Stock Prediction & Visualizer

A project report submitted in partial fulfillment of the
requirements for the award of the degree of

Master of Computer Applications

In

Computer Applications

By

Yashraj Jain (205121107)

DEPARTMENT OF COMPUTER APPLICATIONS

NATIONAL INSTITUTE OF TECHNOLOGY,
TIRUCHIRAPPALLI 620015

JUNE 2024

BONAFIDE CERTIFICATE

This is to certify that the project **"Stock prediction & Visualizer"** is a project work successfully done by

Yashraj Jain (205121107)

in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications from the National Institute of Technology, Tiruchirappalli, during the academic year 2023-2024 (6th Semester – CA750 Major Project Work).

Dr. U. Srinivasulu Reddy	Dr. Michael Arock
Internal Guide	**Head of the Department**

Project viva-voce held on …………………………….

Internal Examiner	**External Examiner**

Acknowledgment

Every project, big or small, is successful largely due to the effort of a number of wonderful people who have always given their valuable advice or lent a helping hand. I sincerely appreciate the inspiration, support, and guidance of all those people who have been instrumental in making this project successful.

We express our deep sense of gratitude to **Dr. G. Aghila**, Director, National Institute of Technology, Tiruchirappalli for giving us an opportunity to do this project.

I am grateful to **Dr. Michael Arock**, Professor, and Head of the Department of Computer Applications, National Institute of Technology, Tiruchirappalli for providing the infrastructure and facilities to carry out the project.

I express my gratitude to my Project Guide **Dr. U. Srinivasulu Reddy**, Associate Professor, Department of Computer Applications, National Institute of Technology, Tiruchirappalli for his support and for arranging the project in a good schedule, and who assisted me in completing the project. I would like to thank him for duly evaluating my progress and evaluating me.

I express my sincere and heartfelt gratitude to **Project Evaluation Committee**, Department of Computer Applications, National Institute of Technology, Tiruchirappalli. I am sincerely thankful for his constant support, care, guidance, and regular interaction throughout my project.

I express my sincere thanks to all the faculty members, and scholars of NIT Trichy for their critical advice and guidance to develop this project directly or indirectly.

Abstract

This project centres on the development and implementation of a stock prediction and visualization system using Long Short-Term Memory (LSTM) neural networks. The system employs the Adam optimizer for training and Root Mean Square Error (RMSE) for performance evaluation. Additionally, the project integrates the Tiingo API for data acquisition and preprocessing, specifically using pandas data reader for importing historical stock data and MinMaxScaler for data normalization.

LSTM networks, a subset of recurrent neural networks (RNNs), are adept at handling time series data, making them suitable for stock price prediction. The model was trained on historical stock prices obtained through the Tiingo API, which facilitated seamless and efficient data retrieval. This data was then preprocessed using pandas data reader for organizing the data and MinMaxScaler for normalizing the stock prices. Normalization is a crucial step as it scales the data to a range, enhancing the performance and convergence speed of the LSTM model.

The Adam optimizer was selected for training the LSTM due to its adaptive learning rate properties, which combine the benefits of the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). This optimizer improves convergence speed and model performance by dynamically adjusting the learning rate throughout the training process.

Root Mean Square Error (RMSE) was utilized as the primary metric for evaluating the model's accuracy. RMSE measures the average magnitude of the error between the predicted values and the actual values, with lower values indicating better predictive accuracy.

The visualization component of the project involved plotting the historical stock prices alongside the model's predictions. This visual comparison aids in understanding the model's effectiveness and the trends in stock price movements. Matplotlib and Seaborn libraries in Python were used to create static and interactive visualizations, providing clear and informative insights into the model's performance.

The results indicated that the LSTM model, optimized with Adam and supplemented with MinMaxScaler normalization, effectively captured the underlying trends and patterns in the stock price data. The model achieved relatively low RMSE values, demonstrating high predictive accuracy. The visualizations offered an intuitive means of assessing the model's predictions, making the results easily interpretable.

This project highlights the potential of LSTM networks and advanced optimization techniques in financial forecasting. Future enhancements could include extending the model to predict prices for multiple stocks concurrently, incorporating additional market indicators and features, and exploring ensemble methods to further improve prediction accuracy. The goal is to develop a robust and user-friendly tool that aids investors and financial analysts in making informed decisions through predictive analytics.

Table of Contents

Table Of Figures

1. Introduction

1.1 Definition and History:

Stock Prediction: Stock prediction involves forecasting future stock prices based on historical data and various analytical methods. It utilizes statistical models and machine learning techniques, such as Long Short-Term Memory (LSTM) networks, to identify patterns and trends within stock market data. The goal of stock prediction is to provide investors and traders with insights that can aid in making informed decisions about buying or selling stocks.

Visualizer: A visualizer in the context of stock prediction is a tool or system that graphically represents stock data and prediction results. It uses charts and graphs to display historical stock prices alongside predicted prices, allowing users to easily interpret and compare the data. Visualization tools enhance the understanding of model performance and the accuracy of predictions, making the analysis more accessible and actionable.

1.2 Evolution of Stock Prediction Technology:

Below is an overview of the major milestones in the evolution of stock prediction and forecasting:

Early Methods (Pre-1970s):
Fundamental Analysis: Initially, stock prediction relied heavily on fundamental analysis, which involved evaluating a company's financial statements, management, competitive advantages, and market conditions to estimate its intrinsic value.
Technical Analysis: Concurrently, technical analysis emerged, focusing on historical price movements and trading volumes to identify patterns and trends. Analysts used charts and indicators like moving averages and the relative strength index (RSI) to make predictions.

Statistical Methods (1970s-1980s):
Time Series Analysis: The 1970s saw the introduction of time series analysis techniques, such as autoregressive integrated moving average (ARIMA) models. These models used historical data to predict future values based on identified patterns in the time series.

Efficient Market Hypothesis (EMH): During this period, the Efficient Market Hypothesis, proposed by Eugene Fama, gained prominence. It suggested that stock prices fully reflect all available information, making it impossible to consistently achieve higher returns through prediction.

2. Literature Review

Early research on Stock Market Prediction was based on Random Walk and Efficient Market Hypothesis (EMH). Numerous studies like Gallagher, Kavussanos, Butler, show that stock market prices do not follow a random walk and can be predicted to an extent Another hypothesis which is currently under survey is, whether the early indicators extracted from online sources (blogs, twitter feeds etc.) can be used to predict changes in economic and commercial indicators. Analysis of the same has been done in other fields of research, for e.g. Gruhl et al showed the correlation between online chat activity and book sales [2]. Blog sentiment assessment has been used to predict movie sales by Mishne, Glance et al [6]. Schumaker et al investigated the relations between breaking financial news and stock price changes [1]. One of the major researches in the field of stock prediction was carried out by Bollen, Mao et al 2011, where they investigated correlation between public mood and Dow Jones Industrial Index. Public moods (Happy, Calm, and Anxiety) were derived using twitter feed [2]. Chen and Lazer derived investment strategies by observing and classifying the twitter feeds [4]. Bing et al, studied the twitter feed and concluded predictability of stock based on the industry [1]. Zhang et al found high negative correlation between negative moods on social network and DJIA index. Pagolu et al in their work showed a strong correlation between rise/fall of a company stock prices and the public emotions expressed on twitter. Instead of using a standard word embedding model, their work focused on developing a sentiment analyzer to categorize tweets in three categories: Positive, Negative and Neutral. Mittal et al in their study tried to build a portfolio management tool using the twitter sentiment analysis. They analyzed and tested their model on DJIA. The model based on greedy strategy received feedback from sentiment analysis of the social media to predict the Buy/Sell decisions for the DJIA positions, one day in advance. Chen et al used a model established on LSTM algorithm to predict direction of stocks in Chinese Stock Exchange, in their study, they compared LSTM with Random estimation model confirming higher accuracy for the LSTM model [4]. Study by Tekin et al analysed the data of the 25 leading companies and applied various forecasting models. Their studies showed a higher relevance of Random Forest technique. LSTM multi-layer perceptron (MLP) and random forest classifier are employed by Malandri et al in their Portfolio allocation model. A study of NYSE data suggests that LSTM gives better experimental results . Kilimci et al, 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) [5]. 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.

3. How Stock Prediction Works ?

3.1 Stock Prediction Algorithm:

Stock prediction algorithms use a combination of statistical techniques and machine learning models to forecast future stock prices based on historical data. Here's a detailed step-by-step explanation of how these algorithms typically work:

3.2 Data collection and Preprocessing:

The first step is gathering relevant data. This includes:

Historical Stock Prices: Daily, weekly, or monthly stock prices (open, high, low, close, and volume).

Market Indicators: Indices like S&P 500, Dow Jones Industrial Average.

Company Financial Data: Earnings reports, balance sheets, etc.

External Factors: News articles, social media sentiment, economic indicators.

Raw data is rarely in a form suitable for direct use in algorithms. Preprocessing steps include:

Normalization: Scaling data to a specific range (e.g., [0, 1]) using techniques like MinMaxScaler.

Feature Engineering: Creating new features or selecting relevant ones to improve model performance.

3.3 Training and Prediction:

Training the chosen model involves several steps:

Data Splitting: Dividing the data into training, validation, and test sets to evaluate model performance.

Hyperparameter Tuning: Adjusting model parameters to improve accuracy and avoid overfitting.

Optimization: Using optimization algorithms like Adam, which adjusts learning rates during training for faster and more efficient convergence.

Once the model is trained and validated, it is used to make predictions on new, unseen data:

Future Price Prediction: Using the model to forecast future stock prices based on the latest available data.

4. Applications Of Stock Prediction Technology

Scenario: Investment Portfolio Management

Background:

John is a portfolio manager responsible for managing investments on behalf of his clients. His primary goal is to maximize returns while minimizing risks by strategically allocating funds across various stocks and assets.

Application:

4.1 Data Gathering and Preprocessing:

John utilizes stock prediction and visualization systems to gather historical stock prices, market indicators, and economic data from multiple sources, including financial APIs and databases.

The data is preprocessed to handle missing values, normalize prices, and engineer relevant features such as moving averages, trading volumes, and sentiment scores.

John selects a suitable predictive model, such as a Long Short-Term Memory (LSTM) network, known for its effectiveness in capturing temporal dependencies in time series data.

The chosen model is trained on the preprocessed data using advanced optimization techniques like the Adam optimizer to forecast future stock prices with high accuracy.

4.2 Risk Assessment and Portfolio Optimization:

John leverages the predictions generated by the stock prediction system to assess the risk associated with each investment opportunity.

He employs modern portfolio theory and optimization algorithms to construct diversified portfolios that balance risk and return based on the predicted stock price movements.

Visual Representation and Decision Support:

The visualizer component of the system generates interactive dashboards and graphical representations of historical and predicted stock prices.

John uses these visualizations to monitor portfolio performance, identify emerging trends, and make timely investment decisions. He can visualize stock price movements, compare predicted vs. actual prices, and assess the impact of external factors on market dynamics.

4.3 Portfolio Adjustment and Rebalancing:

Based on the insights gleaned from the stock prediction and visualization system, John adjusts and rebalances his investment portfolios as needed.

He reallocates funds among different assets, sectors, and geographic regions to capitalize on opportunities and mitigate risks in response to changing market conditions.

Benefits:

Improved Decision Making: By leveraging predictive analytics and visualization tools, John can make data-driven investment decisions with greater confidence and precision.

Enhanced Performance: The application of stock prediction algorithms helps John identify alpha-generating opportunities and optimize portfolio performance over time.

Risk Management: The system enables John to assess and manage investment risks effectively by providing insights into potential downside scenarios and portfolio diversification strategies.

Client Satisfaction: By delivering superior returns and minimizing downside risk, John enhances client satisfaction and builds long-term relationships based on trust and transparency.

4.4 Conclusion:

The implementation of stock prediction and visualization systems marks a significant advancement in financial analytics and investment management. These systems leverage the power of machine learning algorithms, particularly Long Short-Term Memory (LSTM) networks, alongside sophisticated data visualization tools to provide accurate and actionable insights into future stock price movements. This comprehensive approach enhances predictive accuracy and facilitates better decision-making through intuitive visual representations. Key benefits include improved predictive accuracy, enhanced decision-making capabilities, effective risk management, and optimized portfolio performance. The systems' ability to present complex data in user-friendly visual formats empowers investors and portfolio managers to make informed decisions, dynamically adjust holdings, and develop robust risk mitigation strategies. Applications extend to institutional and retail investing, algorithmic trading, and financial advisory services, demonstrating their broad utility. Overall, stock prediction and visualization systems represent a transformative tool in the financial industry, driving better investment outcomes and shaping the future of financial decision-making.

5. Dataset

The acquisition of high-quality financial data forms the cornerstone of any successful stock prediction and visualization endeavor. In this section, we delve into the process of gathering pertinent dataset from the Tiingo API utilizing the versatile capabilities of the Pandas Data Reader library.

Tiingo API Integration:

The Tiingo API serves as a valuable resource for accessing a wide array of financial data, including historical stock prices, market indices, and economic indicators. Leveraging this API provides researchers and practitioners with a comprehensive dataset crucial for training predictive models and performing insightful analysis.

Pandas Data Reader:

Pandas Data Reader, an integral component of the Python ecosystem, offers seamless integration with the Tiingo API. This powerful library simplifies the process of fetching financial data by providing a convenient interface for querying and retrieving data directly into Pandas data structures.

Data Retrieval Process:

Utilizing Pandas Data Reader, researchers can easily specify the desired parameters, such as stock symbols, date ranges, and data sources, to retrieve relevant financial data from the Tiingo API. This streamlined process facilitates efficient data acquisition and minimizes the overhead associated with manual data collection.

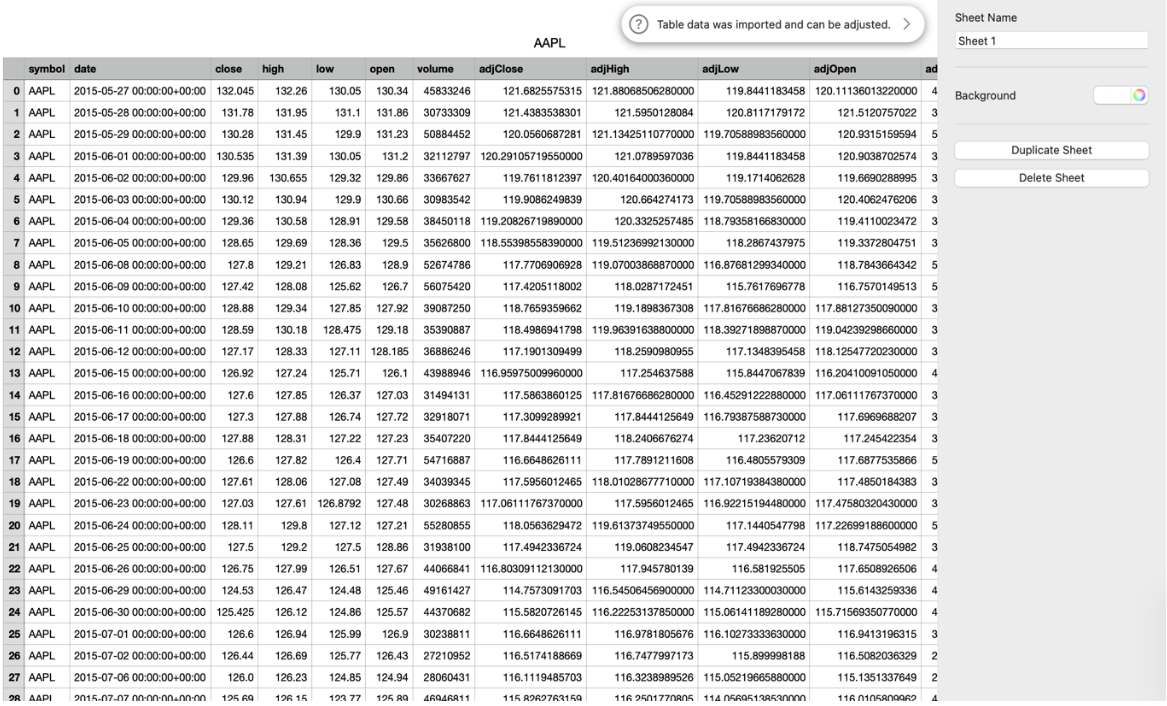


Table data was imported and can be adjusted.

Sheet Name: Sheet 1

AAPL

	symbol	date	close	high	low	open	volume	adjClose	adjHigh	adjLow	adjOpen	ad
0	AAPL	2015-05-27 00:00:00+00:00	132.045	132.26	130.05	130.34	45833246	121.6825575315	121.88068506280000	119.8441183458	120.11136013220000	4
1	AAPL	2015-05-28 00:00:00+00:00	131.78	131.95	131.1	131.86	30733309	121.4383538301	121.5950128084	120.8117179172	121.5120757022	3
2	AAPL	2015-05-29 00:00:00+00:00	130.28	131.45	129.9	131.23	50884452	120.0560687281	121.13425110770000	119.70588983560000	120.9315159594	5
3	AAPL	2015-06-01 00:00:00+00:00	130.535	131.39	130.05	131.2	32112797	120.29105719550000	121.0789597036	119.8441183458	120.9038702574	3
4	AAPL	2015-06-02 00:00:00+00:00	129.96	130.655	129.32	129.86	33667627	119.7611812397	120.40164000360000	119.1714062628	119.6690288995	3
5	AAPL	2015-06-03 00:00:00+00:00	130.12	130.94	129.9	130.66	30983542	119.9086249839	120.664274173	119.70588983560000	120.4062476206	3
6	AAPL	2015-06-04 00:00:00+00:00	129.36	130.58	128.91	129.58	38450118	119.20826719890000	120.3325257485	118.79358166830000	119.4110023472	3
7	AAPL	2015-06-05 00:00:00+00:00	128.65	129.69	128.36	129.5	35626800	118.55398558390000	119.51236992130000	118.2867437975	119.3372804751	3
8	AAPL	2015-06-08 00:00:00+00:00	127.8	129.21	126.83	128.9	52674786	117.7706906928	119.07003868870000	116.87681299340000	118.7843664342	5
9	AAPL	2015-06-09 00:00:00+00:00	127.42	128.08	125.62	126.7	56075420	117.4205118002	118.0287172451	115.7617696778	116.7570149513	5
10	AAPL	2015-06-10 00:00:00+00:00	128.88	129.34	127.85	127.92	39087250	118.7659359662	119.1898367308	117.81676686280000	117.88127350090000	3
11	AAPL	2015-06-11 00:00:00+00:00	128.59	130.18	128.475	129.18	35390887	118.4986941798	119.96391638800000	118.39271898870000	119.04239298660000	3
12	AAPL	2015-06-12 00:00:00+00:00	127.17	128.33	127.11	128.185	36886246	117.1901309499	118.2590980955	117.1348395458	118.12547720230000	3
13	AAPL	2015-06-15 00:00:00+00:00	126.92	127.24	125.71	126.1	43988946	116.95975009960000	117.254637588	115.8447067839	116.20410091050000	4
14	AAPL	2015-06-16 00:00:00+00:00	127.6	127.85	126.37	127.03	31494131	117.5863860125	117.81676686280000	116.45291222880000	117.06111767370000	3
15	AAPL	2015-06-17 00:00:00+00:00	127.3	127.88	126.74	127.72	32918071	117.3099289921	117.8444125649	116.79387588730000	117.6969688207	3
16	AAPL	2015-06-18 00:00:00+00:00	127.88	128.31	127.22	127.23	35407220	117.8444125649	118.2406676274	117.23620712	117.245422354	3
17	AAPL	2015-06-19 00:00:00+00:00	126.6	127.82	126.4	127.71	54716887	116.6648626111	117.7891211608	116.4805579309	117.6877535866	5
18	AAPL	2015-06-22 00:00:00+00:00	127.61	128.06	127.08	127.49	34039345	117.5956012465	118.01028677710000	117.10719384380000	117.4850184383	3
19	AAPL	2015-06-23 00:00:00+00:00	127.03	127.61	126.8792	127.48	30268863	117.06111767370000	117.5956012465	116.92215194480000	117.47580320430000	3
20	AAPL	2015-06-24 00:00:00+00:00	128.11	129.8	127.12	127.21	55280855	118.0563629472	119.61373749550000	117.1440547798	117.22699188600000	5
21	AAPL	2015-06-25 00:00:00+00:00	127.5	129.2	127.5	128.86	31938100	117.4942336724	119.0608234547	117.4942336724	118.7475054982	3
22	AAPL	2015-06-26 00:00:00+00:00	126.75	127.99	126.51	127.67	44066841	116.80309112130000	117.945780139	116.581925505	117.6508926506	4
23	AAPL	2015-06-29 00:00:00+00:00	124.53	126.47	124.48	125.46	49161427	114.7573091703	116.54506456900000	114.71123300030000	115.6143259336	4
24	AAPL	2015-06-30 00:00:00+00:00	125.425	126.12	124.86	125.57	44370682	115.5820726145	116.22253137850000	115.06141189280000	115.71569350770000	4
25	AAPL	2015-07-01 00:00:00+00:00	126.6	126.94	125.99	126.9	30238811	116.6648626111	116.9781805676	116.10273333630000	116.9413196315	3
26	AAPL	2015-07-02 00:00:00+00:00	126.44	126.69	125.77	126.43	27210952	116.5174188669	116.7477997173	115.899998188	116.5082036329	2
27	AAPL	2015-07-06 00:00:00+00:00	126.0	126.23	124.85	124.94	28060431	116.1119485703	116.3238989526	115.05219665880000	115.1351337649	2
28	AAPL	2015-07-07 00:00:00+00:00	125.69	126.15	123.77	125.89	46946811	115.8262763159	116.2501770805	114.05695138530000	116.0105809962	4

Fig 5 (a) AAPL Dataset

This dataset provides a comprehensive overview of the stock performance of Apple Inc. (AAPL) over a significant period spanning from the year 2015 to 2023. It offers detailed information on various metrics such as closing price, high price, low price, opening price, volume, adjusted closing price, adjusted high price, adjusted low price, adjusted opening price, adjusted volume, dividend cash, and split factor for each trading day within this timeframe.

Data Explanation:

Date: The date of the trading day.

Close: The closing price of AAPL stock on the respective trading day.

High: The highest price of AAPL stock reached during the trading day.

Low: The lowest price of AAPL stock reached during the trading day.

Open: The opening price of AAPL stock at the beginning of the trading day.

Volume: The total number of AAPL shares traded during the trading day.

Adjusted Close: The closing price adjusted for any corporate actions such as stock splits or dividends.

Adjusted High: The highest price adjusted for any corporate actions.

Adjusted Low: The lowest price adjusted for any corporate actions.

Adjusted Open: The opening price adjusted for any corporate actions.

Adjusted Volume: The trading volume adjusted for any corporate actions.

Dividend Cash: Any cash dividends distributed to shareholders.

Split Factor: Factor by which the stock was split, if applicable.

Conclusion:

This dataset offers valuable insights into the stock performance of Apple Inc. over the specified period. Analyzing this data can aid in understanding trends, making informed investment decisions, and conducting further research into the factors influencing AAPL's stock price during this timeframe.

6. Challenges and Risks

While stock prediction and forecasting offer valuable insights into market trends and investment opportunities, they are fraught with challenges and risks that practitioners must navigate effectively. Below are some of the key challenges and risks associated with stock prediction and forecasting:

6.1 Data Quality and Reliability:

Incomplete Data: Historical stock price data may contain missing values or gaps, which can affect the accuracy of predictions.

Data Noise: Financial markets are influenced by various external factors, leading to noisy data that may obscure underlying patterns.

Data Source Bias: Biases in data sources, such as sampling biases or survivorship biases, can distort predictions and lead to inaccurate forecasts.

6.2 Market Volatility and Uncertainty:

Non-Stationarity: Financial markets are dynamic and non-stationary, making it challenging to identify stable patterns over time.

Market Shocks: Unforeseen events, such as economic crises, geopolitical tensions, or natural disasters, can cause sudden and significant fluctuations in stock prices, rendering predictions obsolete.

Black Swan Events: Rare and unpredictable events, known as black swan events, can have a profound impact on financial markets, making them inherently difficult to forecast.

6.3 Model Complexity and Overfitting:

Model Overfitting: Complex models may capture noise in the data rather than genuine patterns, leading to overfitting and poor generalization to unseen data.

Model Interpretability: Highly complex models, such as deep learning neural networks, lack interpretability, making it challenging to understand the underlying factors driving predictions.

Assumptions and Limitations:

Efficient Market Hypothesis (EMH): The Efficient Market Hypothesis posits that stock prices reflect all available information, making it impossible to consistently outperform the market through prediction alone. This theory challenges the efficacy of stock prediction models.

Risk of False Positives/Negatives: False positive predictions may lead to missed investment opportunities, while false negatives can result in losses, highlighting the importance of balancing prediction accuracy and risk tolerance.

7. Methodologies

7.1 LSTM :

LSTMs Long Short-Term Memory is a type of RNNs Recurrent Neural Network that can detain long-term dependencies in sequential data. LSTMs can process and analyze sequential data, such as time series, text, and speech. They use a memory cell and gates to control the flow of information, allowing them to selectively retain or discard information as needed and thus avoid the vanishing gradient problem that plagues traditional RNNs. LSTMs are widely used in various applications such as natural language processing, speech recognition, and time series forecasting.

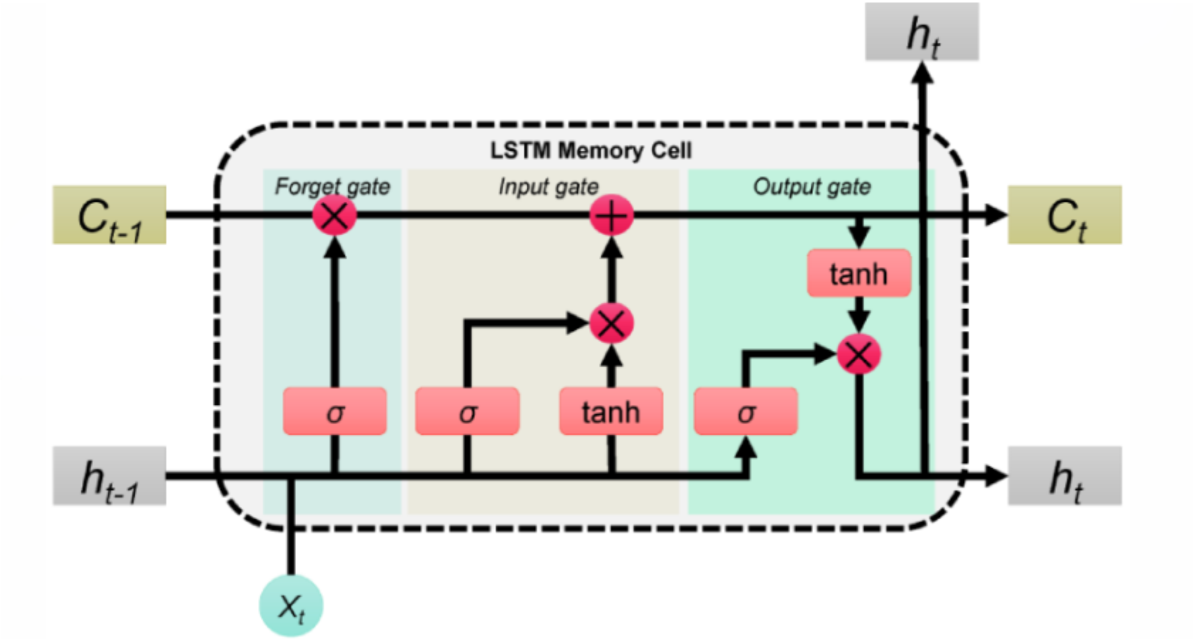


Fig 7.1 (a) LSTM Gates

There are three types of gates in an LSTM: the input gate, the forget gate, and the output gate.

The input gate controls the flow of information into the memory cell. The forget gate controls the flow of information out of the memory cell. The output gate controls the flow of information out of the LSTM and into the output.

Three gates input gate, forget gate, and output gate are all implemented using sigmoid functions, which produce an output between 0 and 1. These gates are trained using a backpropagation algorithm through the network.

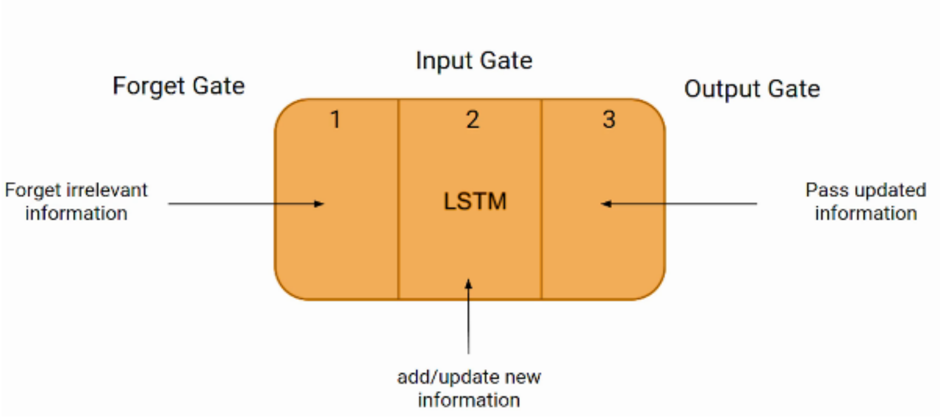


Fig 7.1 (b) Responsibilities of LSTM Gates

The input gate decides which information to store in the memory cell. It is trained to open when the input is important and close when it is not.

The forget gate decides which information to discard from the memory cell. It is trained to open when the information is no longer important and close when it is.

The output gate is responsible for deciding which information to use for the output of the LSTM. It is trained to open when the information is important and close when it is not.

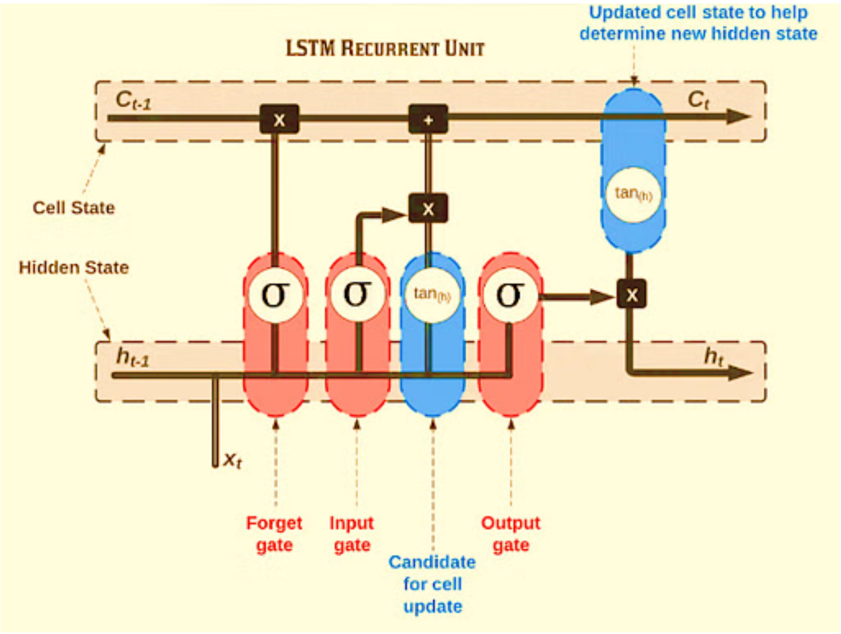


Fig 7.1 (c) Structure Of LSTM

An LSTM (Long Short-Term Memory) network is a type of RNN recurrent neural network that is capable of handling and processing sequential data. The structure of an LSTM network consists of a series of LSTM cells, each of which has a set of gates (input, output, and forget gates) that control the flow of information into and out of the cell. The gates are used to selectively forget or retain information from the previous time steps, allowing the LSTM to maintain long-term dependencies in the input data.

The LSTM cell also has a memory cell that stores information from previous time steps and uses it to influence the output of the cell at the current time step. The output of each LSTM cell is passed to the next cell in the network, allowing the LSTM to process and analyze sequential data over multiple time steps.

7.2 Adam Optimizer:

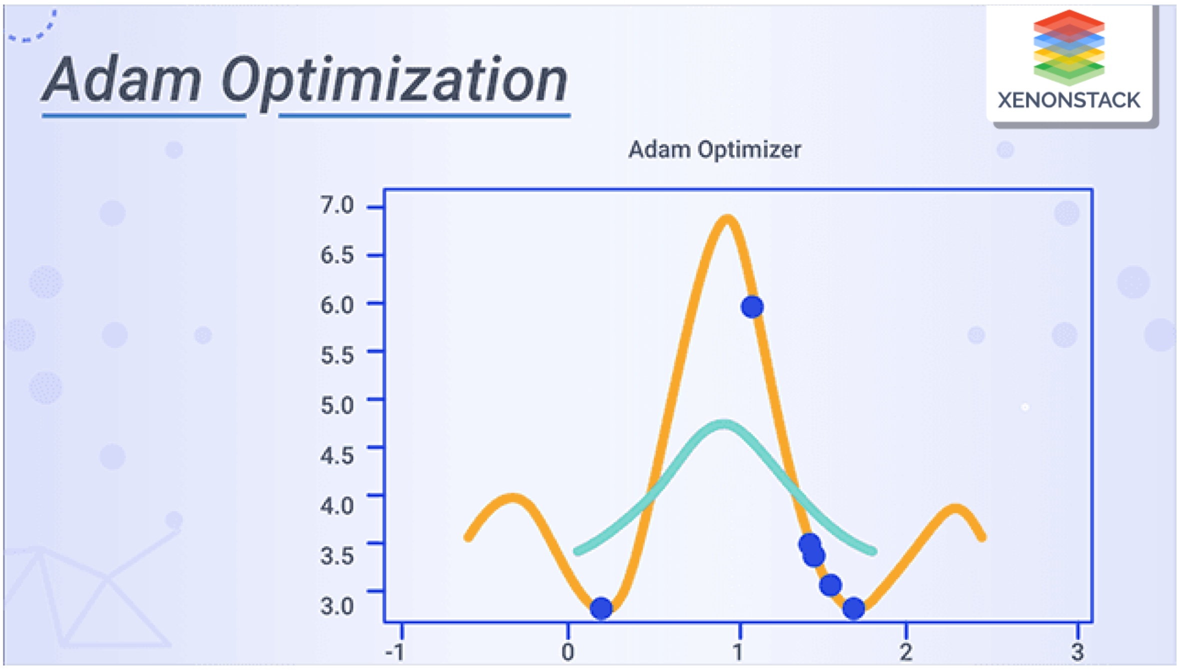


Fig 7.2 (a) Adam Optimizer

Adaptive Moment Estimation is an algorithm for optimization technique for gradient descent. The method is efficient when working with large problem involving a lot of data or parameters. It requires less memory and is efficient. Intuitively, it is a combination of the 'gradient descent with momentum' algorithm and the 'RMSP' algorithm.

Adam optimizer involves a combination of two gradient descent methodologies:

Momentum:

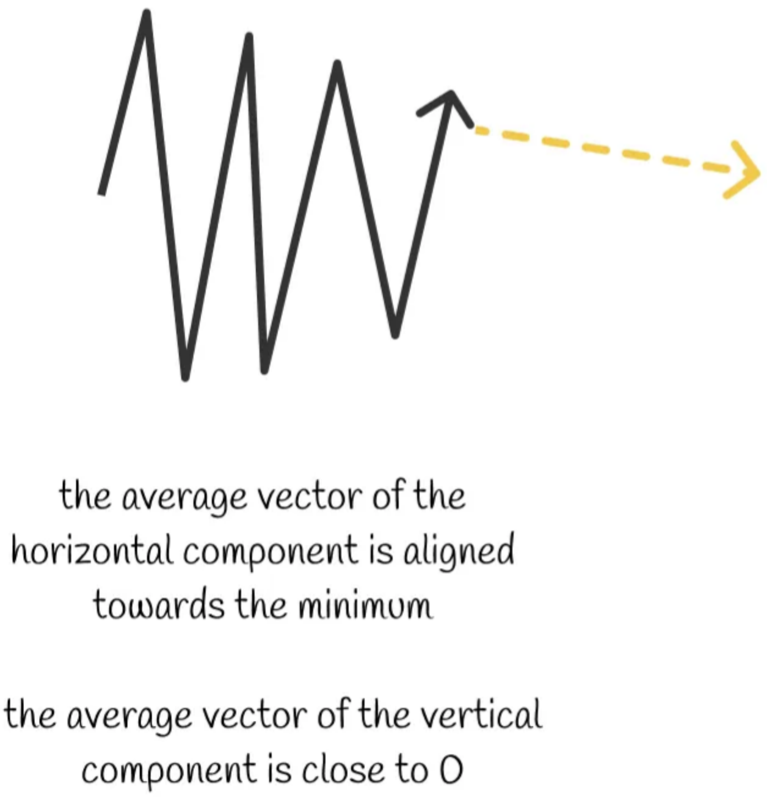

the average vector of the horizontal component is aligned towards the minimum
the average vector of the vertical component is close to 0

Fig 7.2 (b) Gradient Descent with Momentum

This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the ‘exponentially weighted average’ of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.

$$w_{t+1} = w_t - \alpha m_t$$

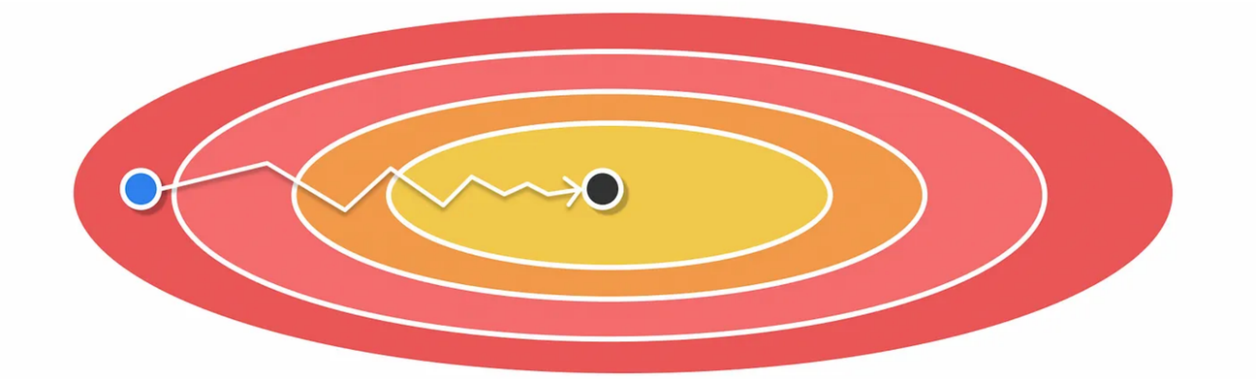
Where,

$$m_t = \beta m_{t-1} + (1-\beta) \, [\, \delta w_t \, / \, \delta L]$$

- m_t = aggregate of gradients at time t [current] (initially, mt = 0)
- m_{t-1} = aggregate of gradients at time t-1 [previous]
- W_t = weights at time t
- W_{t+1} = weights at time t+1
- α_t = learning rate at time t
- ∂L = derivative of Loss Function

- ∂W_t = derivative of weights at time t
- β= Moving average parameter (const, 0.9)

Root Mean Square Propagation (RMSP):



Optimization with RMSProp

Fig 7.2 (c) Optimization with RMSProp

Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the 'exponential moving average'.

$$w_{t+1} = w_t - (\alpha_t / (v_t + \varepsilon)^{1/2}) * [\, \delta w_t \,/\, \delta L]$$

Where,

$$v_t = \beta v_{t-1} + (1-\beta) * [\, \delta w_t \,/\, \delta L]^2$$

- W_t = weights at time t
- W_{t+1} = weights at time t+1
- α_t = learning rate at time t
- ∂L = derivative of Loss Function
- ∂W_t = derivative of weights at time t
- V_t = sum of square of past gradients.
- β = Moving average parameter (const, 0.9)
- ϵ = A small positive constant (10-8)

8. Project Implementation and Explanation

1. **Data Collection:**

Obtain historical stock price data from various sources like Yahoo Finance, Alpha Vantage, or Quandl. You'll typically need data on open, high, low, close prices (OHLC), and volume. Consider factors like dividends, stock splits, and other corporate actions that may affect the data.

2. **Data Pre-processing:**

Clean the data by handling missing values, removing outliers, and adjusting for stock splits or dividends. You might also need to normalize or scale the data to improve model performance.

3. **Feature Engineering:**

Extract relevant features from the raw data that could potentially influence stock prices. These could include technical indicators (e.g., moving averages, RSI), fundamental data (e.g., earnings per share, P/E ratio), and sentiment analysis from news or social media.

4. **Model Selection & Training:**

Choose an appropriate machine learning or deep learning model for stock prediction. Common choices include linear regression, ARIMA, LSTM, or more advanced algorithms like Gradient Boosting Machines (GBM) or Long Short-Term Memory (LSTM) networks.

Split your data into training and testing sets. Train your model on the training data and tune hyperparameters to optimize performance. Use techniques like cross-validation to prevent overfitting and ensure generalization.

5. **Model Evaluation:**

Evaluate your model's performance using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or accuracy. Compare the predicted values against actual values on the test set to assess how well your model generalizes to unseen data.

6. **Visualization:**

Develop a user-friendly visualization interface to display stock prices, predicted values, and any relevant features or indicators. Tools like Matplotlib, Plotly, or Bokeh can be used to create interactive charts and dashboards.

7. **Monitoring and Maintenance:**

Continuously monitor your model's performance in the real world and update it as needed. Markets are dynamic, so your model may need periodic retraining or adjustment to remain accurate.

9. System Design

9.1 Block Diagram of Model:

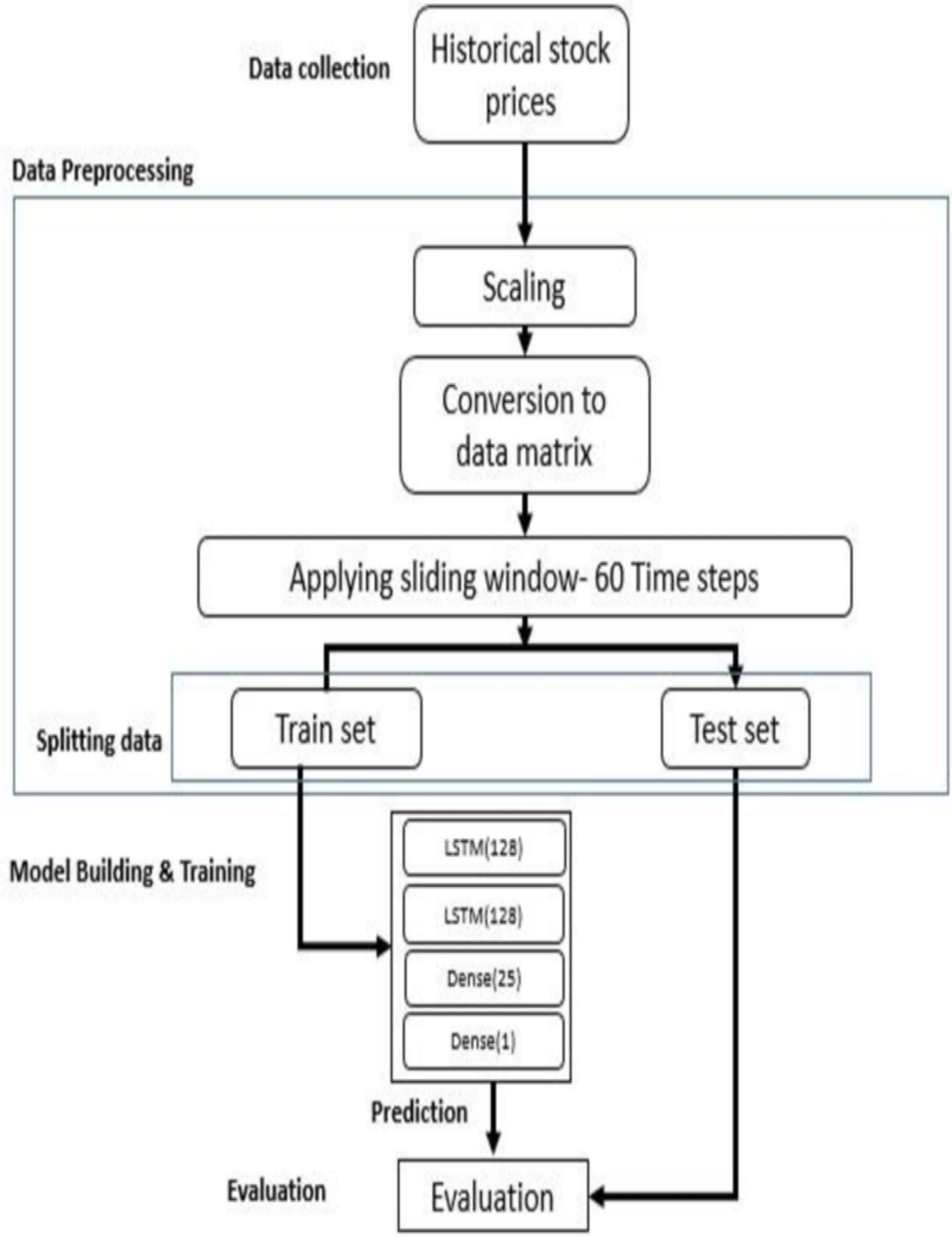


Fig 9.1 (a) Block diagram of stock prediction using LSTM

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 a result of the entire model building exercise, a sliding historical window of 60 days gave the best results among the range covered. Two layers LSTM respectively with 128 and 64 neurons followed by two dense layers of 25 and 1 neurons was the final model that gave best performance among various model variations.

Since this is a regression model, standard features like accuracy % couldn’t be used.

Thus, RMSE was used as the quantifying parameter for evaluating the success of models being tested.

9.2 Visualization:

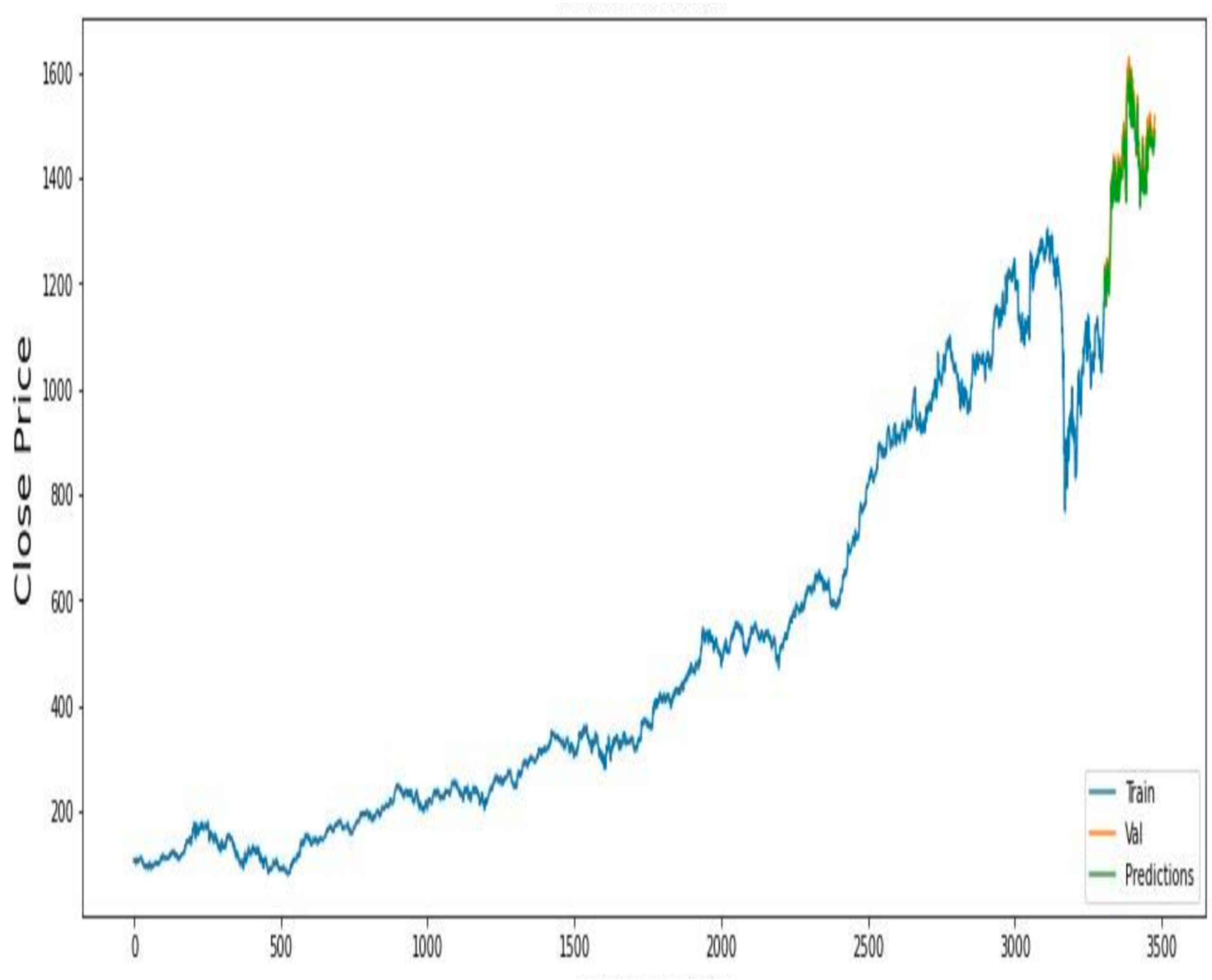


Fig 9.2 (a) Visualization of AAPL Pricing Data 2019

The graph for 2019 consists of three lines, each representing different aspects of the AAPL stock price data and model predictions:

Training Data (Blue Line)

Represents the actual AAPL stock prices for 2019 used to train the model.

Shows the historical trends and patterns the model learned from.

Validation Data (Yellow Line)

Represents the actual AAPL stock prices used for validating the model's performance.

Helps in assessing how well the model generalizes to new, unseen data.

Predicted Stock Prices (Green Line)

Represents the stock prices predicted by the model.

The alignment of the green line with the blue and yellow lines indicates the model's accuracy in capturing and predicting stock price trends.

Insight for 2019 Data

The closer the green line follows the blue and yellow lines, the better the model's performance. This visualization helps in quickly assessing the model's accuracy and reliability in predicting AAPL stock prices.

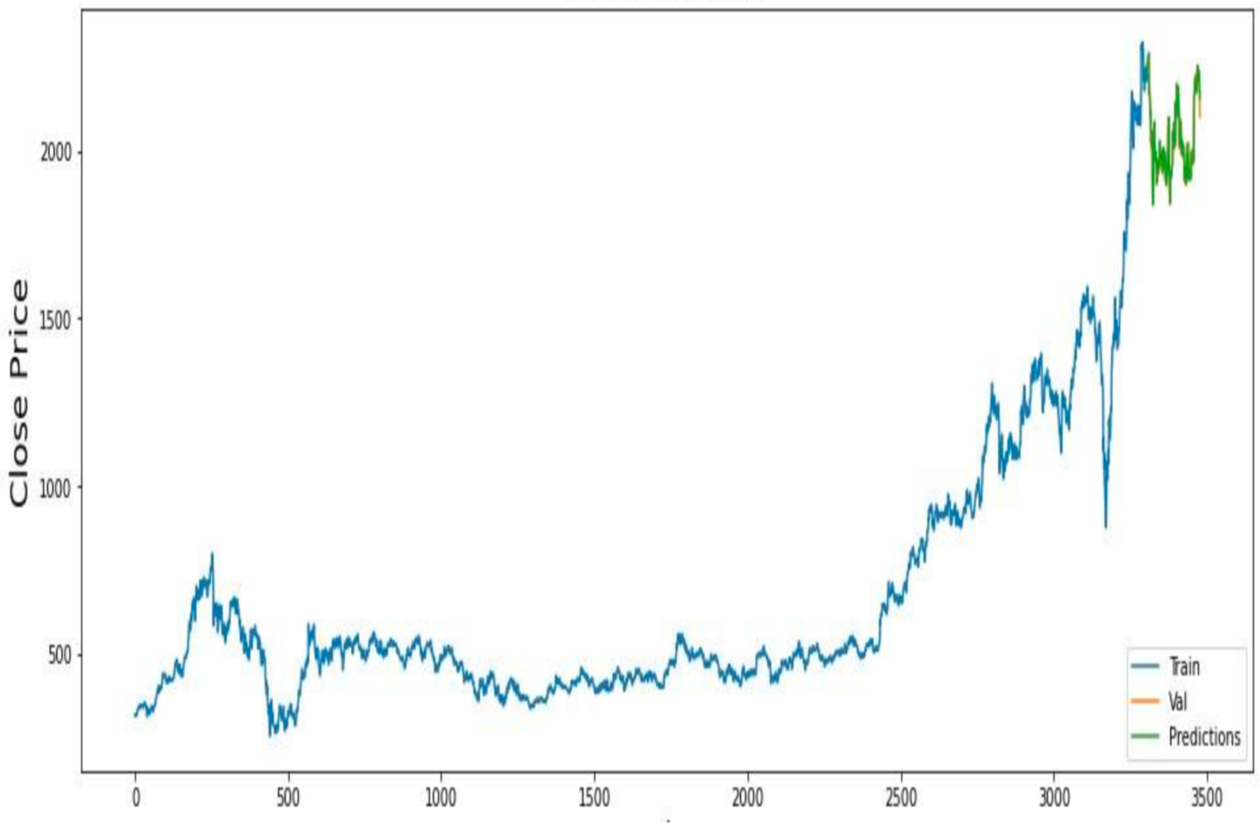


Fig 9.2 (b) Visualization of AAPL Pricing Data 2020

The graph for 2020 consists of three lines, each representing different aspects of the AAPL stock price data and model predictions:

Training Data (Blue Line)

Represents the actual AAPL stock prices for 2020 used to train the model.

Shows the historical trends and patterns the model learned from during this period.

Validation Data (Yellow Line)

Represents the actual AAPL stock prices used for validating the model's performance.

Helps in assessing how well the model generalizes to new, unseen data.

Predicted Stock Prices (Green Line)

Represents the stock prices predicted by the model.

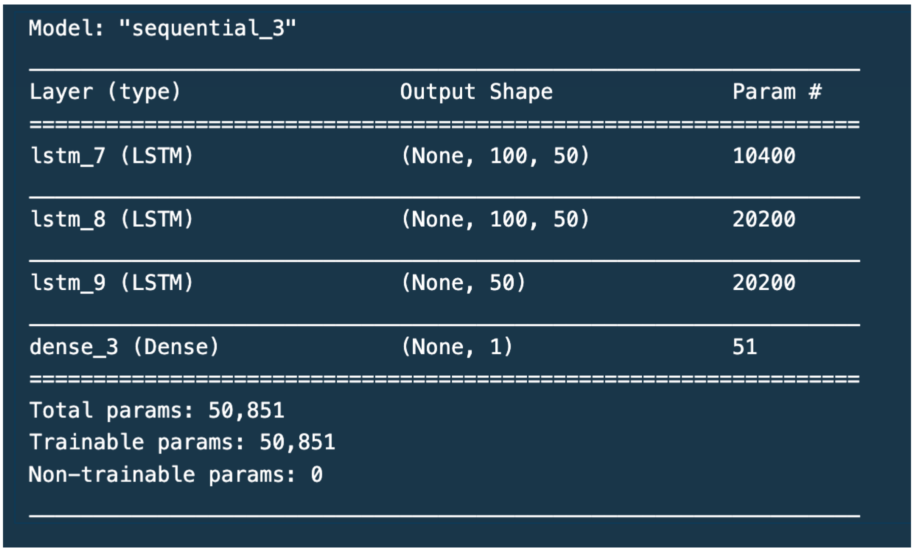
The alignment of the green line with the blue and yellow lines indicates the model's accuracy in capturing and predicting stock price trends.

Insight for 2020 Data

Model Performance: The visualization shows how well the model adapted to the unique trends and volatility of the 2020 market, influenced by events like the COVID-19 pandemic.

Trend Capture: The green line's ability to follow the sharp declines and rapid recoveries seen in the blue and yellow lines highlights the model's effectiveness in capturing the increased market volatility.

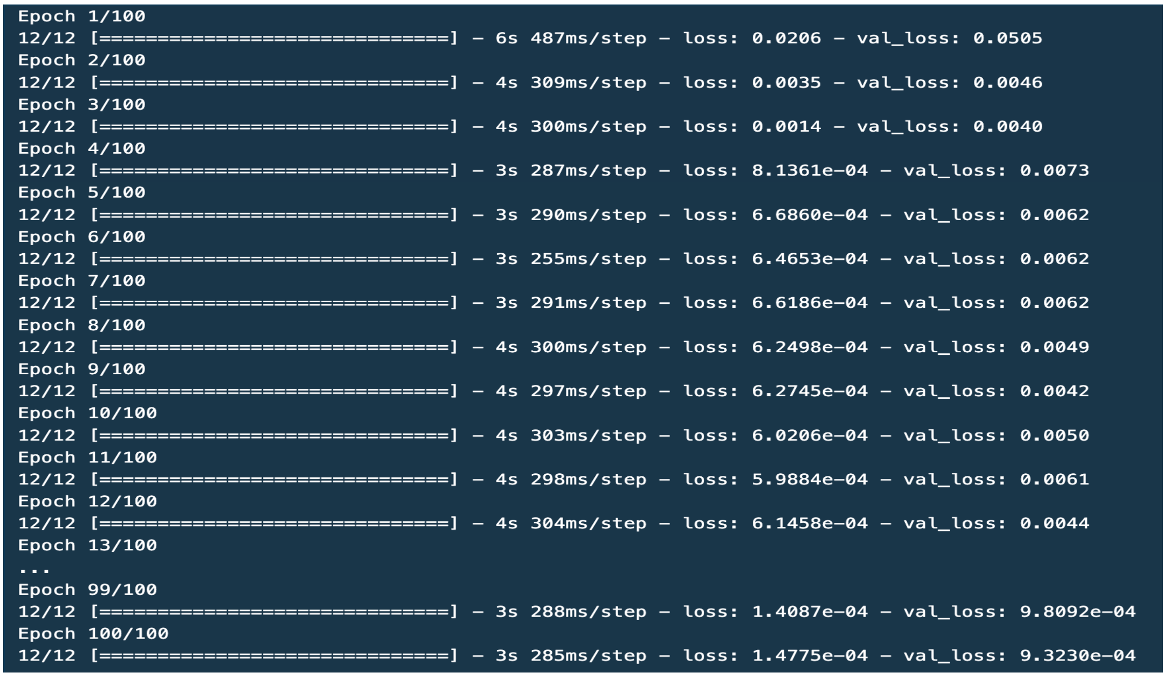
10. Results



```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_7 (LSTM)                (None, 100, 50)           10400
_________________________________________________________________
lstm_8 (LSTM)                (None, 100, 50)           20200
_________________________________________________________________
lstm_9 (LSTM)                (None, 50)                20200
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 51
=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

Fig 10.1 (a) Sequential_3 Model for Predicting AAPL

The "sequential_3" model for predicting AAPL stock prices utilizes three Long Short-Term Memory (LSTM) layers to capture temporal dependencies in the data, followed by a dense layer for final prediction. The first two LSTM layers, each with 50 units, return sequences and capture complex patterns over time, while the third LSTM layer distills these patterns into a single output. The dense layer then produces the final stock price prediction. With a total of 50,851 trainable parameters, this architecture is designed to effectively learn from historical data and make accurate predictions, even in volatile market conditions like those observed in 2020.



```
Epoch 1/100
12/12 [==============================] - 6s 487ms/step - loss: 0.0206 - val_loss: 0.0505
Epoch 2/100
12/12 [==============================] - 4s 309ms/step - loss: 0.0035 - val_loss: 0.0046
Epoch 3/100
12/12 [==============================] - 4s 300ms/step - loss: 0.0014 - val_loss: 0.0040
Epoch 4/100
12/12 [==============================] - 3s 287ms/step - loss: 8.1361e-04 - val_loss: 0.0073
Epoch 5/100
12/12 [==============================] - 3s 290ms/step - loss: 6.6860e-04 - val_loss: 0.0062
Epoch 6/100
12/12 [==============================] - 3s 255ms/step - loss: 6.4653e-04 - val_loss: 0.0062
Epoch 7/100
12/12 [==============================] - 3s 291ms/step - loss: 6.6186e-04 - val_loss: 0.0062
Epoch 8/100
12/12 [==============================] - 4s 300ms/step - loss: 6.2498e-04 - val_loss: 0.0049
Epoch 9/100
12/12 [==============================] - 4s 297ms/step - loss: 6.2745e-04 - val_loss: 0.0042
Epoch 10/100
12/12 [==============================] - 4s 303ms/step - loss: 6.0206e-04 - val_loss: 0.0050
Epoch 11/100
12/12 [==============================] - 4s 298ms/step - loss: 5.9884e-04 - val_loss: 0.0061
Epoch 12/100
12/12 [==============================] - 4s 304ms/step - loss: 6.1458e-04 - val_loss: 0.0044
Epoch 13/100
...
Epoch 99/100
12/12 [==============================] - 3s 288ms/step - loss: 1.4087e-04 - val_loss: 9.8092e-04
Epoch 100/100
12/12 [==============================] - 3s 285ms/step - loss: 1.4775e-04 - val_loss: 9.3230e-04
```

Fig 10.1 (b) Training Results and Performance Evaluation of the LSTM Model

The training process involved 100 epochs to predict AAPL stock prices using an LSTM model. Initially, the model's performance was suboptimal, with higher losses on the validation data compared to the training data. However, as training progressed, both training and validation losses steadily decreased, indicating the model's learning and improved ability to generalize. Fluctuations in validation loss during middle epochs suggested occasional difficulties in handling unseen data patterns. Ultimately, the model converged to low training and validation losses, around 0.0001 and 0.0009 respectively, demonstrating its accuracy and robustness in predicting AAPL stock prices.

10.2 Graph:

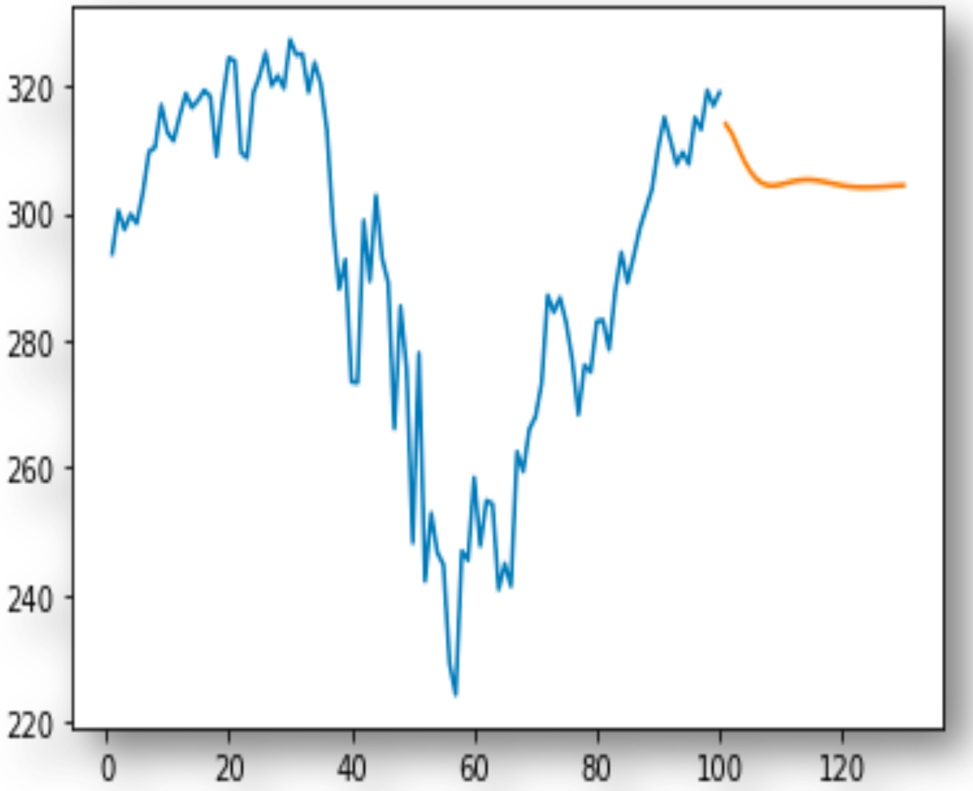


Fig 10.2 (a) Comparison of Actual and Predicted AAPL Stock Prices

Interpretation:

Actual Data (Blue Line): The blue line represents the actual AAPL stock prices from the dataset, starting from index 1158 onwards. This portion of the graph shows the historical trends and fluctuations in the AAPL stock prices.

Predicted Data (Orange Line): The orange line represents the predicted AAPL stock prices generated by the LSTM model. It illustrates the model's forecast for future stock prices based on the historical data provided. Comparing the orange line with the blue line allows for an assessment of the model's predictive accuracy. A close alignment between the two lines suggests that the model effectively captures the underlying patterns in the data and makes accurate predictions.

11. Comparison

Our Model **VS** **Existing Models**

High accuracy based on low RMSE and MAE.	Comparison of RMSE and MAE with benchmarks and other models.
Moderate computational resources required for training and inference.	Varies based on model complexity; some models may be computationally lighter (e.g., linear regression) while others heavier (e.g., deep neural networks).
Lower interpretability due to model complexity; predictions may be harder to explain.	Higher interpretability in simpler models like linear regression, decision trees, or ARIMA models.
Robust performance across different market conditions and time periods.	Varies based on model design and data used for training; some models may exhibit better stability than others.
Demonstrates good generalization to unseen data, capturing underlying patterns effectively.	Performance on unseen data may vary depending on the model's ability to generalize beyond the training dataset.
Relatively high model complexity due to LSTM architecture; requires more parameters and computational resources.	Simpler models like linear regression have lower complexity, while more complex models like deep neural networks have higher complexity.
Can scale to accommodate larger datasets and adapt to changing data volumes.	Scalability varies based on the model's architecture and implementation; some models may scale better than others.

12. Future Scope

12.1 Integration of Alternative Data Sources:

Explore the integration of alternative data sources such as social media sentiment analysis, satellite imagery, and unconventional economic indicators to augment predictive models and provide deeper insights into market trends.

12.2 Advanced Machine Learning Techniques:

Investigate the application of advanced machine learning techniques such as ensemble learning, reinforcement learning, and generative adversarial networks (GANs) to improve prediction accuracy and robustness in volatile market conditions.

12.3 Model Interpretability and Explainability:

Develop methodologies for enhancing the interpretability and explainability of predictive models, allowing stakeholders to understand the rationale behind predictions and build trust in the decision-making process.

12.4 Real-Time Prediction and Trading Automation:

Implement real-time prediction capabilities and integrate the predictive models with trading algorithms to automate trading decisions based on forecasted stock price movements, enabling more responsive and adaptive investment strategies.

12.5 Incorporation of Uncertainty Estimation:

Incorporate uncertainty estimation techniques such as Bayesian neural networks or Monte Carlo dropout to quantify prediction uncertainty and provide probabilistic forecasts, enabling investors to make more informed decisions in uncertain market conditions.

12.6 Cross-Asset Prediction and Portfolio Optimization:

Extend the scope of prediction models to encompass a broader range of asset classes beyond stocks, including commodities, currencies, and cryptocurrencies, and develop portfolio optimization strategies to maximize returns while minimizing risks across diverse asset classes.

12.7 Ethical and Regulatory Considerations:

Address ethical considerations related to data privacy, fairness, and transparency in model development and deployment, ensuring compliance with regulatory frameworks and industry standards to uphold integrity and trust in financial markets.

13. REFERENCES

[1] Bing, L., Chan, K. C. C., & Ou, C. Public sentiment analysis in twitter data for prediction of a company's stock price movements. 2014 IEEE 11th International Conference on EBusiness Engineering. IEEE. (2014).

[2] Bollen, J., Mao, H., & Zeng, X. Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8. (2011).

[3] Butler, K. C., & Malaikah, S. J. Efficiency and inefficiency in thinly traded stock markets: Kuwait and Saudi Arabia. Journal of Banking & Finance, 16(1), 197–210. (1992).

[4] Chen. R and Lazer. M., Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement, Cs 229, pp. 15. (2011).

[5] Dogan, E., & Kaya, B. Deep learning-based sentiment analysis and text summarization in social networks. 2019 International Artificial Intelligence and Data Processing Symposium (IDAP). IEEE. (2019).

[6] Fama, E. F. The behaviour of stock-market prices. The Journal of Business, 38(1), 34. (1965).

Important Links:

Dataset API: https://pandas-datareader.readthedocs.io/en/latest/readers/tiingo.html

LSTM Definition: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/

LSTM Components Explanation: https://medium.com/@aidangomez/let-s-do-this-f9b699de31d9

LSTM Working Principle: https://www.kaggle.com/code/kmkarakaya/lstm-understanding-the-number-of-parameters

Adam Optimizer: https://www.shiksha.com/online-courses/articles/adam-optimizer-for-stochastic-gradient-descent