The model then predicted the next day's closing price, which was appended to the input data, and the process was repeated for 30 iterations to generate the 30-day forecast. The predicted values were scaled back to the original price scale for accurate visualization. If the red line closely follows the trajectory of the blue line, it suggests that the model has learned the patterns in the data well and is making reliable predictions.

5.2 Streamlit App Development:

With the models trained and evaluated, I transitioned to building the Streamlit app for stock price forecasting. Leveraging Streamlit's intuitive API, I created a user-friendly interface with interactive widgets for data input and model selection. I incorporated Plotly for data visualization, allowing users to explore forecasted prices through interactive plots.

Once the app was developed, I deployed it to a hosting platform to make it accessible to users. Before making it live, I conducted rigorous testing to ensure that the app functioned smoothly across different devices and browsers. I also performed stress testing to gauge its performance under varying levels of user traffic.

5.2.1 Evaluation Metrics and Future Prediction

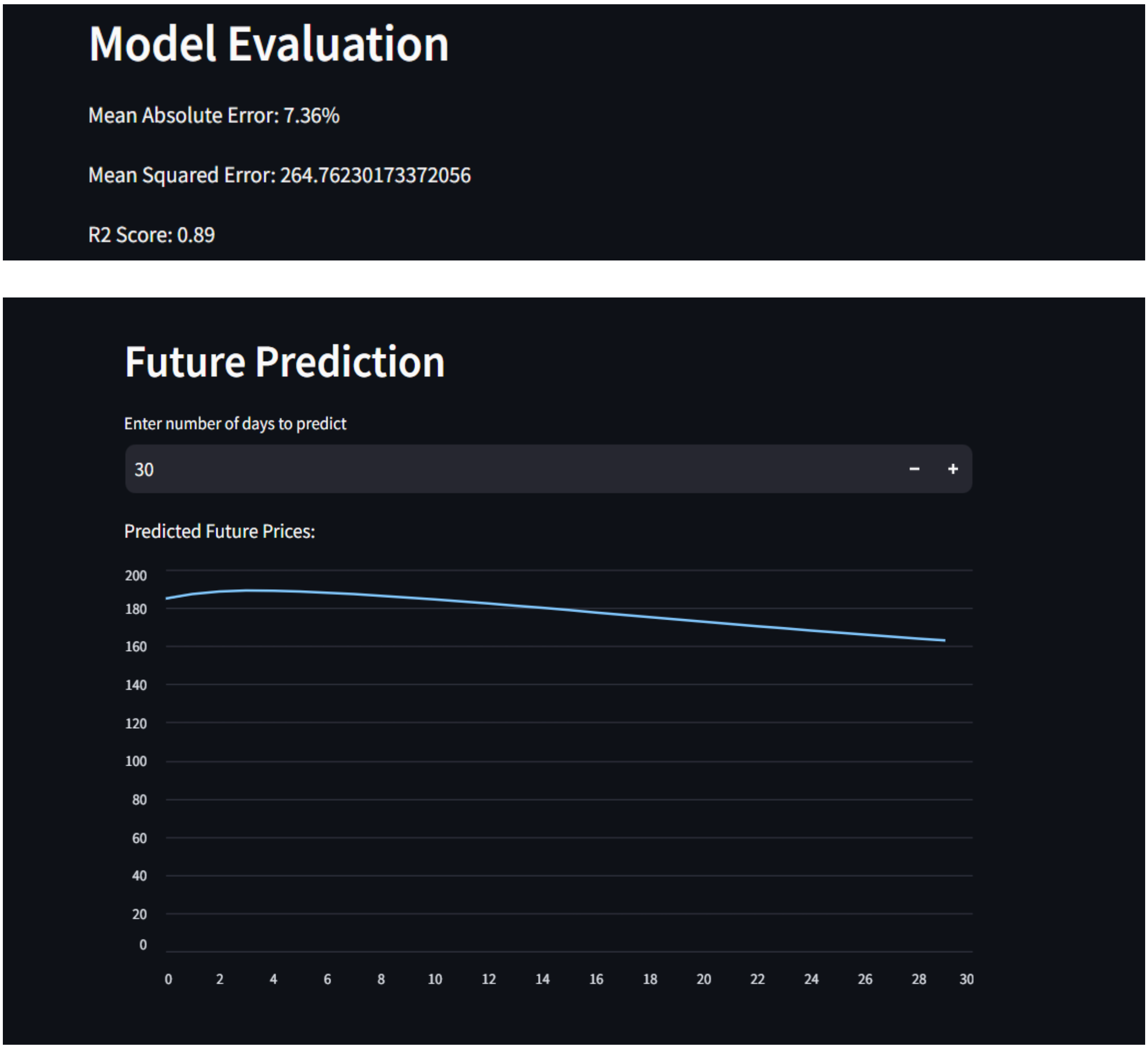


Figure 22: Evaluation Metrics & Future Prediction.

In addition to generating predictions for future stock prices, the app offers future prediction forecast based on the selected model and historical data. This feature allows users to anticipate market trends and potential investment opportunities.

Future Price Forecast: The app provides a forecast of future stock prices based on historical data and selected models. Users have to enter the number of days to predict.

Dynamic Forecasting: Forecasted prices are updated in real-time based on changes in historical data and model selection.

Building a stock price forecasting app with Streamlit was an enriching experience that combined elements of data science, machine learning, and web development. By leveraging Streamlit's simplicity and flexibility, I was able to create a powerful yet accessible tool for investors and traders to analyze historical stock data, predict future prices, and make informed investment decisions. Through continuous iteration and refinement, I'm confident that this app will continue to evolve and serve as a valuable resource for the financial community.

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this study, we explored the application of Long Short-Term Memory (LSTM) networks for stock market analysis and prediction, focusing on Tesla. Our research demonstrated the potential of LSTM models in capturing complex patterns and trends in stock price movements, leading to reasonably accurate predictions of future prices.

The findings of this study hold significant implications for both academia and the finance industry. For researchers and data scientists, the successful application of LSTM models in stock market prediction offers a promising avenue for further exploration and refinement of machine learning techniques in financial analysis. For investors and traders, the insights gained from our model could aid in making more informed decisions, potentially leading to improved investment strategies and portfolio management.

The conclusion drawn from the project involves evaluating the effectiveness of the LSTM model in capturing temporal dependencies and predicting stock price trends. The visual comparison of predicted and actual values provides insights into the model's performance and potential areas for improvement. Further analysis, including additional features and fine-tuning of hyperparameters, could enhance the model's accuracy and applicability in real-world stock market scenarios.

Our study underscores the potential of LSTM networks in stock market analysis and prediction, offering valuable insights for both researchers and practitioners in the field of finance. As machine learning continues to evolve, its application in financial markets holds the promise of unlocking new opportunities for understanding and predicting stock price movements.

6.2 Future Work

In the future, we plan to enhance our model by incorporating unstructured textual data, such as investor sentiment from social media, earnings reports, policy news, and market analysts' research reports. Additionally, we aim to develop hybrid predictive models by combining LSTM with other neural network architectures. To further improve prediction accuracy, we will implement hybrid optimization algorithms that blend local optimizers with global optimizers like genetic algorithms and particle swarm optimization. We also plan to extend our work to sentiment analysis, using data from platforms like Facebook and Twitter to understand market sentiment regarding price changes for specific stocks.

REFERENCES

[1] Shah, D., Isah, H. and Zulkernine, F., 2019. Stock market analysis: A review and taxonomy of prediction techniques. International Journal of Financial Studies, 7(2), p.26.

[2] Tsai, C.-F.; Hsiao, Y.-C. Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. Decis. Support Syst. 2010, 50, 258–269.

[3] Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). Stock price prediction using LSTM, RNN and CNN-sliding window model. In 2017 international conference on advances in computing, communications and informatics (icacci) (pp. 1643-1647). IEEE.

[4] Khadjeh Nassirtoussi, A.; Aghabozorgi, S.; Wah, T.Y.; Ngo, D. Text mining of news-headlines for FOREX market prediction: A multi-layer dimension reduction algorithm with semantics and sentiment. Expert Syst. Appl. 2015, 42, 306–324.

[5] Masoud, Najeb MH. (2017) “The impact of stock market performance upon economic growth.” International Journal of Economics and Financial Issues 3 (4) : 788–798.

[6] Hur, Jung, Manoj Raj, and Yohanes E. Riyanto. (2006) “Finance and trade: A cross-country empirical analysis on the impact of financial development and asset tangibility on international trade.” World Development 34 (10) : 1728-1741.

[7] Sirignano, J.; Cont, R. Universal features of price formation in financial markets: Perspectives from deep learning. Quant. Financ. 2019, 19, 1449–1459.

[8] Soni, P., Tewari, Y., & Krishnan, D. (2022). Machine Learning approaches in stock price prediction: A systematic review. In Journal of Physics: Conference Series (Vol. 2161, No. 1, p. 012065). IOP Publishing.

[9] Li, Lei, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, and Daoxin Chen. (2017) “Research on machine learning algorithms and feature extraction for time series.” IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC): 1-5.

[10] Seber, George AF and Lee, Alan J. (2012) “Linear regression analysis.” John Wiley & Sons 329.

[11] Reichek, Nathaniel, and Richard B. Devereux. (1982) “Reliable estimation of peak left ventricular systolic pressure by M-mode echographicdetermined end-diastolic relative wall thickness: identification of severe valvular aortic stenosis in adult patients.” American heart journal 103 (2): 202-209.

[12] Chong, Terence Tai-Leung, and Wing-Kam Ng. (2008) “Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30.” Applied Economics Letters 15 (14) : 1111-1114.

[13] Zhang, G. Peter. (2003) “Time series forecasting using a hybrid ARIMA and neural network mode.” Neurocomputing 50 : 159-175.

[14] Liaw, Andy, and Matthew Wiener. (2002) “Classification and regression by Random Forest.” R news 2 (3) : 18-22.

[15] Oyeyemi, Elijah O., Lee-Anne McKinnell, and Allon WV Poole. (2007) “Neural network-based prediction techniques for global modeling of M (3000) F2 ionospheric parameter.” Advances in Space Research 39 (5) : 643-650.

[16] Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. Procedia computer science, 170, 1168-1173.

[17] Jordan Prosky, Andrew Tan, Xingyou Song, Micael Zhao. "Sentiment Predictability for Stocks" - arXiv:1712.05785v2 [cs.CL], 18 Jan 2018.

[18] Yadav, A. and Vishwakarma, D.K., 2020. Sentiment analysis using deep learning architectures: a review. Artificial Intelligence Review, 53(6), pp.4335-4385.

[19] Mehtab, S. and Sen, J.: A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing. In: Proc. of the 7th Int. Conf. on Business Analytics and Intelligence, Bangalore, India, December 5 – 7 (2019).

[20] Malav Shastri, Sudipta Roy, & Mamta Mittal. Stock Price Prediction using Artificial Neural Model: An Application of Big Data. Icst Transactions on Scalable Information Systems. 2018; pp.6(20), 156085.

[21] Kara, Y., Acar Boyacioglu, M., & Ömer Kaan Baykan (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange. Expert Systems with Applications, 38(5), 5311–5319.

[22] P. Mondal, L. Shit, S. Goswami, Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices, Int. J. Comput. Sci. Eng. Appl. 4 (2) (2014) 13.

[23] Kilimci, Z. H., & Akyokus, S. The analysis of text categorization represented with word embeddings using homogeneous classifiers. 2019 IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA). IEEE. (2019).

[24] Samarawickrama, A.J.P.; Fernando, T.G.I. A Recurrent Neural Network Approach in Predicting Daily Stock Prices. In Proceedings of the 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 15–16 December 2017; pp.1–6.

[25] Hiransha, M. E. A. G., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE stock market prediction using deep-learning models. Procedia computer science, 132, 1351-1362.

[26] Gao, P., Zhang, R., & Yang, X. (2020). The application of stock index price prediction with neural network. Mathematical and Computational Applications, 25(3), 53.

[27] Kusuma RMI, Ho TT, Kao WC, Ou YY and Hua KL, "Using deep learning neural networks and candlestick chart representation to predict stock market," arXiv preprint arXiv:1903.12258, 2019.

[28] Hoseinzade E, Haratizadeh S (2019) CNNpred: CNN-based stock market prediction using a diverse set of variables. Expert Syst Appl 129:273–285.

[29] Kusuma RMI, Ho TT, Kao WC, Ou YY and Hua KL, "Using deep learning neural networks and candlestick chart representation to predict stock market," arXiv preprint arXiv:1903.12258, 2019.

[30] Chen, K., Zhou, Y., & Dai, F. (2015). A LSTM-based method for stock returns prediction: A case study of China stock market. In 2015 IEEE international conference on big data (Big Data) (pp. 2823–2824).

[31] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," PloS one, vol. 14, no. 2, p. e0212320, April 2019.

[32] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena.“Stock Market Prediction Using Machine Learning”.2018.IEEE

[33] Z. Hu, W. Liu, J. Bian, X. Liu, and T. Y. Liu, 2018, February. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction, 2018.

[34] https://www.kaggle.com/datasets/jimschacko/tesla-inc-last-5-years-stock-historical-data?select=TSLA+training.csv

[35] https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data