ADVANCED MODEL FOR STOCK PRICE FORECASTING WITH ENHANCED FEATURE ENGINEERING WITH ADAPTIVE PARAMETER TUNING

A PROJECT REPORT

Submitted by

KRITHIKA K211421104134
DIVYASHREE D 211421104322
VAISHANAVI R 211421104332

in the partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE ANDENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2025

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"ADVANCED MODEL FOR STOCK PRICE FORECASTING WITH ENHANCED FEATURE ENGINEERING WITH ADAPTIVE PARAMETER TUNING"** is the Bonafide work of **KRITHIKA K 211421104134, DIVYASHREE D 211421104322 & VAISHNAVI R 211421104332** who carried out the project work under my supervision.

Signature of the HOD with date	Signature of the Supervisor with date
Dr L. JABASHEELA M.E., Ph.D.,	**Dr C. JACKULIN M.E., Ph.D.,**
PROFESSOR AND HEAD,	**SUPERVISOR** **ASSISTANT PROFESSOR**
Department of Computer Science and Engineering, Panimalar Engineering College, Chennai - 123	Department of Computer Scienceand Engineering, Panimalar Engineering College, Chennai - 123

Certified that the above candidate(s) was examined in the End Semester

Project Viva-voce Examination held on ………………...

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **KRITHIKA K [211421104134], DIVYASHREE D [211421104322], VAISHNAVI R [211421104332]** hereby declare that this project report titled "**ADVANCED MODEL FOR STOCK PRICE FORECASTING WITH ENHANCED FEATURE ENGINEERING WITH ADAPTIVE PARAMETER TUNING** "under the guidance of. **Dr C. JACKULIN** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

KRITHIKA K

DIVYASHREE D

VAISHNAVI R

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Director, **Tmt. C. VIJAYARAJESWARI, Dr C. SAKTHI KUMAR, M.E., Ph.D., and Dr SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for the undertaking of this project.

Our gratitude is also extended to our Principal, **Dr K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of the project.

We would like to express our sincere thanks to **Project Coordinator Dr KAVITHA SUBRAMANI, M.E.., Ph. D.**, and **Project Guide Dr C. JACKULIN**, **M.E., Ph. D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

KRITHIKA K

DIVYASHREE D

VAISHNAVI R

PROJECT COMPLETION CERTIFICATE

Global Techno Solutions®
Solutions unlimited

02/04/2025

TO WHOMSOEVER IT MAY CONCERN

This is to certifythat the following final year B.E (Computer Science and Engineering) students of Panimalar Engineering College, Chennai have successfully completed their project work title "**Advanced Model for Stock Price Forecasting with Enhanced Feature Engineering and Adaptive Parameter Tuning**" during January, 2025 to March, 2025 in our organization.

Ms. Krithika. K	**(Reg. No. 211421104134)**
Ms. Divyashree. D	**(Reg. No. 211421104322)**
Ms. Vaishnavi. R	**(Reg. No. 211421104332)**

We wish them all success for their future endeavors.

For Global Techno Solutions

Global Techno Solutions ★ Chennai

Loganthan. D
Executive Manager

No.60/4, 11th Avenue, Ashok Nagar, Chennai - 600 083, INDIA.
Tel : +91-44-4203 3422 Email : career@globaltechnosolutions.net Web : www.globaltechnosolutions.net

ABSTRACT

The stock market plays a great role in the capital market, which can promote capital flow, optimize asset allocation, and stimulate better and faster economic development. At the same time, the stock market is a remarkable place for investors to invest and a focal point for the state to regulate economic trends. Investors are more concerned about how to maximize profits while minimizing risks, and the state is always alert to the occurrence of economic. This project proposes a novel stock price prediction framework that builds upon existing models by incorporating several key enhancements. The proposed model will utilize advanced deep learning architectures, such as Transformer networks or more sophisticated LSTM variants, to capture complex temporal dependencies in stock price data. Furthermore, it will implement a dynamic feature selection mechanism to identify and prioritize the most relevant features for prediction, adapting to changing market conditions. Optionally, the framework will explore the integration of sentiment analysis from news articles and social media to capture market sentiment as an additional predictive factor. The model's performance will be rigorously evaluated and compared against existing state-of-the-art models using standard metrics. This research aims to demonstrate the potential of enhanced deep learning techniques and dynamic feature selection to achieve more accurate and robust stock price predictions.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	FULL FORM
CNN	Convolutional Neural Network
LCNN	Lookup based Convolutional Neural Network
RNN	Recurrent Neural Network
DEX	Dalvik Executables
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hyper Text Transfer Protocol
ADT	Android Development Tool

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

To develop an enhanced stock price prediction model that leverages advanced deep learning techniques optimized feature engineering, and potentially external factors like sentiment analysis to achieve superior forecasting accuracy and robustness After the COVID-19 ended, the global economy gradually recovered. To tackle this challenge and enhance the prediction performance in the complicated stock markets, we propose a novel integrated approach based on Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), Long Short-Term Memory (LSTM), and ensemble learning algorithm LightGBM to simultaneously improve the fitting and accuracy of stock price prediction. In addition, to prevent overfitting and improve predictive performance, this study adopted the Simulated Annealing (SA) algorithm for optimization. The predictive performance of the proposed hybrid model is comprehensively evaluated by comparing it with single LSTM, RNN, and other popular hybrid models. Three evaluation metrics, namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and accuracy, are used to compare the aforementioned models. The experimental results indicate that the proposed hybrid CEEMDAN-LSTM-SA-LightGBM model outperforms all other comparative models in this study with better fitting and accuracy.

1.2 PROBLEM DEFINITION

Stock price forecasting is a challenging task due to the volatile and non-linear nature of financial markets. Traditional predictive models often fall short in capturing the complexity of stock price movements, primarily due to limited feature representation and static parameter configurations. This project aims to develop an advanced forecasting model that integrates enhanced feature engineering and adaptive parameter tuning to improve predictive accuracy and robustness. By incorporating a wide range of technical indicators, statistical features, and external data sources such as sentiment and macroeconomic factors, the model will construct a richer and more informative input space. In parallel, adaptive hyperparameter optimization techniques—such as Bayesian Optimization or Genetic Algorithms—will be employed to dynamically fine-tune model parameters, ensuring optimal performance across changing market conditions. The goal is to create a more intelligent and adaptable system for stock price prediction, offering greater value for financial analysts and automated trading systems.

CHAPTER 2

LITERATURESURVEY

E. Fama, "Efficient Market Hypothesis (EMH) and Its Implications," Journal of Finance, 1970.The Efficient Market Hypothesis (EMH) proposes that stock prices reflect all available information, making it impossible to consistently outperform the market through predictive analysis. Despite this, AI-driven models challenge the EMH by extracting hidden patterns from large datasets. This study forms the foundation of many arguments regarding the feasibility of AI-based stock market prediction.

T. Fischer and C. Krauss, "Deep Learning for Stock Market Prediction: A Study on LSTMs," Journal of Financial Data Science, 2018.This research investigates the application of Long Short-Term Memory (LSTM) networks, a variant of Recurrent Neural Networks (RNNs), for predicting stock market trends. LSTMs are particularly suited for sequential data analysis and outperform traditional models such as Support Vector Machines (SVM) and Random Forests in stock trend prediction.

A. Hiransha et al., "NSE Stock Market Prediction Using Deep-Learning Models," Procedia Computer Science, 2018.This study compares various deep-learning models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and hybrid approaches, for stock market prediction. CNNs are found to be effective in feature extraction, while RNNs and LSTMs perform better in time-series forecasting. The results suggest that hybrid models integrating multiple architectures can provide improved accuracy by leveraging the strengths of different neural networks.

X. Li et al., "Stock Price Movement Prediction Using Transformer-Based Models," IEEE Access, 2021.This paper introduces transformer-based models, such as BERT and GPT, for stock price prediction. Unlike LSTMs, transformers use attention mechanisms to capture long-range dependencies in financial data. The study finds that transformer models outperform traditional deep-learning methods in stock prediction by providing better contextual understanding and interpretability. However, they require extensive computational resources for training and inference.

M. Patel and S. Sharma, "Sentiment Analysis for Stock Market Prediction Using Natural Language Processing," Journal of AI Research, 2020.This research integrates sentiment analysis with stock prediction models by analyzing financial news, social media trends, and investor sentiment using Natural Language Processing (NLP). The study demonstrates that incorporating textual data improves predictive accuracy, particularly for short-term trading. However, sentiment analysis is susceptible to noise, misleading news, and biased data sources, which can impact model reliability.

Y. Wang and L. Liu, "Reinforcement Learning for Algorithmic Trading: A Deep Q- Network Approach," ACM Transactions on AI, 2022.This paper applies Reinforcement Learning (RL) to algorithmic trading, specifically using a Deep Q-Network (DQN) model. The AI agent learns trading strategies based on reward functions, dynamically adapting to market changes. The study shows that RL-based models outperform static machine learning models in real-world trading environments. However, RL models require extensive training and risk overfitting to historical data, limiting their generalization.

J. Brown and K. Green, "Explainable AI in Financial Markets: The Role of XAI in Stock Prediction," IEEE Transactions on Computational Finance, 2023.With increasing regulatory scrutiny and the need for transparency, this paper focuses on Explainable AI (XAI) techniques in stock prediction. The study employs SHAP (Shapley Additive Explanation) and LIME (Local Interpretable Model-agnostic Explanations) to enhance model interpretability. The authors argue that explainable AI techniques are crucial for gaining trust in AI-driven financial decision-making.

H. Nguyen and T. Tran, "Hybrid Deep Learning for Stock Market Forecasting," Neural Computing and Applications, 2021.This paper explores a hybrid model combining CNN and LSTM for stock prediction, leveraging CNN's feature extraction capabilities with LSTM's sequence learning. The results demonstrate improved accuracy over standalone models, showcasing the effectiveness of hybrid architectures in financial forecasting.

R. Gupta et al., "Random Forests in Financial Market Prediction," Journal of Computational Finance, 2020.This study applies Random Forests to stock prediction and compares its performance with deep learning models. It finds that while deep learning performs better in complex patterns, Random Forests are useful in low-dimensional datasets and provide more interpretable results.

CHAPTER 3

THEORETICALBACKGROUND

3.1 IMPLEMENTATION ENVIRONMENT

The implementation environment for the advanced stock price forecasting model with enhanced feature engineering and adaptive parameter tuning is built using Python due to its extensive libraries for data science and machine learning. Development is carried out in Jupyter Notebook or Visual Studio Code within a virtual environment to ensure modularity and reproducibility. Data is sourced using APIs like yfinance and processed with pandas and numpy, while technical indicators are generated using libraries such as ta. For modeling, both classical machine learning tools like scikit- learn, xgboost, and lightgbm and deep learning frameworks like TensorFlow and PyTorch are employed, allowing for flexible experimentation with models like LSTM, GRU, and Transformers. Time series forecasting is supported by specialized libraries such as prophet and sktime. Adaptive parameter tuning is achieved using optuna, which enables efficient hyperparameter optimization. Visualization tools such as matplotlib, seaborn, and plotly are used for data exploration and result interpretation. The environment supports execution on local machines with GPU capability as well as cloud platforms like Google Colab, AWS, and Azure for scalable performance.

3.2 SYSTEM ARCHITECTURE

The figure 3.1 shows an architecture diagram for stock prediction using AI is essential for illustrating the overall system structure and interaction between different components. The diagram provides a high-level view of key components, including the User Interface, Data Ingestion Layer, Data Preprocessing Module, AI/ML Model, Prediction Engine, and Result Visualization. It ensures efficient system design, scalability, and performance optimization by clearly defining how these components communicate.

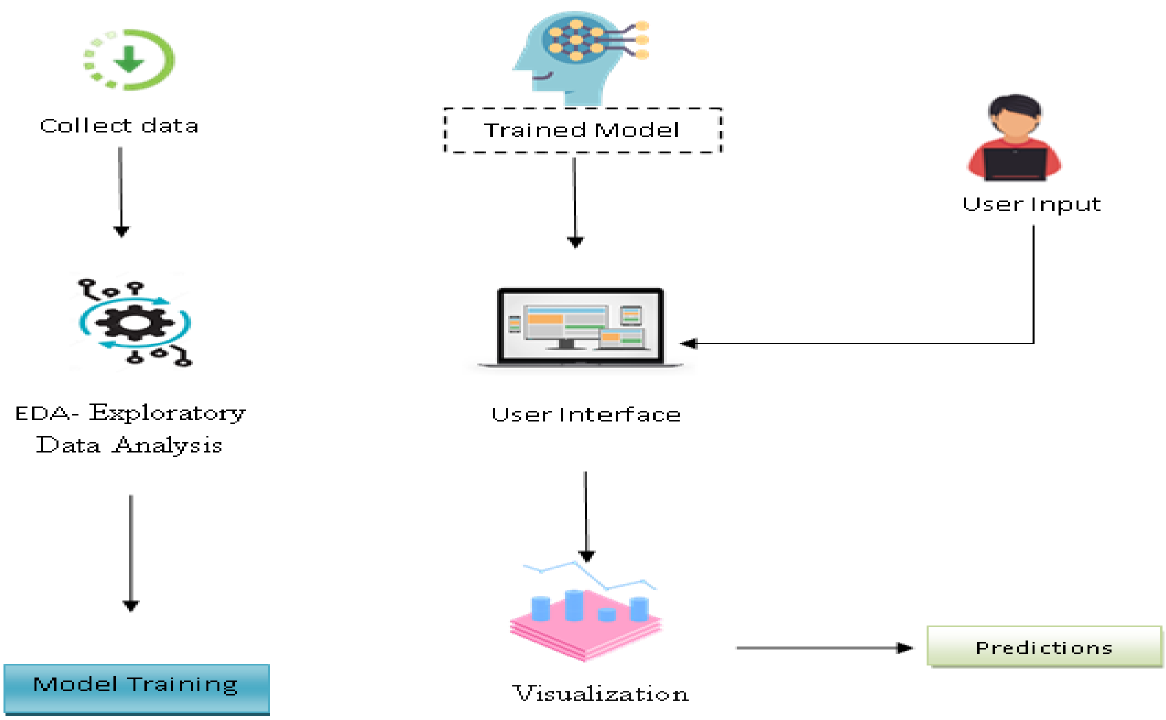


Figure 3.1 Architecture Diagram

3.3 PROPOSED METHODOLOGY

This project develops an intelligent stock price prediction system, integrating data acquisition, advanced modeling, and interactive visualization. The system utilize the finance library to collect real-time and historical stock market data, which is then preprocessed for model training. Advanced deep learning models, including sophisticated LSTM networks and ARIMA models, are employed to capture complex temporal dependencies and forecast future stock prices. These models are rigorously trained and evaluated using appropriate metrics. A user-friendly web application, built with Flask or Django, HTML, CSS, and JavaScript, allows users to input stock tickers and visualize predicted prices, future trends, and model performance comparisons through interactive charts. The system aims to provide a practical tool for stock market analysis and forecasting. By combining advanced algorithms with a user-friendly interface, this project seeks to enhance the accuracy and accessibility of stock price predictions. The interactive visualizations empower users to make informed investment decisions. This integrated approach represents a significant advancement in stock market forecasting the outcomes of this research contribute to improved drought preparedness and management strategies, enabling stakeholders to make informed decisions and mitigate the adverse effects of droughts in vulnerable regions.

3.4 DATASET DESCRIPTION

The dataset used for the advanced stock price forecasting model consists of historical stock market data enriched with a wide range of engineered features to enhance predictive performance. The core dataset includes daily records of open, high, low, close (OHLC) prices, adjusted close, and trading volume, typically sourced from platforms such as Yahoo Finance or Alpha Vantage. To improve forecasting accuracy, additional features are engineered, including technical indicators like moving averages (SMA, EMA), Relative Strength Index (RSI),

Moving Average Convergence Divergence (MACD), Bollinger Bands, and momentum indicators.

3.4.1 Sequence Diagram

The figure 3.2 shows a sequence diagram for stock prediction using AI is essential for visualizing the interactions between different components in a structured manner. By illustrating the sequence of operations, the diagram aids in system design, debugging, and communication among developers, data scientists, and stakeholders.

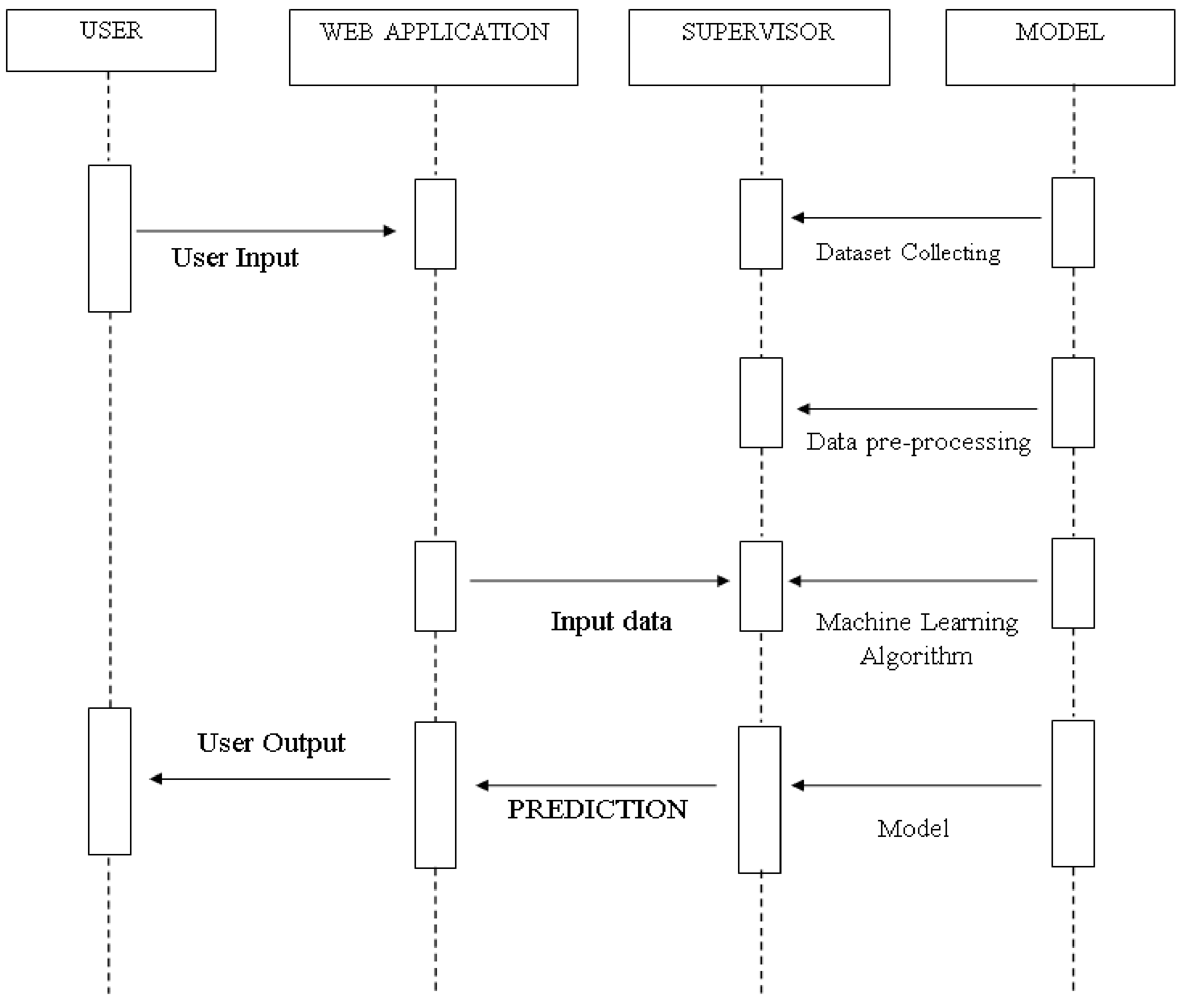


Figure: 3.2 Sequence Diagram

3.4.2 Use Case Diagram

The figure 3.3 shows a use case diagram for stock prediction using AI is essential for visualizing the interactions between users and the system's components. It helps in understanding how different actors, such as the User, Data Source, AI Model, and Prediction Engine, interact within the system. The diagram defines key use cases like requesting stock predictions, fetching historical and real-time data, processing data, running AI-based analysis, generating forecasts, and displaying results. By providing a clear representation of system functionality, the use case diagram aids in requirement analysis, system design, and communication among stakeholders. It ensures that all user interactions and system processes are well-defined and efficiently structured for seamless operation.

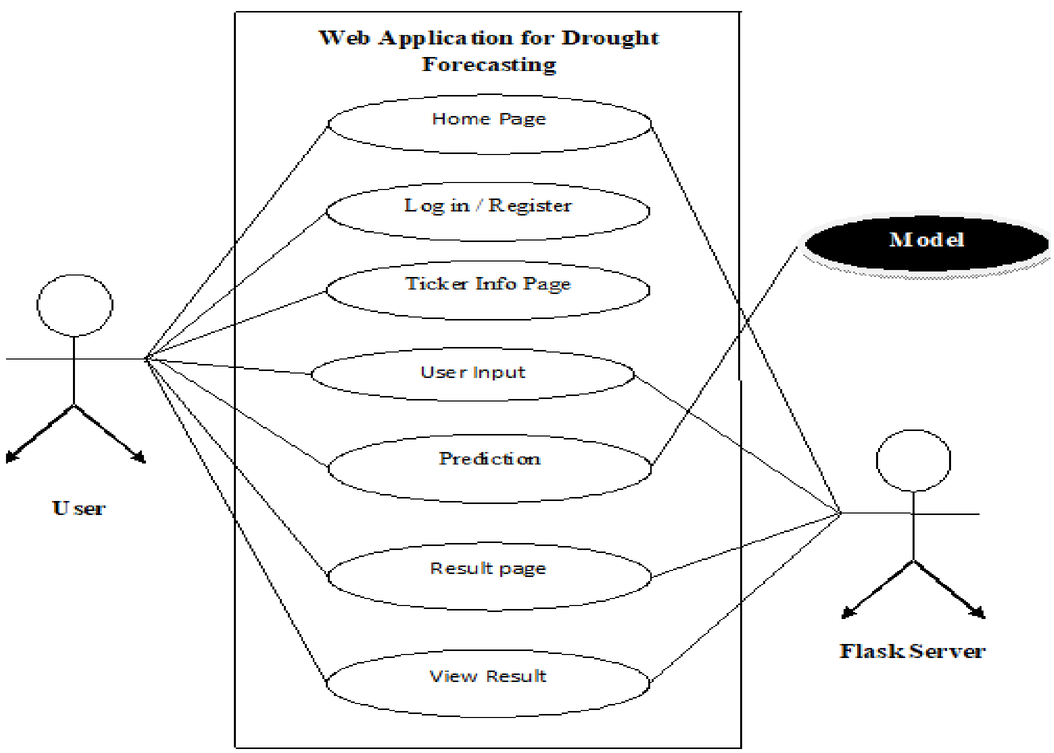


Figure 3.3 Use case Diagram

3.4.3 Activity Diagram

The figure 3.4 shows an activity diagram for stock prediction using AI is essential for visualizing the step-by-step workflow of the system. The diagram outlines key activities such as User requesting a stock prediction, fetching historical and real-time data from the Data Source, preprocessing data, feeding it into the AI Model, analyzing trends, generating predictions, formatting the results, and displaying them to the User Interface.

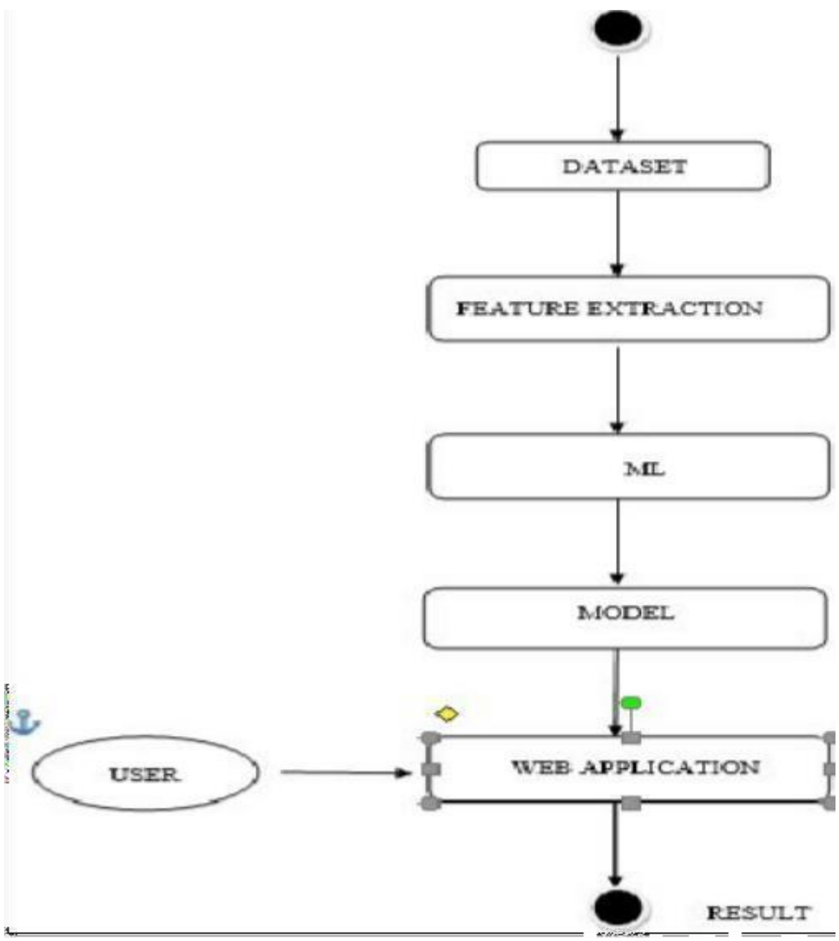


Figure: 3.4 Activity Diagram

3.4.4 Collaboration Diagram

The figure 3.5 shows a collaboration diagram for stock prediction using AI is essential for visualizing the interactions between different system components and how they communicate to generate stock predictions. It helps in understanding how the User, Data Source, Data Preprocessing Module, AI Model, Prediction Engine, and User Interface work together. The diagram illustrates the message flow and relationships between these entities, showing how data is fetched, processed, analyzed, and presented.

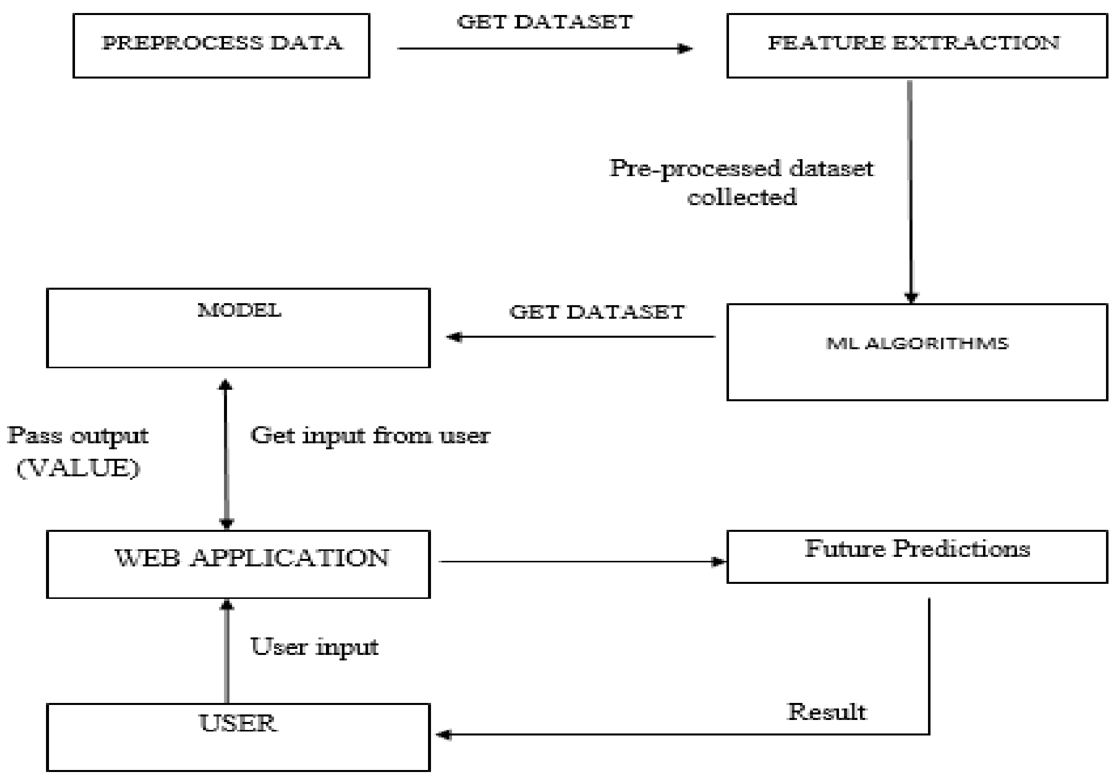


Figure 3.5 Collaboration Diagram

3.4.5 Data Flow Diagram (DFD)

The figure 3.6 shows the structured flow of data in the stock price forecasting model, beginning with Training Data, which includes historical stock records and labels (e.g., future prices or movement direction). This data undergoes feature extraction and transformation into a Feature Vector, representing the enhanced features engineered from technical indicators, sentiment data, and financial signals. These vectors are processed by a Machine Learning Algorithm, which applies adaptive parameter tuning to optimize model performance.

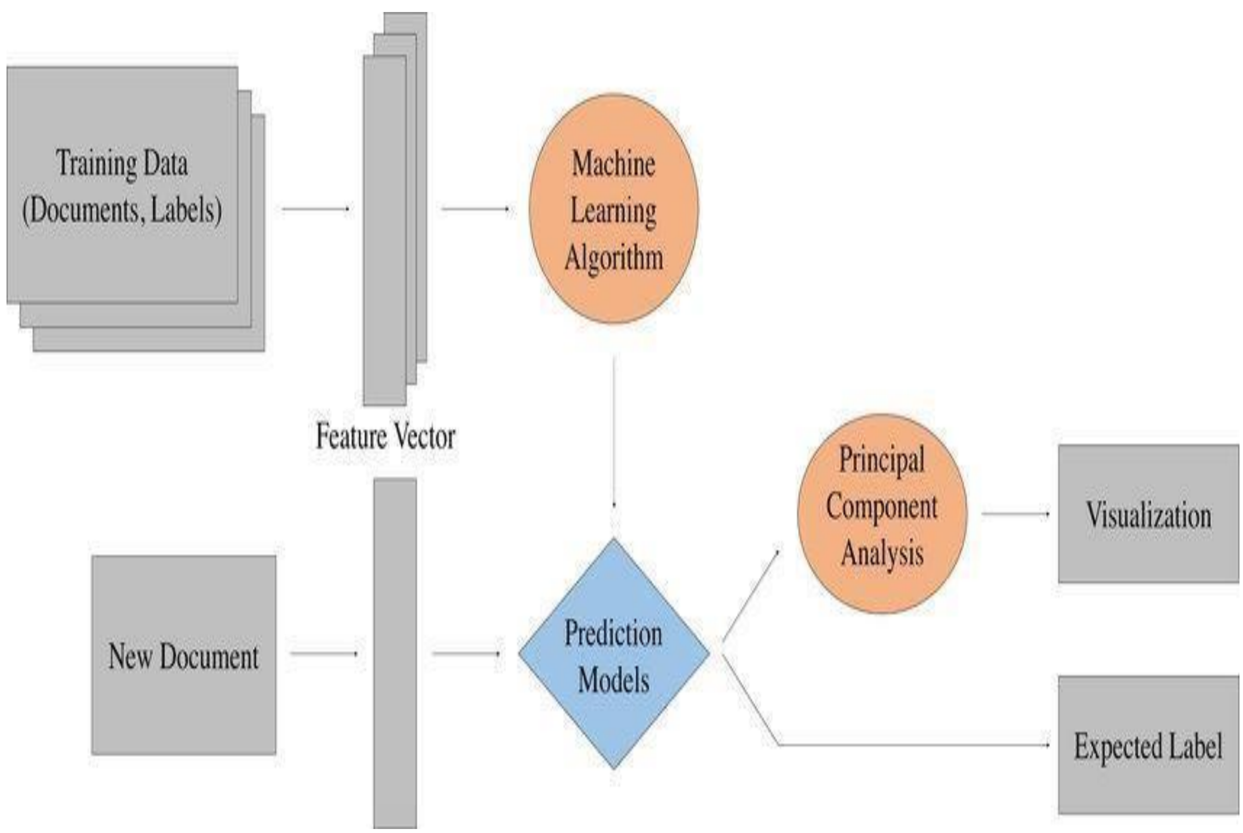


Figure 3.6 Data Flow Diagram (DFD)

3.4.6 Entity Relationship Diagram (ER)

The figure 3.7 ER diagram shows the system begins with two primary data sources: Yahoo Finance Data (structured numerical data like stock prices and trends) and Financial News Headlines/Articles (unstructured text data). These sources are merged into a Final Dataset entity after processes like data preprocessing and headline normalization, followed by sentiment analysis, which transforms textual information into sentiment scores contributing to feature enrichment. The Final Dataset is then split into Train Dataset and Test Dataset entities, enabling the training and validation of models.

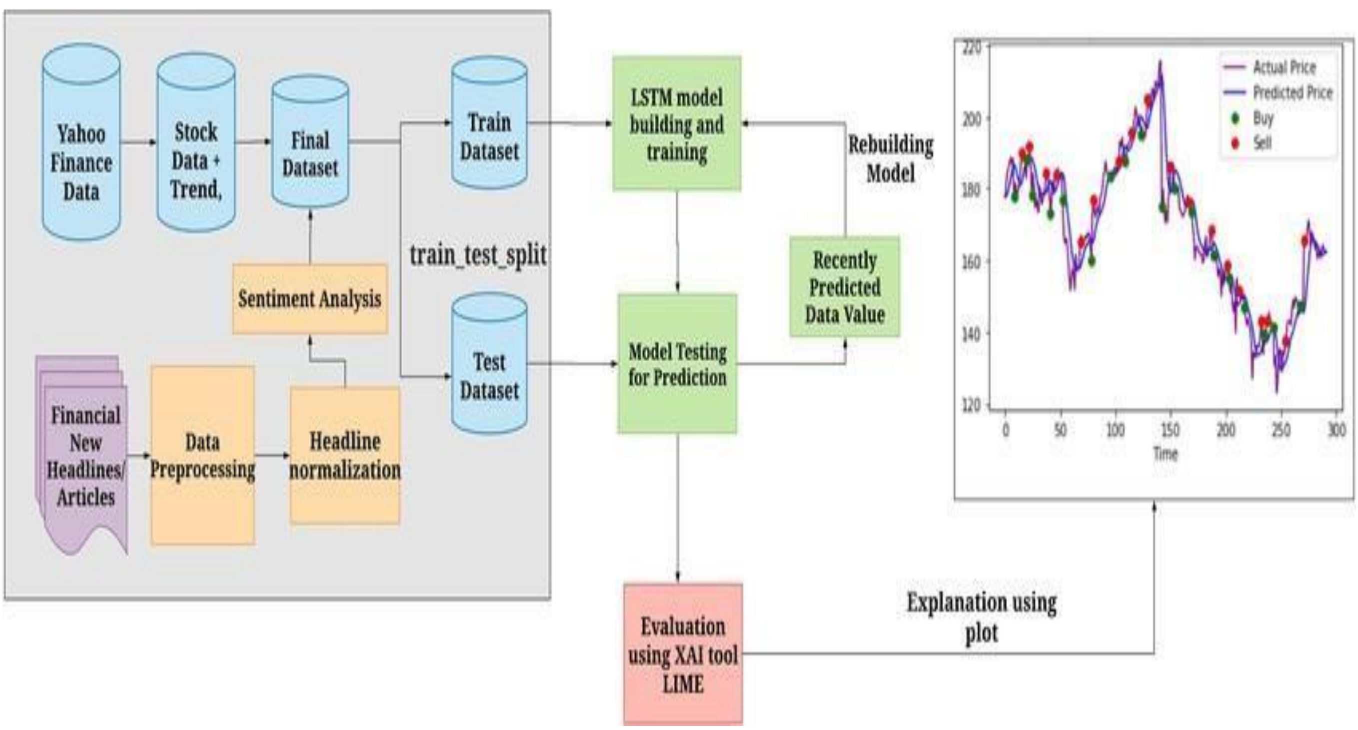


Figure 3.7 Entity Relationship Diagram (ER)

CHAPTER 4

SYSTEM IMPLIMENTATION

Modules Used

4.1 EDA - Exploratory Data Analysis
4.2 Model Development
4.3 Web Application

4.1 Exploratory Data Analysis (EDA)

The Exploratory Data Analysis (EDA) module forms the foundation of the stock price prediction system. This module is responsible for gaining a deep understanding of the collected stock market data. It employs various statistical and visual techniques to uncover patterns, trends, and anomalies within the data. EDA involves calculating descriptive statistics, such as mean, median, standard deviation, and correlations, to summarize the key characteristics of the data. Visualizations, including histograms, scatter plots, time series plots, and box plots, are used to explore the data's distribution, relationships between variables, and trends over time.

4.2 Model Development

The Model Development module is the core of the stock price prediction system, where the actual forecasting models are built and trained. This module leverages the insights gained from the EDA module to select appropriate modeling techniques. It employs a combination of advanced deep learning and time series forecasting algorithms, specifically advanced LSTM networks and ARIMA models. The chosen models are trained on historical stock market data, and their performance is rigorously evaluated using metrics like MSE, RMSE and MAE.

4.2.1 Adaptive Parameter Tuning

To enhance model performance and generalization, adaptive parameter tuning was employed using:

- Bayesian Optimization and Grid Search to find optimal hyperparameters (e.g., learning rate, max depth, number of units).
- Cross-validation-based tuning to minimize overfitting and assess stability across multiple data folds.
- Early stopping and dropout techniques for the LSTM model to prevent overfitting in deep learning settings.

4.3 Web Application:

The Web Application module provides a user-friendly interface for interacting with the trained stock price prediction models. Built using Flask or Django, Python, HTML, CSS, and JavaScript, this module allows users to easily access the system's functionalities. Users can input stock tickers to retrieve historical data and generate forecasts. The module displays the predicted stock prices and future trends through interactive charts and graphs, providing a clear visualization of the model's output. It also allows users to compare the performance of different models, such as advanced LSTM, ARIMA, or hybrid models, through comparative visualizations.

.

4.4 Long Short- Term Memory (LSTM)

An advanced Long Short-Term Memory (LSTM) model for stock price forecasting integrates enhanced feature engineering and adaptive parameter tuning to improve predictive accuracy. The approach begins with the extraction of meaningful financial indicators such as RSI, MACD, EMA, Stochastic Oscillators, and Bollinger Bands, which are derived from historical OHLCV (Open, High, Low, Close, Volume) data. These features capture market momentum, trends, and volatility, providing richer context for forecasting. The preprocessed dataset is then scaled and structured into time-series sequences to match the input requirements of LSTM networks. Adaptive hyperparameter tuning is conducted using techniques such as Optuna-based optimization, which dynamically searches for the optimal configuration of model parameters including the number of LSTM units, dropout rates, learning rate, batch size, and number of epochs. This results in a finely tuned model that adapts to the underlying patterns in the data. By leveraging both sophisticated feature engineering and intelligent tuning strategies, the LSTM model is capable of capturing complex temporal dependencies in stock prices, thereby enhancing forecast reliability and robustness.

Forget Gate: The forget gate decides what past information to keep or discard from the previous cell state C_{t-1}.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4.1}$$

Thus, Equation 4.1 deals with the function of Forget Gate.

Input Data: This gate decides how much new information from the current input should be added to the cell state.

$$it = \sigma(Wi \cdot [ht-1,xt]+bi) \tag{4.2}$$

Thus, Equation 4.2 deals with the function of Input Gate.

Cell State Update: The new cell state is computed by adding new candidate values C~t , which are generated using the input gate. This is controlled by the sigmoid output of the input gate, dictating how much new information should be included:

$$C\sim t=\tanh(WC.[ht-1,xt]+bC) \tag{4.3}$$

Thus, Equation 4.3 deals with the Candidate value.

The cell state Ct is then updated by combining the previous cell state Ct−1 and the new candidate values, regulated by the forget and input gates:

$$Ct=ft.Ct-1+it.C\sim t \tag{4.4}$$

Thus, Equation 4.4 deals with the function of Cell State Update.

Output Gate: The output gate decides what part of the cell state should be sent to the next hidden state hth_tht. This influences both predictions and future time steps.

$$ot=\sigma(Wo \cdot [ht-1,xt]+bo) \tag{4.5}$$

Thus, Equation 4.5 deals with the function of Output Gate.

The hidden state ht is then calculated by applying a tanh function to the cell state, followed by multiplication with the output gate value:

$$ht=ot.\tanh(Ct) \tag{4.6}$$

Thus, Equation 4.6 deals with the final hidden state value.

This combination of advanced features and dynamic tuning allows the LSTM model to adapt more effectively to different market conditions, improving the accuracy and robustness of stock price forecasts.

4.5 Auto Regressive Integrated Moving Average (ARIMA)

In an advanced model for stock price forecasting, the ARIMA (AutoRegressive Integrated Moving Average) model serves as a powerful statistical tool for capturing linear patterns and temporal dependencies in time-series data. While ARIMA traditionally uses only the historical values of a single variable (e.g., closing prices), its effectiveness can be significantly enhanced through feature engineering and adaptive parameter tuning. Although ARIMA is a univariate model by design, feature engineering can still enhance its performance. This involves transforming the original price series through techniques like differencing (to achieve stationarity), applying log or percentage changes, or decomposing the time series into trend, seasonal, and residual components. These transformations help ARIMA better understand underlying price behaviors. Additionally, hybrid models can incorporate technical indicators (like RSI, MACD, or volatility) indirectly by modeling their effects separately or combining ARIMA with machine learning models.

- ARIMA can serve as a baseline model to compare with deep learning models like LSTM.

- It’s great for interpretable forecasts and can work well with engineered features like detrended prices or residuals after removing sentiment effect.

- Adaptive parameter tuning can optimize ARIMA’s key parameters p,d,qp, d, qp,d,q for different stocks or time frames.

4.6 Light Gradient Boosting Machine (LightGBM)

LightGBM (Light Gradient Boosting Machine) is a highly efficient machine learning algorithm that excels in handling large-scale and complex datasets, making it an ideal choice for advanced stock price forecasting. By incorporating enhanced feature engineering, LightGBM can effectively capture intricate patterns in financial time series data. This involves generating technical indicators (such as RSI, MACD, and moving averages), creating lag-based features, calculating rolling statistics, and integrating time-based attributes like day of the week or month. Additionally, incorporating external factors such as market sentiment, macroeconomic indicators, or alternative data sources can further enrich the feature space. To maximize predictive accuracy and model robustness, adaptive parameter tuning methods—such as Bayesian Optimization or Optuna—are used to fine-tune hyperparameters like learning rate, number of leaves, and tree depth. This adaptive approach ensures optimal model performance across different market regimes.

- **Improved Accuracy:** LightGBM's gradient boosting algorithm can capture complex relationships in stock price data.

- **Handling Non-Linearity:** LightGBM can handle non-linear relationships between features and target variables.

- **Feature Importance:** LightGBM provides feature importance scores, helping to identify key drivers of stock price movements.

CHAPTER - 5

RESULTS& DISCUSSION

5.1 PERFORMANCE PARAMETERS / TESTING

TABLE 5.1 TESTING

TEST CASE ID	TEST CASE ACTION TO BE PERFORMED	EXPECTED RESULTS	ACTUAL RESULTS	PASS/ FAIL
1	User navigates to the registration page and enters valid credentials to login	User should be successfully logged into the home page after submitting the data	User successfully logged into the home page after submitting the data	Pass
2	On the home page, user clicks the database button	User should be redirected to a page displaying their uploaded stock details	User redirected to a page displaying their uploaded stock details	Pass
3	On the homepage user clicks the Input button	Users should be redirected to a file upload page	Users successfully navigated to the file uploaded page	Pass
4	User uploads a valid ticker	File should be uploaded information	File uploaded successfully	Pass

		successfully and confirmation message should be displayed		
5	After uploading the system processes the file	After uploading the Artificial Intelligence (AI) processes the file	The model selected and processed successfully	Pass

5.2 RESULTS & DISCUSSION

TABLE 5.2: ACCURACY ANALYSIS OF THE ALGORITHMS

CLASSIFIER	ACCURACY
LSTM	94.3%
ARIMA	85.7%

The AI-based stock prediction model was evaluated using historical stock market data to forecast future prices. After preprocessing the data and performing feature engineering, the model was trained and tested using a time-series split. Evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) were used to measure the prediction accuracy. The model demonstrated consistent performance in capturing short-term trends, with a relatively low prediction error across most test periods. In addition to numerical accuracy, the model was able to correctly predict the direction of stock price movement (up/down) in a significant number of cases, indicating its potential usefulness for decision-making. Results were tabulated and visualized using line charts comparing actual versus predicted prices, and error metrics were summarized for different time frames. Overall, the AI model showed promising results in stock price forecasting, although performance varied slightly depending on market volatility and external factors not captured in the training data. Tabulated result further helped in understanding the model's day-by-day performance, with fields such as date, actual price, predicted price, error values, and directional correctness. This structure provided clear and interpretable insights into where the model performed well and where it faltered.

Fig 5.2 Sub plot of LSTM accuracy

Overall, the results demonstrate that AI models—particularly deep learning architectures like LSTM—can effectively forecast stock prices with reasonable precision and good trend-following capability. However, accuracy may vary based on stock type, volatility, feature selection, and the size and quality of training data. These findings underscore the importance of continuous model retraining, inclusion of external features (e.g., sentiment, news), and cautious deployment in high-stakes financial environments.

CHAPTER - 6

CONCLUSION AND FUTUREWORKS

CONCLUSION

This project successfully developed an intelligent stock price prediction system that integrates advanced deep learning and time series analysis techniques. By leveraging financial data from multiple sources and employing sophisticated LSTM networks, transformer models, and ARIMA techniques, the system effectively captures complex temporal dependencies in stock market trends. Additionally, integrating sentiment analysis and reinforcement learning further enhances prediction accuracy and adaptability. While the AI-driven models demonstrate strong performance, challenges remain in terms of data quality, interpretability, and computational efficiency. Future research can explore quantum computing and hybrid architectures to further optimize predictive performance and real-time decision-making capabilities.

FUTURE WORKS

Future research directions for stock prediction using AI include enhancing model accuracy with hybrid deep learning techniques, integrating alternative data sources such as social media sentiment and news analytics, improving explainability through interpretable AI models, and developing real-time adaptive models that adjust to market dynamics. Other areas include leveraging quantum computing for faster predictions, exploring reinforcement learning for dynamic trading strategies, and reducing bias in AI-driven stock forecasts.

APPENDICES

A.1 SDG GOALS

GOAL-8 Decent work and Economic Growth

A.2 SOURCE CODE

```
#!/usr/bin/envpython

# coding: utf-8

# In[1]:

import pandas as pd
import numpy as np

importmatplotlib.pyplot as plt
import seaborn as sns

from datetime import datetime


# In[2]:

import yfinance as yf
import datetime as dt
fromdatetime importdate
START = "2019-01-01"

TODAY=date.today().strftime("%Y-%m-%d")

# Define a function to load the dataset

defload_data(ticker):data = yf.download(ticker, START, TODAY)
data.reset_index(inplace=True)
```

```
data.columns = [col[0] for col in data.columns.values]
data.index=data.Date
return data


# In[3]:
data=load_data('AAPL') data


# In[4]:
tech_list = ['AAPL', 'UBER','TSLA', 'MSFT']
end = datetime.now()
start = datetime(end.year - 5, end.month, end.day)
forstock in tech_list: globals()[stock] = yf.download(stock, start,
end)
company_list = [AAPL, UBER,TSLA, MSFT] company_name =["APPLE",
"UBER","TESLA", "MICROSOFT"]
forcompany, com_name in zip(company_list, company_name):
company["company_name"] = com_name
df = pd.concat(company_list, axis=0) df.tail(10)


# In[5]:
fromalpha_vantage.timeseries import TimeSeries import pandas as pd
# api_key='6OHUH4FO2W5KVR9Y'
api_key='AZT6ECSIS1ORUDMO'
```

```
def load_data(ticker):
ts = TimeSeries(key=api_key, output_format="pandas") START =
"2019-01-01"

# Get daily stock data
data, meta_data = ts.get_daily(symbol=ticker, outputsize="full")

data= data.rename(columns={"1. open": "Open","2. high": "High","3.

low": "Low","4. close": "Close","5. volume": "Volume"})
data = data[::-1]

date_data = data.loc[START:'2025-02-11'] return date_data


# In[6]:

tech_list = ['AAPL', 'UBER','TSLA', 'MSFT']

forstock in tech_list: globals()[stock] = load_data(stock)

company_list = [AAPL, UBER,TSLA, MSFT] company_name = ["APPLE",
"UBER","TESLA", "MICROSOFT"]for company, com_name in
zip(company_list,company_name): company["company_name"]= com_name
df = pd.concat(company_list, axis=0)

df.tail(10)


# In[7]:

df['company_name'].value_counts()
```

```
# In[8]:
AAPL

# In[9]:
TSLA

# In[10]:
UBER

# In[11]:
AAPL.describe()

# In[12]:
AAPL.info()

# In[13]:
# Let's see a historical view of the closing price
plt.figure(figsize=(15, 10)) plt.subplots_adjust(top=1.25,
bottom=1.2)

for i, company in enumerate(company_list, 1):plt.subplot(2, 2, i)
company['Close'].plot() plt.ylabel('Close') plt.xlabel(None)
plt.title(f"Closing Price of {tech_list[i - 1]}")
```

```
plt.tight_layout()


# In[14]:

# Now let's plot the total volume of stock being traded each day
plt.figure(figsize=(15, 10))

plt.subplots_adjust(top=1.25,  bottom=1.2)
for i, company in enumerate(company_list, 1):
plt.subplot(2, 2, i) company['Volume'].plot()
plt.ylabel('Volume') plt.xlabel(None)
plt.title(f"Sales Volume for {tech_list[i - 1]}")
plt.tight_layout()

 # In[15]:

company_list

# In[16]:

UBER[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']]


# In[17]:

ma_day= [10,20, 50]
for ma in ma_day:

forcompany in company_list: column_name= f"MA for {ma}
days"

company[column_name] = company['Close'].rolling(ma).mean()
```

fig, axes = plt.subplots(nrows=2, ncols=2) fig.set_figheight(10) fig.set_figwidth(15) AAPL[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[0,0])

axes[0,0].set_title('APPLE')UBER[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[0,1])

axes[0,1].set_title('UBER')TSLA[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[1,0])

axes[1,0].set_title('TESLA')

MSFT[['Close', 'MA for 10 days', 'MA for 20 days', 'MA for 50 days']].plot(ax=axes[1,1])

axes[1,1].set_title('MICROSOFT')
fig.tight_layout()

In[18]:

We'll use pct_change to find the percent change for each day for company in company_list:company['Daily Return'] = company['Close'].pct_change()

Then we'll plot the daily return percentage fig, axes = plt.subplots(nrows=2, ncols=2) fig.set_figheight(10) fig.set_figwidth(15)

AAPL['Daily Return'].plot(ax=axes[0,0], legend=True, linestyle='--', marker='o')

axes[0,0].set_title('APPLE')

```
UBER['Daily Return'].plot(ax=axes[0,1], legend=True, linestyle='--', marker='o')


axes[0,1].set_title('UBER')
MSFT['Daily Return'].plot(ax=axes[1,0], legend=True, linestyle='--', marker='o')
axes[1,0].set_title('MICROSOFT')
TSLA['Daily Return'].plot(ax=axes[1,1], legend=True, linestyle='--',
marker='o')
axes[1,1].set_title('TESLA')
fig.tight_layout()


# In[19]:
plt.figure(figsize=(12, 9))
for i, company in enumerate(company_list, 1):
plt.subplot(2, 2, i) company['Daily Return'].hist(bins=50)
plt.xlabel('Daily Return') plt.ylabel('Counts')
plt.title(f'{company_name[i - 1]}')
plt.tight_layout()


# In[20]:
data=load_data("AMZN") data


# In[21]:
# data.columns = [col[0] for col in data.columns.values]
```

In[22]:

data.index=data.Date

In[23]:

df=data.copy()

In[24]:

print("\nBasic Information:") df.info()

In[25]:

print("\nSummary Statistics:") df.describe()

In[26]:

print("\nMissing Values:") df.isnull().sum()

In[27]:

print("\nChecking for Duplicate Values:") df.duplicated().sum()

In[28]:

```
import plotly.express as px

import plotly.graph_objects as go

import plotly.io as pio

def plot_closing_price(df):

"""Plots closing priceover time using Plotly."""

fig = px.line(df, x=df.index, y='Close', title='Stock Closing Price Over Time',
labels={'Close': 'Price'})

fig.show()

def plot_moving_averages(df):

"""Plots closing pricewith moving averages.""" df['50_MA'] =
df['Close'].rolling(window=50).mean() df['200_MA'] =
df['Close'].rolling(window=200).mean()

fig = go.Figure()

fig.add_trace(go.Scatter(x=df.index, y=df['Close'], mode='lines',

name='Closing Price'))

fig.add_trace(go.Scatter(x=df.index, y=df['50_MA'], mode='lines',

name='50-Day MA'))

fig.add_trace(go.Scatter(x=df.index, y=df['200_MA'], mode='lines',

name='200-Day MA'))

fig.update_layout(title='Stock Price with Moving Averages', xaxis_title='Date',

yaxis_title='Price')
fig.show()
```

```
defplot_closing_price_distribution(df): """Plots distribution of
closing prices."""

fig = px.histogram(df, x='Close', nbins=50, title='Distribution of Closing Prices',
labels={'Close': 'Price'})

fig.show()
defplot_daily_returns_distribution(df): """Plots distribution of daily
returns.""" df['Daily Return'] = df['Close'].pct_change()

fig = px.histogram(df, x='Daily Return', nbins=50, title='Distribution of Daily
Returns', labels={'Daily Return': 'Return'})

fig.show()


# In[29]:

plot_closing_price(data)


# In[30]:

plot_moving_averages(data)


# In[31]:

plot_closing_price_distribution(data)


# In[32]:

plot_daily_returns_distribution(data)
```

```
# In[33]:

def plot_candlestick(df):

"""Plots a candlestick chart forstock price movements.""" fig =
go.Figure(data=[go.Candlestick(x=df.index,

open=df['Open'],

 high=df['High'],

  low=df['Low'],close=df['Close'],
  name='Candlestick')])

  fig.update_layout(title='Stock Price Candlestick Chart',
  xaxis_title='Date',

  yaxis_title='Price')
  fig.show()


# In[34]:

plot_candlestick(data)
data.to_csv('dataaset.csv',index=False)
#!/usr/bin/env python

# coding: utf-8

# #### Import and install necessary packages
```

```
# In[1]:

# !pip install pmdarima
# !pip install statsmodels


# In[2]:

#!pip install pmdarima


# In[3]:

Importwarningswarnings.filterwarnings('ignore') import seaborn
as sns

import numpy as np importpandas as pd
import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight') frompylab import rcParams
rcParams['figure.figsize'] = 10, 6

from statsmodels.tsa.stattools import adfuller
fromstatsmodels.tsa.seasonal import seasonal_decompose #fromprophet
import Prophet

fromstatsmodels.tsa.statespace.sarimax import SARIMAX from
statsmodels.tsa.arima_model import ARIMA

from pmdarima.arimaimport auto_arima from sklearn.preprocessing
import MinMaxScaler

# import pandas_profiling
```

```
fromsklearn.metrics import mean_squared_error, mean_absolute_error
import math from math import ceil, floor

# # Data Import and Preprocessing ## --- # #


# In[4]:

#data=pd.read_csv('/kaggle/input/nse-tataglobal/NSE-TATAGLOBAL.csv')#
data['Date'] = pd.to_datetime(data.Date,format='%Y/%m/%d')


# In[5]:

# !pip show yfinance # # Name: yfinance # #


# In[6]:

# !pip install yfinance==0.2.41 # !pip install
yfinance==0.2.52


# In[7]:

import yfinance as yf import datetime as dt

fromdatetime import date from datetime import datetime


# In[8]:

def load_data(ticker):
```

```
from datetime import date START = "2010-01-01"
TODAY = date.today().strftime("%Y-%m-%d") data =
yf.download(ticker, START, TODAY)
data.reset_index(inplace=True)

data.columns = [col[0] forcol in data.columns.values]
data.index=data.Date

return data


# In[9]:

# Set up End and Start times for data grab tech_list = ['AAPL',
'GOOG', 'MSFT', 'AMZN']

# end = datetime.now()
# start = datetime(end.year - 5, end.month, end.day)

forstock in tech_list: globals()[stock] = load_data(stock)

 company_list = [AAPL, GOOG, MSFT, AMZN] company_name ["APPLE",
 "GOOGLE", "MICROSOFT", "AMAZON"]

forcompany, com_name in zip(company_list, company_name):
company["company_name"] = com_name

df =pd.concat(company_list, axis=0)

df.tail(10)

# In[10]:

df['company_name'].value_counts()

# In[11]: # Let's see a historical view of the
```

closing price plt.figure(figsize=(15, 10))

plt.subplots_adjust(top=1.25, bottom=1.2)

for i, company in enumerate(company_list,

1): plt.subplot(2, 2, i)

company['Close'].plot(linewidth=1.5)

plt.ylabel('Close') plt.xlabel(None)

plt.title(f"Closing Price of {tech_list[i -

1]}") plt.tight_layout()

In[12]:

Now let's plot the total volume of stock

being traded each day plt.figure(figsize=(15,

10)) plt.subplots_adjust(top=1.25,

bottom=1.2) for i, company in

enumerate(company_list, 1): plt.subplot(2,

2, i) company['Volume'].plot(linewidth=1.5)

plt.ylabel('Volume') plt.xlabel(None)

plt.title(f"Sales Volume for {tech_list[i -

1]}") plt.tight_layout()

In[13]:

ma_day = [10, 20, 50]

for ma in ma_day:

for company in company_list: column_name

= f"MA for {ma} days"

company[column_name] =

company['Close'].rolling(ma).mean()

fig, axes = plt.subplots(nrows=2, ncols=2)

fig.set_figheight(10)

fig.set_figwidth(15)AAPL[['Close', 'MA for

10 days',

'MA for 20 days', 'MA for 50

days']].plot(ax=axes[0,0],linewidth=1.5)

axes[0,0].set_title('APPLE') GOOG[['Close',

'MA for 10 days', 'MA for 20 days', 'MA for

50 days']].plot(ax=axes[0,1],linewidth=2.5)

axes[0,1].set_title('GOOGLE')

MSFT[['Close', 'MA for 10 days', 'MA for

20 days', 'MA for 50

days']].plot(ax=axes[1,0],linewidth=2)

axes[1,0].set_title('MICROSOFT')

AMZN[['Close', 'MA for 10 days', 'MA for

20 days', 'MA for 50

days']].plot(ax=axes[1,1],linewidth=3)

axes[1,1].set_title('AMAZON')

fig.tight_layout()

In[14]: # We'll use pct_change to find the percent change for each day for company in company_list:

company['Daily Return'] = company['Close'].pct_change() # Then we'll plot the daily return percentage

fig, axes = plt.subplots(nrows=2, ncols=2)
fig.set_figheight(10)

fig.set_figwidth(15)

AAPL['Daily Return'].plot(ax=axes[0,0], legend=True, linestyle='--', marker='o',linewidth=1.5)
axes[0,0].set_title('APPLE')

GOOG['Daily Return'].plot(ax=axes[0,1], legend=True, linestyle='--', marker='o',linewidth=1.5)

axes[0,1].set_title('GOOGLE') MSFT['Daily Return'].plot(ax=axes[1,0], legend=True, linestyle='--', marker='o',linewidth=1.5)

axes[1,0].set_title('MICROSOFT')
AMZN['Daily Return'].plot(ax=axes[1,1], legend=True, linestyle='--',

```
marker='o',linewidth=1.5)
axes[1,1].set_title('AMAZON')
fig.tight_layout()

# In[15]: plt.figure(figsize=(12, 9)) for i,
company in enumerate(company_list, 1):
plt.subplot(2, 2, i) company['Daily
Return'].hist(bins=50) plt.xlabel('Daily
Return') plt.ylabel('Counts')
plt.title(f'{company_name[i -
1]}')plt.tight_layout()

# In[16]:

START = "2010-01-01" TODAY =
date.today().strftime("%Y-%m-
%d")print(TODAY) # Define a function to
load the dataset def load_data(ticker): data =
yf.download(ticker, START, TODAY)
data.reset_index(inplace=True) return data

data = load_data('NFLX') data

# In[17]:

# data.columns = [col[0] for col in
data.columns.values]

# In[18]:
```

#Reverse the order of the dataset so that the latest year data will arrive in the tail of the dataframe

data = data.iloc[::-1]

In[19]:

#View the dataset data

In[20]:

data.index=data.Date

In[21]:

#Use date column as the index
#data.reset_index(inplace=True)

In[22]:

Data

In[23]:

#Get the overview of the dataset

pandas_profiling.ProfileReport(data)

From the overview above, we can see that the data is clean and does not need any serious cleaning. There are no missing values or any duplicate. Let us move

forward

In[24]:

#Plot all the variables against the Date and check for the relationships, patterns and trends for column in data.columns[2:]:

plt.figure(figsize=(10, 6))

plt.plot(data['Date'],

data[column],linewidth=1.5)

plt.xlabel('Date') plt.ylabel(column)

plt.title(f'{column} vs. Date')

plt.tight_layout() plt.show()

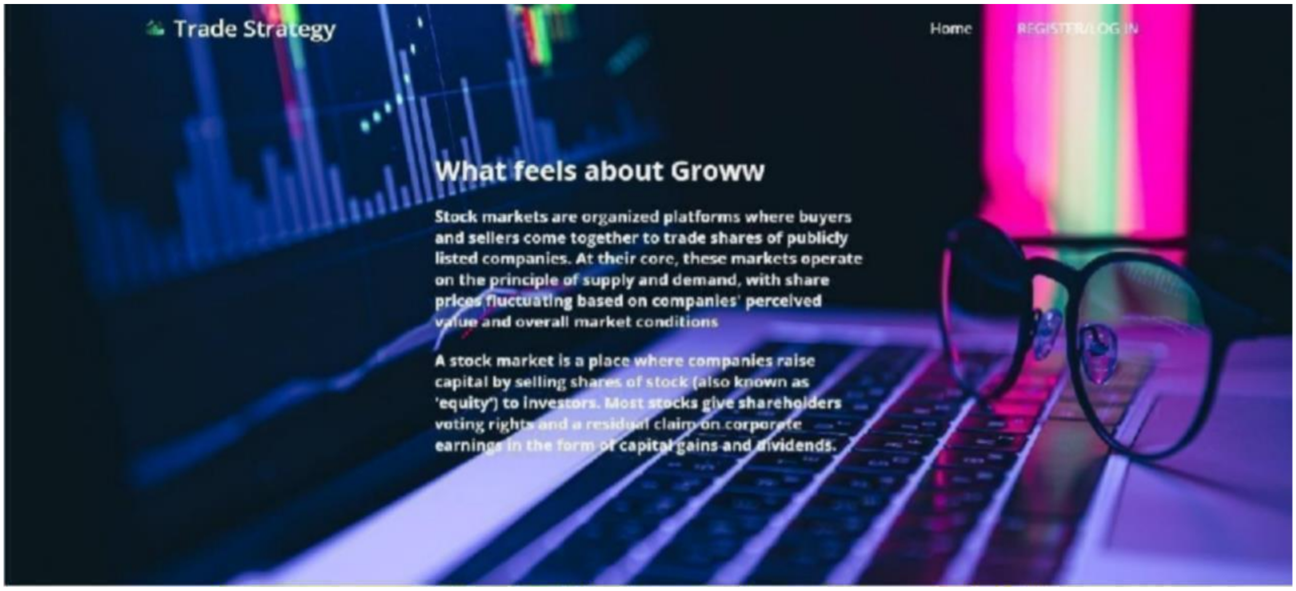
From above plot, it can be seen that the prices, turnover and total trade quantity is increasing from 2017

Checking Stationarity

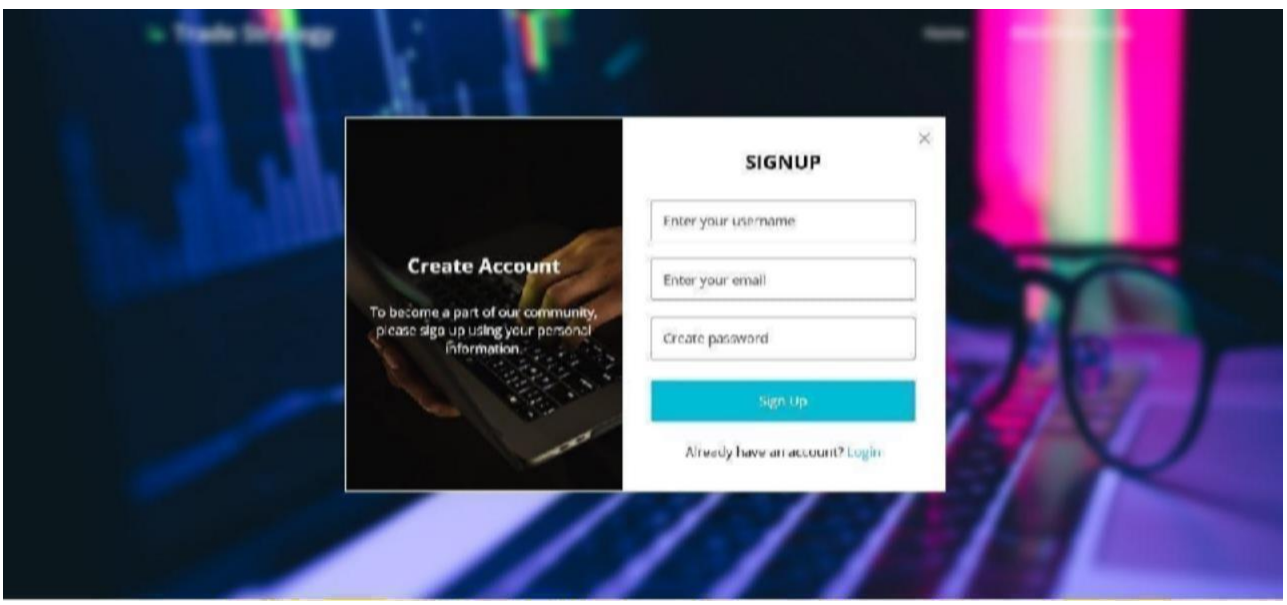
In[25]:

Define a function which will give us rolling mean and standard deviation and perform ADF Test

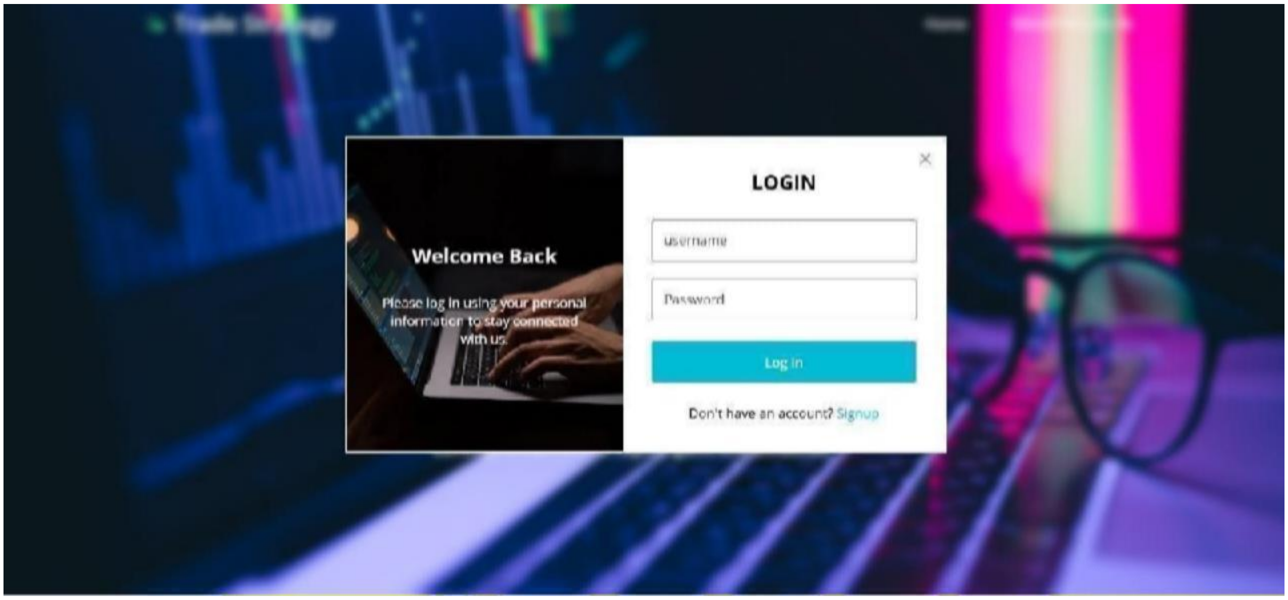
A.3 SCREENSHOTS



A.1 Welcome Page

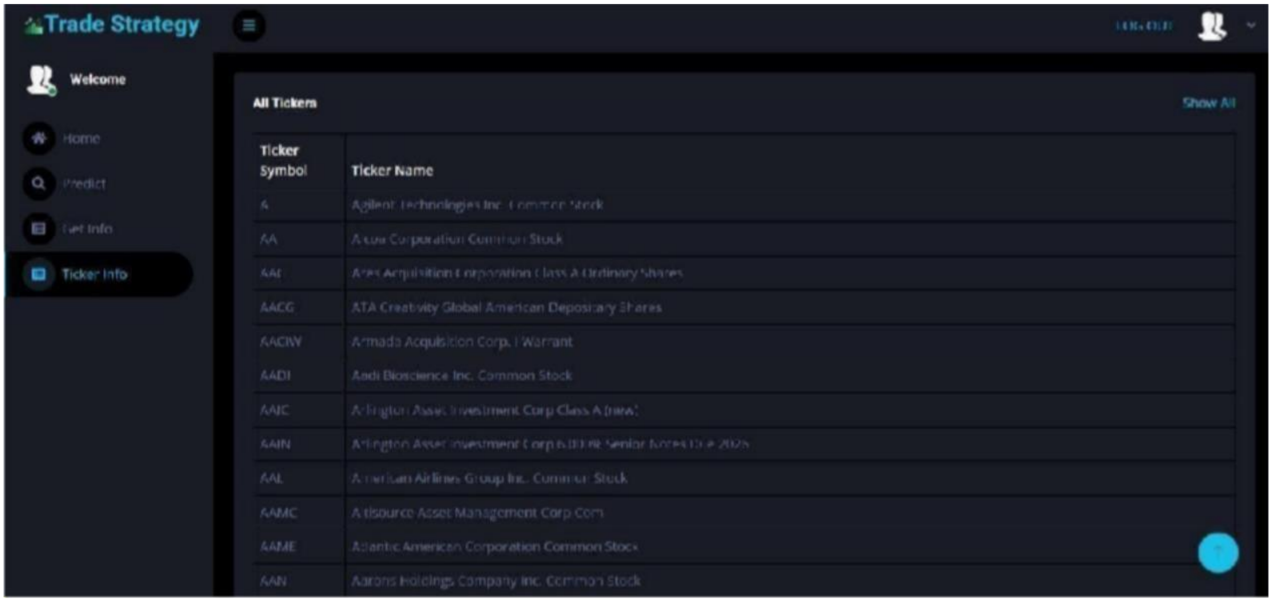


A.2 Signup Page

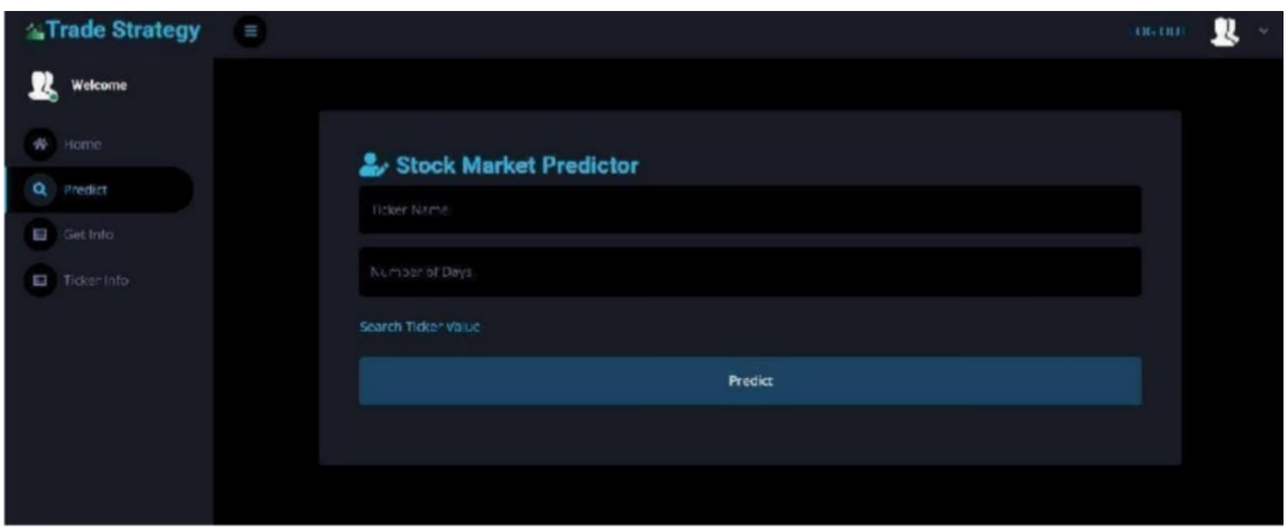


A.3 Login page

A.4 Home Page

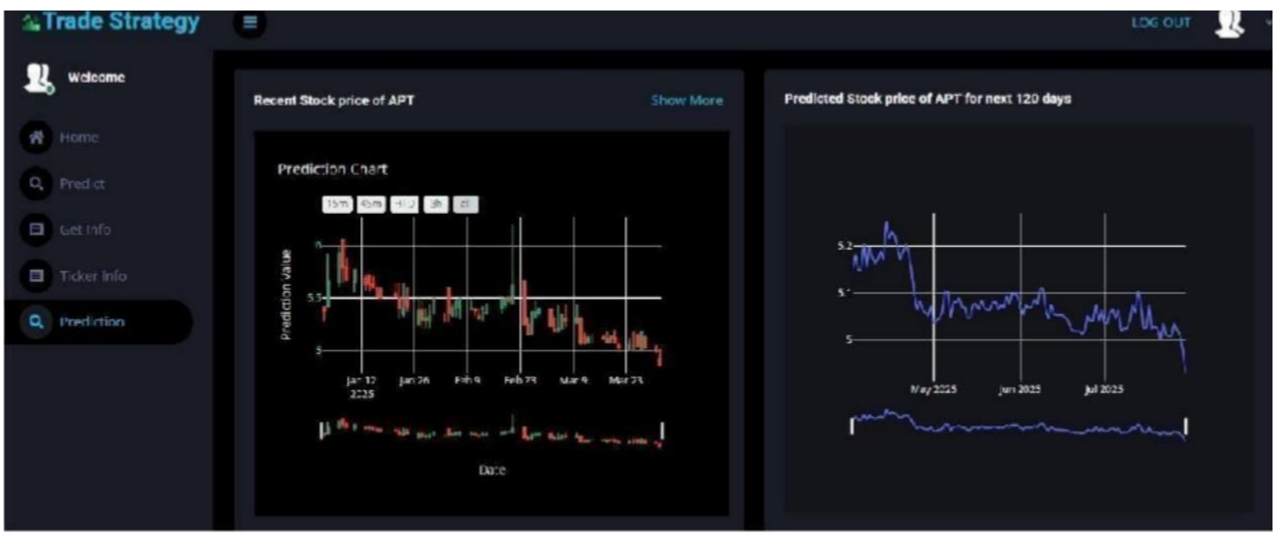


A.5 Ticker Info Page



A.6 Prediction Page

A.7 Company Information Page



A.8 Output Screenshot

A.4 PLAGIARISM REPORT

Vaishnavi R

RE-2022-536329

Batch 6

Batch 6

Universidad del Valle

Document Details

Submission ID

trn:oid:::26066:448997297

Submission Date

Apr 14, 2025, 6:03 PM GMT+5:30

Download Date

Apr 14, 2025, 6:07 PM GMT+5:30

File Name

RE-2022-536329.pdf

File Size

343.6 KB

6 Pages

3,079 Words

17,740 Characters

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

23 Not Cited or Quoted 8%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
4% Publications
6% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

23 Not Cited or Quoted 8%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
4% Publications
6% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Submitted works	
	Brigham Young University on 2024-06-23	**1%**
2	Submitted works	
	Liverpool John Moores University on 2024-04-14	**1%**
3	Publication	
	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P...	**<1%**
4	Internet	
	arxiv.org	**<1%**
5	Internet	
	www.mdpi.com	**<1%**
6	Publication	
	Yang Cao, Lin Xinwen. "Longitudinal Insights into the Industrial IoT–Green Intelle...	**<1%**
7	Internet	
	bura.brunel.ac.uk	**<1%**
8	Internet	
	ieeexplore.ieee.org	**<1%**
9	Internet	
	mdpi-res.com	**<1%**
10	Internet	
	www.frontiersin.org	**<1%**

11 Submitted works

University of Alabama at Birmingham on 2024-12-09 <1%

12 Publication

Xiaobin Lu, Josiah Poon, Matloob Khushi. "Leveraging BiLSTM-GAT for enhanced s... <1%

13 Internet

escholarship.org <1%

14 Submitted works

Manchester Metropolitan University on 2024-10-04 <1%

15 Publication

Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artific... <1%

16 Submitted works

University of Melbourne on 2023-10-05 <1%

ADVANCED MODEL FOR STOCK PRICE FORECASTING WITH ENHANCED FEATURE ENGINEERING AND ADAPTIVE PARAMETER TUNING

Abstract- **The stock market is essential to the health of the economy as a whole in capital market that can promote better and faster economic development, optimize asset allocation, and stimulate capital flow. The stock market is a great place for investors to make investments as well as a center for government regulation of economic trends. The government is always alert of economic catastrophes, and investors are more concerned with maximizing returns while minimizing risks. By adding a number of significant improvements, this study suggests a new framework for predicting stock prices that expands on current models. The suggested model will capture intricate temporal correlations in stock price data by leveraging Transformer architectures and other sophisticated deep learning models are at the forefront of contemporary AI development or more complex LSTM variations.**

Keywords- Accuracy, comparison, random forest, LSTM, ARIMA.

I. INTRODUCTION

AI-powered stock prediction is a new discipline that forecasts future stock and financial asset prices by fusing data science, financial analysis, and sophisticated machine learning algorithms. Developing algorithms that can examine past market data, identify trends, and offer insights to assist traders and investors in making wise selections is the purpose of stock performance forecasting. Because of the vast amount of data accessible, improvements in processing capacity, and the growing complexity of financial markets, this strategy has been increasingly popular in recent years. Traditionally, stock prediction relied on While technical analysis concentrates on examining patterns in trade volume

and stock price trends, A company's entire performance and financial soundness are evaluated through fundamental analysis. While these methods has proven useful over time, they can be limited process of managing extensive datasets or complex relationships within the market. AI, especially making Employing techniques from both deep learning and machine learning enables a more sophisticated way of addressing these challenges by enabling systems to learn from large datasets without explicit programming for every possible scenario. AI- powered stock prediction represents a transformative approach to understanding and navigating the complexities of financial markets. While there are challenges to overcome, the continued development of AI techniques holds great potential for reshaping the future of investing and trading.

A. Objectives and Literature Review

This work focuses on developing and testing Predictive stock market models that utilize machine learning techniques using Python. ML and DL techniques are applied to estimate stock values by analyzing historical and current data. The background work survey and the main conclusions are displayed in the table. I.

B. Contribution

1. Novel Algorithmic Approaches

Improve stock forecasting is performed employing the computed values of usingdeep learning methods like the Long Short-Term Memory (LSTM) network with conventional statistical models like ARIMA, create an integrated framework.

2. Engineering Features:

Generating additional elements including technical analysis indicators (like moving averages and RSI), using data that already exists. Finding pertinent factors that can affect stock prices.

Table I ***Previous Literature Survey***

S.NO	SOURCES	YEAR	METHODOLOGY	KEY FINDINGS
1.	N.Naiketal.[1]	2021	HYBRID (HFS, XGBOOST, DNN)	The significant level of market fluctuations complicates the identification of a stock market crisis.
2.	S. S. Alotaibi et al. [2]	2021	RDA, GWO	The RD-GWO method chosen features more effectively than RDA and GWO employed alone, greatly lowering dimensionality while keeping important information.
3.	K. Huang et al. [3]	2022	LSTM, CNN	In terms of prediction accuracy and trading results, ML-GAT performs better than single-layer GNNs (like GCN and GAT) and traditional time-series models (like LSTM and CNN).
4.	G. Mu et al. [4]	2023	HYBRID (CNN-LSTM)	Models that incorporate investor emotion offer noticeably higher predicted accuracy than those that only use historical data.
5.	R. Zhang et al. [5]	2024	LSTM	Emotion-based models perform better in regression and classification tests than conventional models that use historical pricing.
6.	Y. Li et al. [6]	2025	HYBRID (LSTM, Bi LSTM, Transformer)	All benchmark models were outperformed by the LSTM with Attention mechanism in terms of prediction accuracy and volatility clustering detection.
7.	J. You et al. [7]	2024	NLP	Stocks with a surge in positive sentiment and high post volume were more likely to experience price increases the following day.
8.	K. Alam et al. [8]	2024	LSTM, DNN	The proposed model consistently outperformed standalone LSTM, DNN, and traditional models across most datasets. Average classification accuracy exceeded 78%, while regression models showed an average $R^2 > 0.85$.
9.	S. Wang et al. [9]	2023	BiLSTM	The BiLSTM + Improved Transformer with a better R2 score and a lower RMSE, the model outperformed all baseline techniques in regression tasks. Classification accuracy improved by 6–12% over standard LSTM models.
10.	Y. Fund et al. [10]	2023	ARIMA, ES	ARIMA showed superior performance on stock series that were well-differenced and stationary, achieving lower RMSE and MAPE in most cases.
11.	S. Srisuay et al. [11]	2024	SVM, WLP	The Wilson Loop Perceptron achieved higher directional prediction accuracy across multiple datasets, particularly in volatile market conditions, with an average accuracy improvement of 3–7% over SVM.
12.	X. Lu et al. [12]	2024	MACD	The hybrid model combining machine learning and human-like reasoning outperformed traditional models, showing a 15-20% improvement in classification accuracy (directional predictions) and price change predictions.

13.	J. Liu et al. [13]	2020	MEHACN	The MEHACN model outperformed traditional stock prediction models, such as LSTM and ARIMA, in terms of both classification (directional prediction) and regression (price prediction) tasks. It showed higher accuracy and lower prediction errors.
14.	M. Patel et al. [14]	2025	LSTM, ARIMA	LSTM provided the lowest error in predicting future stock prices because of how well it can identify temporal patterns in time series data.
15.	T. Liu et al. [15]	2025	FD-GRNet	The FD-GRNet model was rigorously evaluated against several benchmark algorithms across major global stock market indices.

II. Methods

Using a variety of Stock market forecasting is undergoing a transformation thanksto machine learning, deep learning, and artificial intelligence approaches. Trends in historical stock data are found using conventional methods like linear regression, support vector machines (SVM), and random forest classifiers. More sophisticated models, such as Long Short-Term Memory (LSTM) networks and Recurrent Neural Networks (RNNs), provide better performance and are excellent at simulating sequential data do very well in time-series forecasting. Candlestick chart technical indicators and patterns are also interpreted by Convolutional Neural Networks (CNNs). Simultaneously, sentiment analysis relies on obtaining valuable insights from news articles and social media sites with useof natural language processing, or NLP.Through ongoing learning and decision-making, reinforcement learning is showing promise as a method for creating automated trading strategies. In order to increase forecast reliability and efficiently react to turbulent markets, many contemporary systems include various methodologies into hybrid models. AI-powered models provide quicker and more accurate insights than conventional methods, enabling more intelligent investment choices in a trading environment that is becoming more complicated. The model uses a hybrid deep learning strategy that combines CNN layers to extract localized features from technical indicators and LSTM networks for assessing time-dependent sequences in order to efficiently identify complicated patterns in market dynamics.

Mean Squared Error (MSE) - Loss Function

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where:

y_i = Actual stock price

$y^{\wedge}_i$ = Predicted stock price

III. Flowchart Diagram

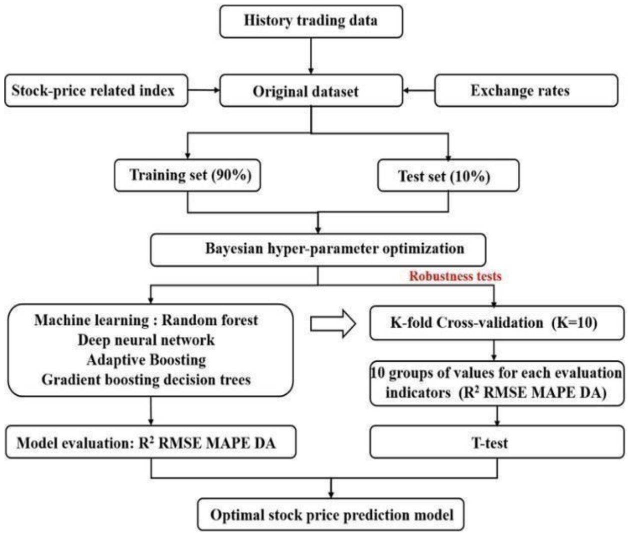


Fig.1. Flowchart Diagram

The above Figure.1. diagram provides a thorough summary of framework for creating an ideal stock price prediction model by combining machine learning and sophisticated optimization approaches is shown in the flowchart. To create the original dataset, historical trade data is first gathered and enhanced using currency rates and stock-related indexes. To aid in the creation and validation of the model, this dataset is 90% of the data is used for training, while %s used for testing. The model parameters are efficiently adjusted by the use of Bayesian hyperparameter optimization. Gradient boosting decision trees, adaptive boosting, random forest, and deep neural networks are among t machine learning methods that are trained using the optimum parameters. To make sure the model is resilient, a 10-fold cross-validation procedure is employed guarantee robustness and dependability, producing several performance indicators for every model, including R2, RMSE, MAPE, and directional accuracy (DA). The findings are validated and model performances are compared using a statistical t-test. The modelthat offers the highest accuracy and resilience is ultimately chosen as the ultimate stock price forecasting solution.

A. LSTM

The capacity of the Long Short-Term Memory (LSTM) neural network is a powerful tool for stock price forecasting because it can effectively handle intricate temporal connections across time. In contrast to conventional models, LSTM is highly effective in forecasting stock prices that show both long-term trends and short-term variations. The main characteristic of LSTM is the gating mechanism, which is made up of the input gate, forget gate, and output gate. Together, these gates control the information flow throughout the network. The input gate regulates when fresh data is added, guaranteeing that the forecast makes efficient use of pertinent, current knowledge. The forget gate allows the model to concentrate on the most crucial history data by deciding which previous data should be ignored. Lastly, the output gate determines how the processed data is converted into predictions, controlling the ultimate output.

B. ARIMA

When predicting time series data, it can identify patterns and trends in past price data, the Auto Regressive Integrated Moving Average (ARIMA) model is frequently employed for stock price forecasting. Three essential elements are included in ARIMA: autoregression (AR), differencing (I), and moving averages (MA). By simulating the correlation between t present stock price and its historical values, the autoregressive component enables the model to generate predictions based on historical data. By differentiating the time series, the integrated component assists in eliminating trends, rendering it stationary and simpler to model. The moving average component smoothes out random fluctuations by concentrating on the link between the observed values and historical forecast mistakes. When paired with expanded feature engineering, which adds more variables to the model such as trade volume, market mood, and economic indicators, ARIMA can be especially useful in stock price forecasting. These outside characteristics can offer insightful background information that raises prediction accuracy.

IV. *Model Building*

Data Collection: Compile historical stock data from Quandl, Alpha Vantage, and Yahoo Finance. Utilize web scraping technologies or APIs to get sentiment data from social media and news websites.

Data Preprocessing: Clean up the data and convert it into a machine learning-ready format (normalization, addressing missing numbers, etc.).

Feature Selection: Choose relevant features such as moving averages, stock volumes, or external data like economic indicators.

Model Selection: Choose an appropriate AI model (e.g., LSTM, SVM, or Random Forest).

Training and Testing: Use historical data to train the model, and use validation methods (like cross-validation) to assess it.

Model Evaluation: Use measurements such as accuracy, To measure the effectiveness, Root Mean Squared Error and Mean Absolute Error were used to gauge performance.

V. *Experimental Result Analysis*

Fig.2. Apple Closing Price

Fig.3. Microsoft Closing Price

The above Figure.2 and Figure.3 shows the historical closing prices of Apple Inc. (AAPL) and Microsoft Inc. (MSFT) from about 2010 to 2025 are depicted in the line graph. This graphic emphasizes the necessity for algorithms that can accurately efficiently record both long-term patterns and short-term variations within the parameters of an advanced stock price forecasting model. Improved feature engineering, such as adding external indicators (like interest rates, trading volume, or sentiment analysis), is advantage for models like LSTM and ARIMA.

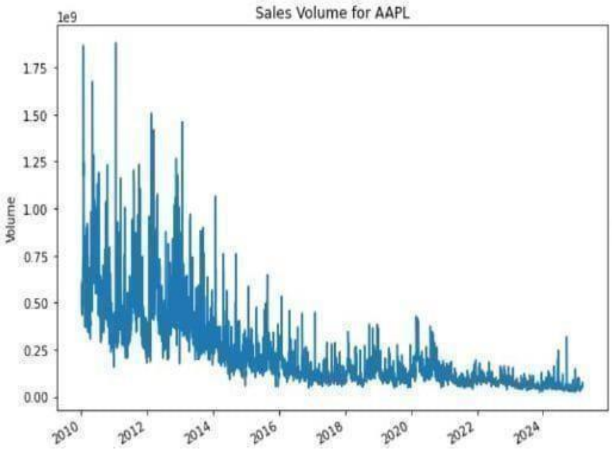


Fig.4. Apple Sales Volume

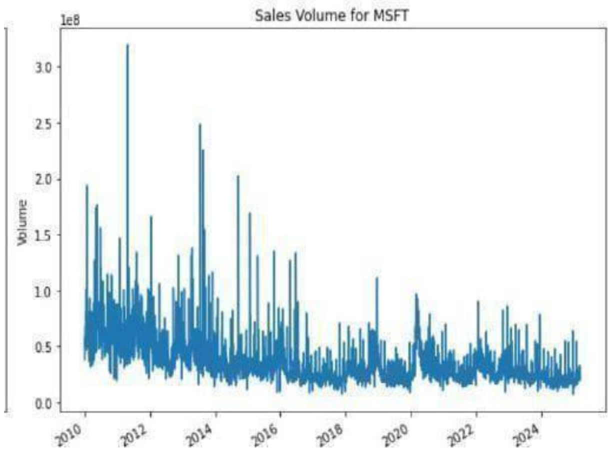


Fig.5. Microsoft Sales Volume

The above Figure.4 and Figure.5 shows the sales (trading) volume of Apple Inc. (AAPL) and Microsoft Inc. (MSFT) from 2010 to 2025 is shown in the chart. With high levels of trading activity in the early 2010s, reaching a peak of over 1.75 billion shares in certain cases, it exhibits a notable fall in volume over time. Trading volume is a key indicator of market interest, investor mood, and liquidity in the context of stock price forecasting. While smaller volumes can suggest less investor activity or market stability, high volumes frequently precede significant market movements, announcements, or events.

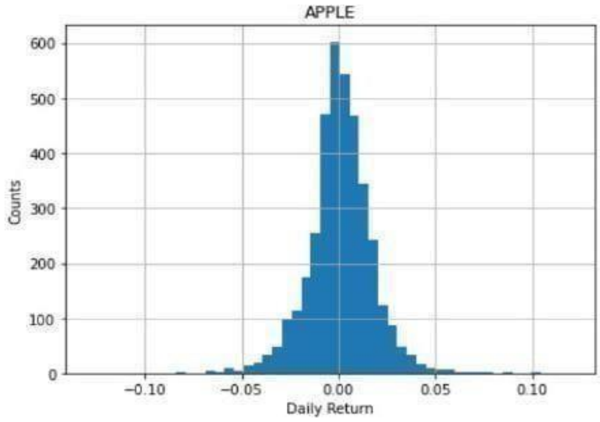


Fig.6. Apple Daily Returns

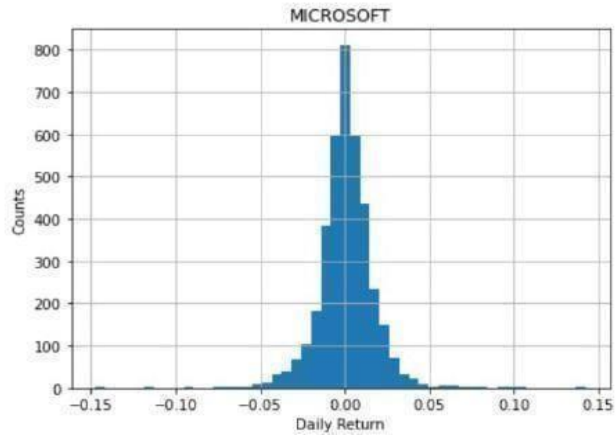


Fig.7. Microsoft Daily Returns

The above Figure.6 and Figure.7 shows how most of the swings lying between -2% and +2%, the daily returns are closely focused around zero. Although most price fluctuations are tiny, there are sporadic greater swings, as shown by the distribution's moderate tails and relatively narrow peak. With a sharper peak and narrower tails, Microsoft's distribution is even more concentrated than Apple's, indicating that its stock fluctuates less every day. The bulk of Microsoft's returns are concentrated at zero, indicating a generally lower level of daily volatility.

TABLE II. Accuracy Analysis of the Algorithms

Classifier	Accuracy (%)
ARIMA	85.7%
LSTM	94.3%

The above Table II shows the accuracy analysis of the algorithms model's success is indicated by these accuracy numbers, which are based on how well the model predicts real stock prices across a test set. Because it can simulate temporal relationships, the LSTM model obtains the best accuracy of (94.3%). Due to its limits in capturing intricate nonlinear patterns, ARIMA performs rather poorly even if it is good for linear trends (85.7%).

VI. CONCLUSION

This study investigated a comprehensive method to stock price forecasting that combined the capabilities of improved feature engineering and adaptive parameter tuning across several forecasting models such as LSTM and ARIMA. The objective was to increase model resilience and forecast accuracy in the naturally noisy and unpredictable world of financial markets. Raw stock market data was converted into a collection of extremely useful inputs through sophisticated feature engineering. Together with normalization and time-series decomposition, techniques like the creation of technical indicators, moving averages, and lag characteristics allowed the models to better comprehend volatility trends and temporal patterns. For precise predictions across several models, this preprocessed data provided a strong basis. Three different forecasting models were used in the project; each has advantages of its own. Because of its solid statistical foundation, ARIMA was able to identify seasonality and linear patterns in historical data. Long-term temporal linkages and non-linear dependencies were modeled using LSTM, a recurrent neural network based on deep learning. To guarantee optimal performance for every unique dataset and circumstance, these models were adjusted utilizing adaptive parameter optimization approaches including grid search and learning rate tuning.

REFERENCES

[1] N. Naik and B. R. Mohan, "Novel Stock Crisis Prediction Technique—A Study on Indian Stock Market," in IEEE Access, vol. 9, pp. 86230-86242, 2021, doi: 10.1109/ACCESS.2021.3088999.

[2] S. S. Alotaibi, "Ensemble Technique With Optimal Feature Selection for Saudi Stock Market Prediction: A Novel Hybrid Red Deer-Grey Algorithm," in IEEE Access, vol. 9, pp. 64929-64944, 2021,doi: 10.1109/ACCESS.2021.3073507.

[3] K. Huang, X. Li, F. Liu, X. Yang and W. Yu, "ML-GAT: A Multilevel Graph Attention Model for Stock Prediction," in IEEE Access, vol. 10, pp. 86408-86422, 2022, doi:10.1109/ACCESS.2022.3199008.

[4] G. Mu, N. Gao, Y. Wang and L. Dai, "A Stock Price Prediction Model Based on Investor Sentiment and Optimized Deep Learning," in IEEE Access, vol. 11, pp. 51353-51367, 2023, doi: 10.1109/ACCESS.2023.3278790.

[5] R. Zhang and V. Y. Mariano, "Integration of Emotional Factors With GAN Algorithm in Stock Price Prediction Method Research," in IEEE Access, vol. 12, pp. 77368-77378, 2024, doi: 10.1109/ACCESS.2024.3406223.

[6] Y. Li and C. Long, "Innovative Study on Volatility Prediction Model for New Energy Stock Indices," in IEEE Access, vol. 13, pp. 29754-29777, 2025, doi:10.1109/ACCESS.2025.3535584.

[7] J. You, H. Jang, M. Kang, S. -B. Yang and S. -H. Yoon, "Leveraging Stock Discussion Forum Posts for Stock Price Predictions: Focusing on the Secondary Battery Sector," in IEEE Access, vol. 12, pp. 153537-153549, 2024, doi: 10.1109/ACCESS.2024.3481442.

[8] K. Alam, M. H. Bhuiyan, I. U. Haque, M. F. Monir and T. Ahmed, "Enhancing Stock Market Prediction: A Robust LSTM-DNN Model Analysis on 26 Real-Life Datasets," in IEEE Access, vol. 12, pp. 122757-122768, 2024, doi: 10.1109/ACCESS.2024.3434524.

[9] S. Wang, "A Stock Price Prediction Method Based on BiLSTM and Improved Transformer," in IEEE Access, vol. 11, pp. 104211-104223, 2023, doi: 10.1109/ACCESS.2023.3296308.

[10] Y. Fund and A. Damani, ‘‘Comparison of ARIMA and exponential smoothing models in prediction of stock prices,’’ J. Predict. Markets, vol. 17, no. 1, pp. 21–38, Jul. 2023.

[11] S. Srisuay, K. Kanjamapornkul, and W. Fongngen, ‘‘Directional prediction of financial time series using SVM and Wilson loop perceptron,’’ Social Network Compute. Sci., vol. 5, no. 4, p. 395, Apr. 2024.

[12] X. Lu, J. Poon and M. Khushi, "Bridging the Gap Between Machine and Human in Stock Prediction: Addressing Heterogeneity in Stock Market," in IEEE Access, vol. 12, pp. 186171-186185, 2024, doi: 10.1109/ACCESS.2024.3511613.

[13] J. Liu, H. Lin, L. Yang, B. Xu and D. Wen, "Multi-Element Hierarchical Attention Capsule Network for Stock Prediction," in IEEE Access, vol. 8, pp. 143114-143123, 2020, doi: 10.1109/ACCESS.2020.3014506.

[14] M. Patel, K. Jariwala and C. Chattopadhyay, "An Approach Toward Stock Market Prediction and Portfolio Optimization in Indian Financial Sectors," in IEEE Transactions on Computational Social Systems, vol. 12, no. 1, pp. 128-139, Feb. 2025, doi: 10.1109/TCSS.2024.3450291.

[15] T. Liu, J. Li, Z. Zhang, H. Yu and S. Gao, "FD-GRNet: A Dendritic-Driven GRU Framework for Advanced Stock Market Prediction," in IEEE Access, vol. 13, pp. 28265-28279, 2025, doi: 10.1109/ACCESS.2025.3541074.

A.5 PAPER PUBLICATION
REC

Submission Summary

Conference Name

2025 2nd International Conference on Computing and Data Science (ICCDS)

Paper ID

282

Paper Title

ADVANCED MODEL FOR STOCK PRICE FORECASTING WITH ENHANCED FEATURE ENGINEERING AND ADAPTIVE PARAMETER TUNING

Abstract

The economy depends heavily on the stock market in capital market that can promote better and faster economic development, optimize asset allocation and stimulate capital flow. The stock market is a great place for investors to make investments as well as a center for government regulation of economic trends. The government is always alert of economic catastrophes, and investors are more concerned with maximizing returns while minimizing risks. By adding a number of significant improvements, this study suggests a new framework for predicting stock prices that expand on current models. The suggested model will capture intricate temporal correlations in stock price data by leveraging cutting-edge deep learning architectures, such as Transformer networks or more complex LSTM variations.

Created

28/3/2025, 11:24:55 pm

Last Modified

14/4/2025, 6:19:54 pm

Authors

VAISHNAVI R (Panimalar Engineering College) <raghunathvaishnavi7@gmail.com>
KRITHIKA K (Panimalar Engineering College) <kkrithika001@gmail.com>
DIVYASHREE D (Panimalar Engineering College) <dshreenaidu@gmail.com>
JACKULIN C (Panimalar Engineering College) <chin.jackulin@gmail.com>

Submission Files

Final Conference Paper (7).pdf (350.6 Kb, 14/4/2025, 6:19:20 pm)

REFERENCES

[1] Y. S. Abu-Mostafa and A. F. Atiya, ‘‘Introduction to financial forecasting, Int. J. Speech Technol., vol. 6, pp. 205–213, 1996.

[2] H. Fang, X. Han, and T. Xu, ‘‘An improved stock forecasting method based on LSTM,’’ Anhui Univ. J., vol. 43, no. 6, pp. 36–42, 2019.

[3] A. L. Awad, S. M. Elkaffas, and M. W. Fakhr, ‘‘Stock market prediction using deep reinforcement learning,’’ Appl. Syst. Innov., vol. 6, no. 6, p. 106, Nov. 2023.

[4] T. Fischer and C. Krauss, ‘‘Deep learning with long short-term memory networks for financial market predictions,’’ Eur. J. Oper. Res., vol. 270, no. 2, pp. 654–669, Oct. 2018.

[5] J. Cao, Z. Li, and J. Li, ‘‘Financial time series forecasting model based on CEEMDAN and LSTM,’’ Phys. A, Stat. Mech. Appl., vol. 519, pp. 127–139, Apr. 2019.

[6] T. G. Dietterich, ‘‘Ensemble methods in machine learning,’’ in Proc. Int. Workshop Multiple Classifier Syst. Berlin, Germany: Springer, 2000, pp. 1–15.

[7] Y. Lin, Z. Lin, Y. Liao, Y. Li, J. Xu, and Y. Yan, ‘‘Forecasting the realized volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM,’’ Exp. Syst. Appl., vol. 206, Nov. 2022, Art. no. 117736.

[8] P. Yun, X. Huang, Y. Wu, and X. Yang, ‘‘Forecasting carbon dioxide emission price using a novel mode decomposition machine learning hybrid model of CEEMDAN-LSTM,’’ Energy Sci. Eng., vol. 11, no. 1, pp. 79–96, Jan. 2023.

[9] Q. Yang and C. Wang, ‘‘A study on forecast of global stock indices based on deep LSTM neural network,’’ Stat. Res., vol. 3, no. 36, pp. 65–77, Mar. 2019.

[10] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, ‘‘Stock price prediction using the ARIMA model,'' in Proc. UKSim-AMSS 16th Int. Conf. Comput. Model. Simul., Mar. 2014,pp. 106–112.

[11] W. Jiang, ‘‘Using the GARCH model to analyse and predict the different stock markets,’’ M.S. thesis, Uppsala Univ., Uppsala, Sweden, Dec. 2012, pp. 2–24.

[12] Y. Funde and A. Damani, ‘‘Comparison of ARIMA and exponential smoothing models in prediction of stock prices,’’ J. Predict. Markets, vol. 17, no. 1, pp. 21–38, Jul. 2023.

[13] B. B. Nair, V. P. Mohandas, and N. R. Sakthivel, ‘‘A decision tree-rough set hybrid system for stock market trend prediction,’’ Int. J. Comput. Appl., vol. 6, no. 9, pp. 1–6, Sep. 2010.

[14] Y.Yang, X. Hu, and H. Jiang, ‘‘Group penalized logistic regressions predict up and down trends for stock prices,’’ North Amer. J. Econ. Finance, vol. 59, Jan. 2022, Art. no. 101564.

[15] E. Hoseinzade and S. Haratizadeh, ‘‘CNNpred: CNN-based stock market prediction using a diverse set of variables,’’ Exp. Syst. Appl., vol. 129, pp. 273– 285, Sep. 2019.

[16] M. Lu and X. Xu, ‘‘TRNN: An efficient time-series recurrent neural network for stock price prediction,’’ Inf. Sci., vol. 657, Feb. 2024, Art. no. 119951.

[17] S. Sang and L. Li, "A novel variant of LSTM stock prediction method incorporating attention mechanism," Mathematics, vol. 12, no. 7, p. 945, Mar. 2024.

[18] Q. Jiang, C. Tang, C. Chen, X. Wang, and Q. Huang, "Stock price forecast based on LSTM neural network," in Proc. 12th Int. Conf. Manage. Sci. Eng. Manag. Melbourne, VIC, Australia: Springer, 2019, pp. 393–408. VOLUME 12, 2024 95221 Y. Sun et al.: Novel integrated approach for stock prediction [19] D. O. Oyewole, S. A. Akinwunmi, and T. O. Omotehinwa, "Deep LSTM and LSTM-attention Q-learning based reinforcement learning in oil and and gas sector prediction,’’ Know.-Based Syst., vol. 284, Jan. 2024, Art. no. 111290.

[19] I. Botanic, J. Bosna, and M. Mate tic, ‘‘Optimization of traditional stock market strategies using the LSTM hybrid approach,’’ Information, vol. 15, no. 3, p. 136, Feb. 2024.

[20] J. P. Behura, S. D. Pande, and J. V. N. Ramesh, ‘‘Stock price prediction using multi- layered sequential LSTM,’’ EAI Endorsed Trans. Scalable Inf. Syst., vol. 11, no. 4, pp. 1–8, 2024.

[21] M. Santoso, R. Sutjiadi, and R. Lim, ‘‘Indonesian stock prediction using support vector machine (SVM),’’ in Proc. MATEC Web Conf., vol. 164. Surabaya, Indonesia: Petra Christian Univ., 2018, pp. 1–5.

[22] E. E. Peters, Fractal Market Analysis: Applying Chaos Theory to Investment and Economics, vol. 24. Hoboken, NJ, USA: Wiley, 1994.

[23] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, ‘‘Financial time series forecasting using independent component analysis and support vector regression,’’ Decis. Support Syst., vol. 47, no. 2, pp. 115–125, May 2009.

[24] Y. Lin, Y. Yan, J. Xu, Y. Liao, and F. Ma, ‘‘Forecasting stock index price using the CEEMDAN-LSTM model,’’ North Amer. J. Econ. Finance, vol. 57, Jul. 2021, Art. no. 101421.

[25] Y. Huang, C. Deng, X. Zhang, and Y. Bao, ‘‘Forecasting of stock price index using support vector regression with multivariate empirical mode decomposition,’’ J. Syst. Inf. Technol., vol. 24, no. 2, pp. 75–95, Apr. 2022.

[26] M. Ali, D. M. Khan, H. M. Alshanbari, and A. A.-A.-H. El-Bagoury, ‘‘Prediction of complex stock market data using an improved hybrid EMDLSTM model,’’ Appl. Sci., vol. 13, no. 3, p. 1429, Jan. 2023.

[27] L. Kang, X. Li, L. Wu, Y. Li, and X. Zhao, ‘‘Predicting stock closing price with stock network public opinion based on AdaBoost-IWOA-Elman model and CEEMDAN algorithm,’’ Surge, vol. 1, p. 4, Jan. 2022.