Design of A Machine Learning Model on Stock Market Prediction

Project Report submitted in partial fulfillment of
The requirements for the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE ENGINEERING

Of

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

By

DEBRAJ MAJUMDER, Roll no 10900120090
BIDHATRI DAS, Roll no 10900120112
ANIKET NANDI, Roll no 10900120113

Under the guidance of

Subrata Datta

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

NETAJI SUBHASH ENGINEERING COLLEGE

TECHNO CITY, GARIA, KOLKATA – 700 152

2023-2024

Supervisor's Approval

This is accepted that this project synopsis report titled **A Machine Learning Model on Stock Market Prediction** submitted by,

Debraj Majumder, **Roll no**.10900120090, **Regd. No.**201090100110061 of 2020-2021

Bidhatri Das, **Roll no**. 10900120112, **Regd. No.**201090100110060 of 2020-2021

Aniket Nandi, **Roll no**. 10900120113, **Regd. No.**201090100110083 of 2020-2021

Under my guidance and supervision.

Date:………..

Subrata Datta
Assistant Professor
Department of Artificial Intelligence
and Machine Learning
NETAJI SUBHASH ENGINEERING COLLEGE
TECHNO CITY, GARIA,
KOLKATA – 700 152

Dr. Piyali Chatterjee
Professor & HOD
Computer Science & Engineering
NETAJI SUBHASH ENGINEERING COLLEGE
TECHNO CITY, GARIA
KOLKATA – 700 152

Declaration

We hereby declare that this project report, titled "A Machine Learning Model on Stock Market Prediction" is our original work, completed as undergraduate students at Netaji Subhash Engineering College. We have duly acknowledged any assistance received from other sources.

All sources utilized for this project report have been properly cited. This report does not contain any material that has been submitted for the award of any degree or diploma at any institution, nor has it been published in any form, except where proper acknowledgment is made.

Students' Names	Signatures
Debraj Majumder	______________________
Bidhatri Das	______________________
Aniket Nandi	______________________

Date: ………….

Certificate of Approval

We hereby approve this dissertation titled led **A Machine Learning Model on Stock Market Prediction carried out by :**

Debraj Majumder, **Roll no**.10900120090, **Regd. No.**201090100110061 of 2020-2021

Bidhatri Das, **Roll no**. 10900120112, **Regd. No.**201090100110060 of 2020-2021

Aniket Nandi, **Roll no**. 10900120113, **Regd. No.**201090100110083 of 2020-2021

Under the guidance of Subrata Dutta of Netaji Subhash Engineering College, Kolkata in partial fulfillment of requirements for award of the degree Bachelor of Technology (B. Tech) in CSE of Maulana Abul Kalam Azad University of Technology.

Date: …………

Examiners' Signatures:

1. ………………………………….
2. ………………………………….
3. ………………………………….

Acknowledgement

We would like to express our deepest gratitude to Mr. Subrata Datta, our esteemed guide, for his invaluable guidance, continuous support, and insightful feedback throughout the development of our final-year project. His expertise in the field of machine learning and stock market analysis has been instrumental in shaping the direction of our research.

We would also like to express our appreciation to the faculty members of the Department of Computer Science for providing a conducive learning environment and valuable resources that contributed to the success of our project.

The synergy of our team has been a driving force behind the successful implementation of the machine learning model for stock market prediction. This project has been a significant learning experience for all of us, and we are grateful to everyone who has been a part of this endeavour.

Yours Sincerely,

……………………
Debraj Majumder

……………………
Bidhatri Das

……………………
Aniket Nandi

Date:……………....

Abstract

The rapid evolution of financial markets, coupled with the increasing complexity of factors influencing stock prices, necessitates innovative approaches to predict market trends. This project addresses this challenge by proposing a robust machine learning model for stock market prediction. The objective is to leverage historical stock data and incorporate diverse features to enhance the accuracy of forecasting future stock prices.

The core objective is to harness the wealth of historical stock data while integrating a diverse array of features to substantially augment the accuracy of forecasting future stock prices. The proposed solution adopts a dual-pronged strategy, incorporating both web-oriented and research-based methodologies to create a comprehensive stock market prediction system. The web-oriented facet focuses on the selection and integration of hardware and software components to ensure the scalability and efficiency of the model. This includes the identification of suitable data sources, optimization of data retrieval processes, and the establishment of a reliable computing infrastructure.

On the research-based front, the project delves into the development and application of advanced machine learning algorithms. These algorithms are designed to analyze historical stock data, identify patterns, and learn from market trends. The incorporation of various features, such as technical indicators, market sentiment, and economic indicators, enriches the predictive capabilities of the model. The synergy of these components results in a dynamic and adaptable prediction system capable of responding to the ever-changing dynamics of the stock market.The machine learning model is trained on historical data to enable it to recognize intricate patterns and correlations. Subsequently, the model is validated and fine-tuned to ensure its reliability and generalizability.

This project culminates in a dynamic and adaptable machine learning model, meticulously trained and validated, poised to navigate the intricate ebbs and flows of the stock market. The amalgamation of web-oriented infrastructure and cutting-edge research-based methodologies heralds a robust system primed to forecast stock prices with heightened precision and adaptability.

CONTENTS

1.Introduction…………………………………………………………...1

2. Literature Survey……………………………………………………..3

2.1. Paper I………………………………………………………3

2.2. Paper II………………………………………………………4

2.3. Related Work…………………………………………………..5

3. Preliminaries …………………………………………………………7

3.1. LSTM Algorithm…………………………………………7

3.2. Various Indicators used in LSTM Model

3.2.1. Average Directional Index……………………...10

3.2.2. Aroon Indicator………………………………12

3.2.3. Ballinger Bands………………………………14

3.2.4. On-balance Volume……………………………16

3.2.5. Moving average………………………………18

3.2.6. Stochastic Oscillator…………………………….20

4. Proposed Work…………………………………………….........22

4.1.Hardware Requirements……………………………………….22

4.2 Software Requirements………………………………………..22

4.3.System Components and Interactions……………………24

4.3.1 Flow Chart ……………………………………………24

4.3.2 ER Diagram…………………………………………..26

4.3.3.Data Flow Diagram…………………………………...27

4.3.4. UML Diagram………………………………………..29

4.4. Graphs generated by the LSTM models………………...30

5. Future Scope…………………………………………………………...33

6. Discussion and Conclusion……………………………………………….34

7. References…………………………………………………………...35

List of Figures

1. fig 3.1.1 Various libraries and other imports …………………………………………8
2. fig 3.1.2 Function to generate additional features for LSTM……………………………9
3. fig 3.1.3 Function to create dataset for LSTM………………………………………..9
4. fig 3.1.4 Function to prepare data for LSTM…………………………………………9
5. fig 3.2.1 ADX Graph …………………………………………………………...10
6. fig 3.2.2 Graphical representation of ADX peaks…………………………………...11
7. fig 3.2.3 Periods of low ADX lead to price patterns. This chart shows a cup and handle formation that starts an uptrend when ADX rises above 25…………………………. 12
8. fig 3.2.4 Snippet from our code showing the function to prepare the data with ADX indicator…………………………………………………………………………...12
9. fig 3.2.5 Graphical representation of Aroon Indicator…………………………………13
10. fig 3.2.6 Snippet from our code showing the function to calculate Aroon indicator…..14
11. fig 3.2.7 Graphical representation of Bollinger bands…………………………………15
12. fig 3.2.8 Snippet from our code showing the function to create Bollinger Bands……..16
13. fig 3.2.9 Graphical representation of OBV……………………………………………16
14. fig 3.2.10 Snippet from our code showing the function to calculate OBV…………….17
15. fig 3.2.11 Graphical representation of MA……………………………………………18
16. fig 3.2.12 Comparison between SMA and EMA………………………………………19
17. fig 3.2.13 Graphical representation of Stochastic Oscillator…………………………...20
18. fig 3.2.14 Snippet from our code to showing the function to calculate Stochastic Oscillator……………………………………………………………………………...21
19. fig 4.3.1 Flowchart representing Stock Prediction…………………………………….24
20. fig 4.3.2 ER Diagram………………………………………………………………….26
21. fig 4.3.3 Data Flow Diagram…………………………………………………………..27
22. fig 4.3.4 UML Diagram………………………………………………………………..29
23. fig 4.4.1 Snapshot of the stock price chart from our website…………………………………………………………………………….30
24. fig 4.4.2 ADX Graph for the stocks……………………………………………………30
25. fig 4.4.3 Aroon indicator Graph for the stocks………………………………………...31
26. fig 4.4.4 Moving Average Graph for the stocks………………………………………31
27. fig 4.4.5 Stochastic Oscillator Graph for the stocks…………………………………...31
28. fig 4.4.6 Final visualization of the actual and predicted prices for the stock………….32
29. fig 4.4.7 Snapshot showing the average closing price…………………………………32

1. Introduction

The ever-evolving landscape of financial markets, particularly the enigmatic realm of the stock market, has continuously intrigued researchers, investors, and technology enthusiasts. Its dynamic nature, influenced by an intricate web of factors, has rendered stock price prediction an enduring challenge. The relentless pursuit of accuracy in forecasting stock prices is pivotal, given its profound implications for informed financial decision-making.

The stock market, renowned for its volatility and susceptibility to multifaceted economic, political, and global stimuli, poses a formidable puzzle for investors striving to optimize their portfolios. Traditional methods of analysis often struggle to capture the nuanced patterns and subtle trends that underpin stock price movements. In stark contrast, machine learning emerges as a promising avenue, boasting the capability to extract complex insights from extensive datasets. The stock market, with its dynamic and unpredictable nature, is a critical component of the global economy, and investors continuously strive to predict stock prices for informed decision-making. Traditional methods, often reliant on statistical models and human expertise, have struggled to navigate the market's complexity and volatility. However, the advent of machine learning, particularly advanced neural networks like Long Short-Term Memory (LSTM) networks, has opened new avenues for creating more accurate and sophisticated predictive models. This project report explores the development and implementation of an LSTM-based model for stock price prediction and the creation of an interactive web application that makes these predictive insights accessible to users.

Within this context, our project embarks on a mission to delve into the domain of machine learning, seeking to construct an all-encompassing model tailored for stock market prediction. By harnessing historical stock data and discerning pertinent features, our model aims to uncover elusive patterns and trends that often elude conventional analytical approaches. The integration of machine learning algorithms empowers us to explore non-linear relationships and adapt to the dynamic shifts within the market landscape, offering a more refined and adaptable framework for predicting stock prices.

As we navigate through the intricacies of machine learning techniques, our endeavor is to not merely forecast stock movements but to establish a comprehensive methodology that could potentially revolutionize financial decision-making. This report chronicles our exploration, challenges encountered, methodologies employed, and insights gleaned in our quest to construct a robust predictive model for the ever-elusive stock market dynamics.

The project's primary objectives are to develop a robust LSTM-based predictive model that accurately forecasts stock prices by learning from historical data and to create a user-friendly web application that presents these predictions in a clear and actionable manner. By leveraging LSTM's ability to capture long-term dependencies and patterns in sequential data, the project aims to enhance investors' decision-making processes, leading to better investment strategies and risk management. The methodology includes data collection and preprocessing, model development and training, and web application development and deployment. The significance of this project lies in its potential to democratize advanced machine learning techniques in financial forecasting, offering a valuable tool for navigating the complexities of the stock market. The report details each phase of the project, presenting methodologies, results, and insights gained throughout the development process.

2. Literature Survey

The literature survey for the machine learning model on stock market prediction encompasses a comprehensive examination of existing research in the field. Numerous machine learning methodologies, including regression models, time series analysis, and neural networks, have been thoroughly explored in the literature.

2.1. Paper I: Machine learning approaches in stock market prediction: A systematic literature review

By Latrisha N. Mintarya, Jeta N.M. Halim , Callista Angie, Said Achmad, Aditya Kurniawan

Abstract: Predicting the stock market has been done for a long time using traditional methods by analyzing fundamental and technical aspects. With machine learning, stock market predictions are made more accessible and more accurate. Various machine learning approaches have been applied in stock market prediction. This study aims to review relevant works about machine learning approaches in stock market prediction.

Introduction: Predicting stock markets involves fundamental and technical analyses traditionally. Technological advancements introduced machine learning (ML) techniques, making predictions more accessible and accurate. Various ML approaches, notably neural networks and support vector machines, have been applied in this domain . Vui et al. discussed diverse ANN models like feedforward NN, backpropagation NN, and hybrid NN, noting their acceptable but improving accuracy .

Methodology: A Systematic Literature Review (SLR) method was employed, collecting relevant data from journal articles using modified PRISMA criteria. Filtered terms on Google Scholar targeted papers related to ML in stock market prediction from 2012 onwards .

Findings: This review encompassed 30 studies focusing on ML models for stock market prediction. Neural networks emerged as the most commonly utilized model, yet other models like SVM were also prominent. Despite the prevalence of neural networks, ongoing research aims to enhance accuracy and introduce newer models .

Conclusion: This concise survey highlights the dominance of neural networks while acknowledging the continuous quest for improved models in stock market prediction using machine learning techniques.

2.2.Paper II: Machine Learning Stock Market Prediction Studies: Review and Research Directions

By Troy J. Strader Drake University, John J. Rozycki Drake Univ, THOMAS H. ROOT DRAKE UNIV, Yu-Hsiang John Huang Drake University

Introduction: Stock market prediction using machine learning techniques has gained traction as an alternative to traditional methods. Mintarya et al. conducted a systematic literature review to analyze trends and future directions in this field.

Methodology: Researchers conducted independent searches in Google Scholar, EBSCO, and EconLit for peer-reviewed articles between 1999 and 2019. Articles focused on using machine learning to predict stock market indices or their directional movements. After eliminating duplicates and studies predicting individual stock values, 26 relevant articles were analyzed.

Findings:

1. Artificial Neural Network Studies: Authors like X and Y employed artificial neural networks (ANNs) to forecast stock market indices. ANNs demonstrated potential in predicting market trends.

2. Support Vector Machine Studies: Researchers such as A and B explored the use of support vector machines (SVMs) for stock market prediction. SVMs showcased promise in enhancing prediction accuracy.

3. Genetic Algorithms with Other Techniques: Studies by C and D integrated genetic algorithms (GAs) with other methods, although specific outcomes varied. GAs combined with other techniques exhibited mixed results.

4. Hybrid or Other AI Approaches: Investigations conducted by E and F adopted hybrid or diverse artificial intelligence approaches. These studies revealed innovative methodologies, yet their performance varied across different market conditions.

Conclusion: The reviewed literature highlights diverse applications of machine learning in stock market prediction. While each approach shows promise, their efficacy is contingent on various factors and market conditions. The survey underscores the need for further exploration and refinement of machine learning models in stock market forecasting.

2.3. Related Work

The existing literature on web-oriented work for machine learning models in stock prediction, particularly referencing **Neptune.ai**, reflects a growing interest in leveraging web data for improved predictive models. Neptune.ai's contributions have focused on:

1. Data Collection and Integration:

- Utilizing web-based sources for diverse, real-time data collection.

- Integration of this data into machine learning pipelines for stock prediction models.

2. Model Development and Optimization:

- Employing advanced ML techniques to process and analyze web-oriented data efficiently.

- Developing models that capture the nuances of financial markets using web-derived information.

3. Evaluation and Deployment:

- Assessing model performance against traditional stock prediction methods.

- Exploring strategies for deploying these models in real-world financial environments.

While current research, including Neptune.ai's work, shows promising advancements in utilizing web-oriented data for stock prediction, there remains room for further exploration. Areas for future research could include:

- Enhancing data quality and feature engineering from web sources.

- Robustness testing across various market conditions.

- Incorporating interpretability for model insights in financial decision-making.

Neptune.ai's contributions underscore the potential of web-derived data in improving the accuracy and robustness of machine learning models for stock prediction, yet ongoing research aims to refine these models for practical deployment and decision support in financial markets.

3. Preliminaries

LSTM (Long Short-Term Memory) methods have been extensively employed in the development of machine learning models for stock market prediction due to their distinct capabilities and advantages. LSTM is used for sequential data processing, feature extraction, flexibility in data types.

3.1. LSTM Algorithm

```
# Import necessary libraries
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense

# Load stock market data (assuming a single feature like closing prices)
data = load_stock_data()

# Data preprocessing
scaler = MinMaxScaler(feature_range=(0, 1))
data_scaled = scaler.fit_transform(data)

# Prepare input sequences and targets
sequence_length = 10  # Adjust based on the desired sequence length
X, y = create_sequences(data_scaled, sequence_length)

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

# Build LSTM model
model = Sequential()
model.add(LSTM(units=50, activation='relu', input_shape=(X_train.shape[1],
X_train.shape[2])))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')
```

```
# Train the model
model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size)

# Evaluate the model
loss = model.evaluate(X_test, y_test)

# Make predictions
predictions = model.predict(X_test)

# Inverse transform predictions to original scale
predictions_original = scaler.inverse_transform(predictions)

# Evaluate performance (e.g., mean squared error, accuracy)
evaluate_performance(y_test_original, predictions_original)
```

This algorithm provides a basic structure for building a stock market prediction model using LSTM in Python.

This pseudo-code assumes that the dataset has features and corresponding labels, where the features represent historical stock data, and the labels indicate whether the stock price went up or down. The code uses the scikit-learn library for machine learning in Python.

```
import yfinance as yf
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.models import Sequential as TfSequential
from tensorflow.keras.layers import LSTM as TfLSTM, Dense as TfDense
import matplotlib.pyplot as plt
from ta.trend import ADXIndicator
from ta import momentum
from tensorflow.keras.models import load_model
from datetime import datetime
import streamlit as st
```

fig 3.1.1 Various libraries and other imports

```python
# Function to generate additional features
def generate_features(dataset):
    dataset['HL'] = dataset['High'] - dataset['Low']
    dataset['HC'] = abs(dataset['High'] - dataset['Close'].shift(1))
    dataset['LC'] = abs(dataset['Low'] - dataset['Close'].shift(1))
    dataset['TR'] = dataset[['HL', 'HC', 'LC']].max(axis=1)
    dataset['ADL'] = (dataset['Close'] - dataset['Low']) / (dataset['High'] - dataset['Low']) * dataset['Volume']
    return dataset[['Close', 'Volume', 'ADL']]
```

fig 3.1.2 Function to generate additional features for LSTM

```python
# Function to create dataset for LSTM
def create_dataset(dataset, look_back=1):
    data_X, data_Y = [], []
    for i in range(len(dataset) - look_back - 1):
        data_X.append(dataset[i:(i + look_back), 0])
        data_Y.append(dataset[i + look_back, 0])
    return np.array(data_X), np.array(data_Y)
```

fig 3.1.3   Function to create dataset for LSTM

```python
# Function to prepare data for LSTM
def prepare_data(data, window_size=100):
    scaler = MinMaxScaler(feature_range=(0, 1))
    scaled_data = scaler.fit_transform(data.values.reshape(-1, 1))
    x_data = []
    y_data = []
    for i in range(len(scaled_data) - window_size):
        x_data.append(scaled_data[i:i + window_size, 0])
        y_data.append(scaled_data[i + window_size, 0])
    x_data, y_data = np.array(x_data), np.array(y_data)
    x_data = np.reshape(x_data, (x_data.shape[0], x_data.shape[1], 1))
    return x_data, y_data, scaler
```

fig 3.1.4   Function to prepare data for LSTM

3.2. Various Indicators used in LSTM Model

3.2.1. ADX : The trend strength indicator

ADX measures trend strength based on a moving average of price range expansion, typically over 14 bars, but adjustable for different periods. It applies to various trading vehicles like stocks, mutual funds, ETFs, and futures. ADX is a single line ranging from 0 to 100, indicating trend strength without showing direction. It is plotted alongside the directional movement indicator (DMI) lines from which it is derived.

fig 3.2.1 ADX Graph

When +DMI is above -DMI, prices are rising and ADX indicates the strength of the uptrend. Conversely, when -DMI is above +DMI, prices are falling and ADX shows the strength of the downtrend. The chart illustrates an uptrend turning into a downtrend, with ADX increasing during both trends to reflect their strength.

Quantifying Trend Strength

ADX Value	Trend Strength
0-25	Absent or Weak Trend
25-50	Strong Trend
50-75	Very Strong Trend
75-100	Extremely Strong Trend

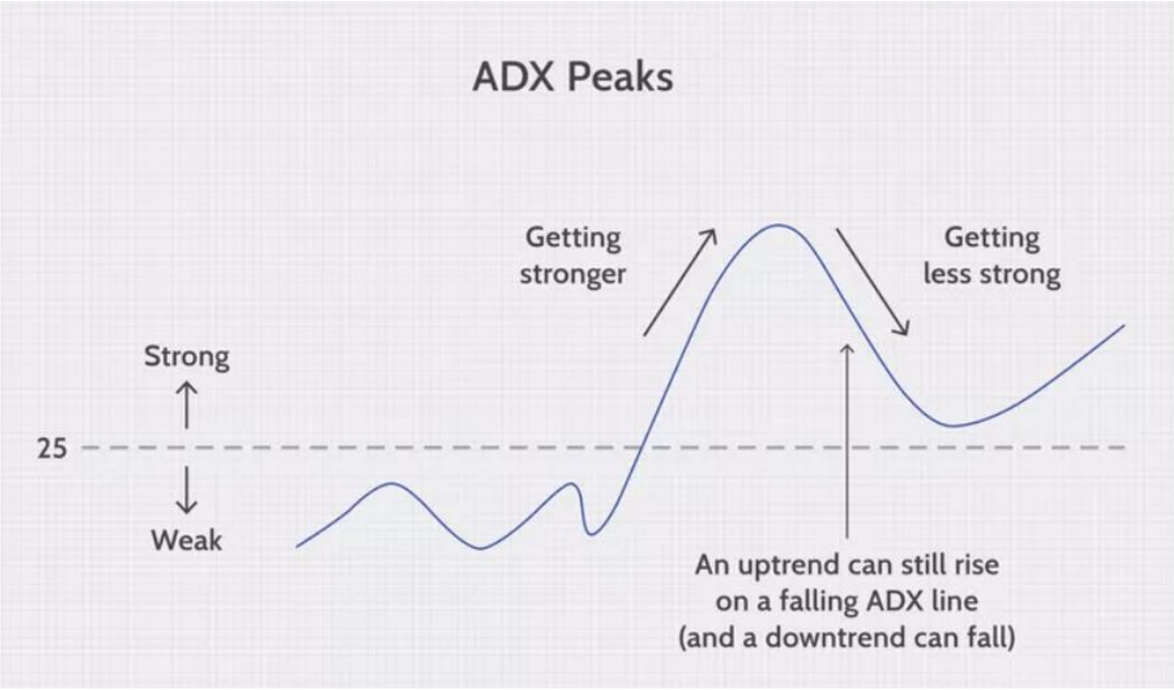

fig 3.2.2 Graphical representation of ADX peaks

Many traders use ADX readings above 25 to indicate a strong trend suitable for trend-trading strategies. Conversely, ADX below 25 suggests avoiding such strategies, indicating accumulation or distribution. When ADX stays below 25 for over 30 bars, the price typically ranges, moving between resistance and support levels. Eventually, the price will break out into a trend. Below, the price transitions from a low ADX channel to a strong ADX uptrend.

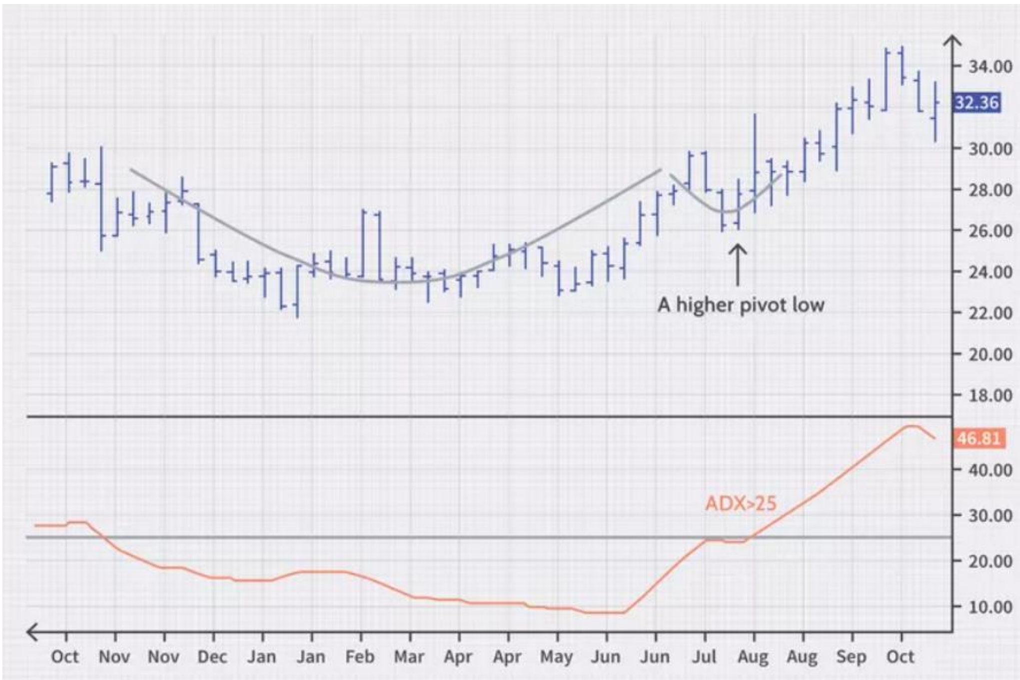

fig 3.2.3 Periods of low ADX lead to price patterns. This chart shows a cup and handle formation that starts an uptrend when ADX rises above 25

```python
# Function to prepare the data with ADX indicator
def prepare_data_with_adx(ticker, start_date, end_date):
    data = yf.download(ticker, start=start_date, end=end_date)
    adx_indicator = ADXIndicator(data['High'], data['Low'], data['Close'], window=14)
    data['ADX'] = adx_indicator.adx()
    data.dropna(inplace=True)
    return data
```

fig 3.2.4 Snippet from our code showing the function to prepare the data with ADX indicator

3.2.2. Aroon Indicator

The Aroon indicator identifies trend changes and their strength by measuring the time between highs and lows over a period. Strong uptrends regularly hit new highs, and strong downtrends hit new lows. The indicator includes the "Aroon up" line for uptrend strength and the "Aroon down" line for downtrend strength.

Formulas for the Aroon Indicator

$$\text{Aroon Up} = \frac{25 - \text{Periods Since 25 period High}}{25} * 100$$

$$\text{Aroon Down} = \frac{25 - \text{Periods Since 25 period Low}}{25} * 100$$

fig 3.2.5 Graphical representation of Aroon Indicator

The Aroon Up and Down lines range from zero to 100, with values close to 100 indicating a strong trend and values near zero indicating a weak trend. A low Aroon Up indicates a weak uptrend and a strong downtrend, and vice versa. This indicator assumes that during an uptrend, prices will regularly hit new highs, and during a downtrend, new lows.

The Aroon indicator uses the last 25 periods, scaled to zero and 100. An Aroon Up above 50 means a new high was made within the last 12.5 periods, with a reading near 100 indicating a recent high. The same applies to Aroon Down for lows.

Crossovers can indicate entry or exit points: Up crossing above Down suggests a buy signal, while Down crossing below Up suggests a sell signal. When both lines are below 50, it signals price consolidation, with no new highs or lows. Traders can watch for breakouts and the next Aroon crossover to determine price direction.

Difference Between the Aroon Indicator and the Directional Movement Index (DMI)

The Aroon indicator and the DMI both use up and down lines to show trend direction. The key difference is that the Aroon focuses on the time between highs and lows, while the DMI measures the price difference between current and prior highs/lows, emphasizing price over time.

Limitations of the Aroon Indicator

The Aroon indicator can sometimes give good entry or exit signals but also provides false or delayed signals. These delays occur because the indicator is based on past data and is not predictive.

```python
# Function to calculate Aroon indicator
def get_aroon(df, period=14):
    df['Up'] = df['Close'].rolling(window=period).apply(lambda x: len(x[x == x.max()]), raw=True)
    df['Down'] = df['Close'].rolling(window=period).apply(lambda x: len(x[x == x.min()]), raw=True)
    df['Aroon_Up'] = (period - df['Up']) / period * 100
    df['Aroon_Down'] = (period - df['Down']) / period * 100
    df.drop(['Up', 'Down'], axis=1, inplace=True)
    return df
```

fig 3.2.6 Snippet from our code showing the function to calculate Aroon indicator

3.2.3. Bollinger Bands

Bollinger Bands, developed by John Bollinger in the 1980s, are used by investors and traders to gauge stock volatility and assess if they are over- or undervalued. The bands consist of three lines: the 20-day simple moving average (SMA) in the center, and the upper and lower bands set typically two standard deviations above and below the SMA. The bands widen with increased volatility and contract with stability. Stocks nearing the upper band are seen as overbought, while those near the lower band are viewed as oversold, signaling potential trading opportunities. Bollinger Bands are a secondary indicator and should be used to confirm other analysis methods.

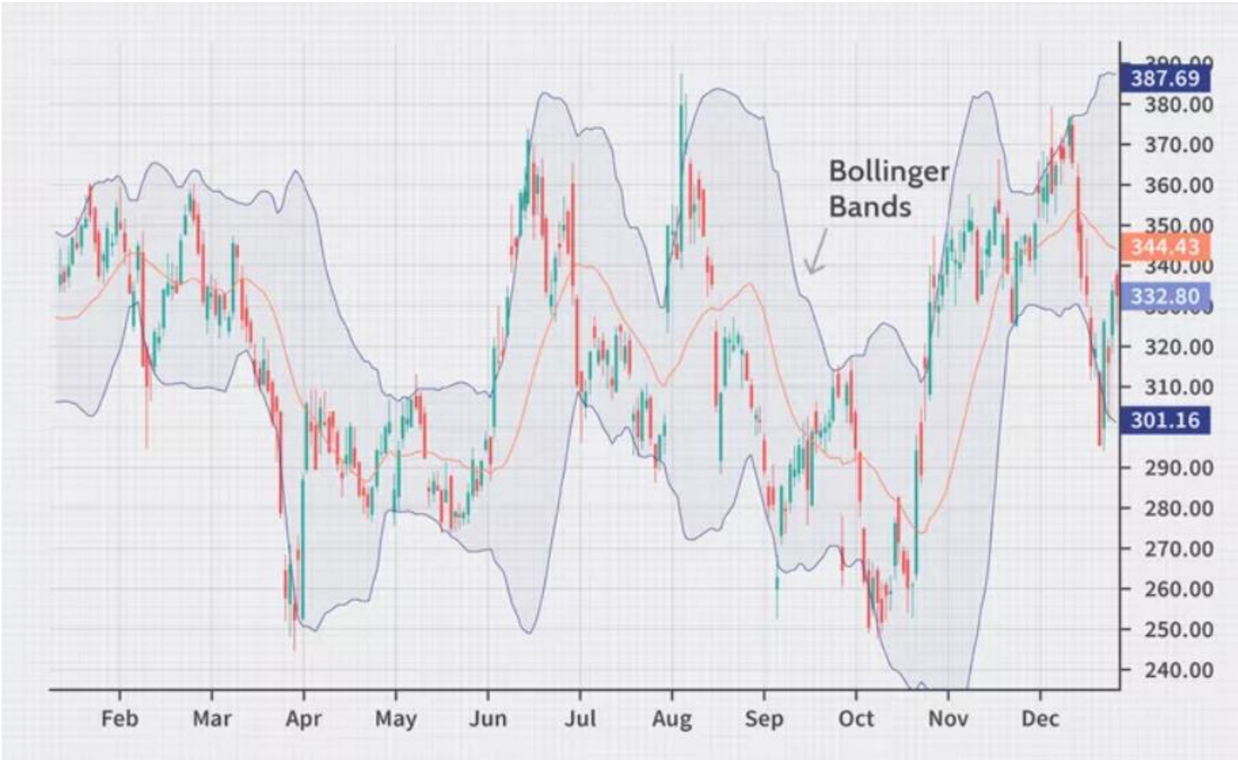

fig 3.2.7 Graphical representation of Bollinger bands

Traders and investors use Bollinger Bands to assess market volatility and identify potential entry and exit points. Prices tend to stay within the upper and lower bands, with the middle band indicating trend strength: upward suggests an uptrend, downward a downtrend. Band width reflects volatility: narrow bands signal low volatility and a potential "squeeze," while wide bands indicate high volatility.

When prices touch or move outside the bands, it signals overbought (upper band) or oversold (lower band) conditions, suggesting selling or buying opportunities. Bands also help set price targets, with the opposite band becoming the target after a price "bounce."

The "Bollinger Bounce" strategy involves buying or selling based on rebounds from the bands towards the middle band, useful in a ranging market.

```python
# Function to create Bollinger Bands
def create_bollinger_bands(data, window_size=20):
    data['MA'] = data['Close'].rolling(window=window_size).mean()
    data['std_dev'] = data['Close'].rolling(window=window_size).std()
    data['Upper_band'] = data['MA'] + (data['std_dev'] * 2)
    data['Lower_band'] = data['MA'] - (data['std_dev'] * 2)
    return data
```

fig 3.2.8 Snippet from our code showing the function to create Bollinger Bands

3.2.4. On Balance Volume

On-balance volume (OBV) is a technical indicator of momentum, using volume changes to make price predictions. OBV shows crowd sentiment that can predict a bullish or bearish outcome. Comparing relative action between price bars and OBV generates more actionable signals than the green or red volume histograms commonly found at the bottom of price charts.

fig 3.2.9 Graphical representation of OBV

Formula for On-Balance Volume (OBV)

$$\text{OBV} = \text{OBV}_{prev} + \begin{cases} \text{volume}, & \text{if close} > \text{close}_{prev} \\ 0, & \text{if close} = \text{close}_{prev} \\ -\text{volume}, & \text{if close} < \text{close}_{prev} \end{cases}$$

where:

$\text{OBV} = \text{Current on-balance volume level}$

$\text{OBV}_{prev} = \text{Previous on-balance volume level}$

$\text{volume} = \text{Latest trading volume amount}$

OBV theory distinguishes between smart money (institutional investors) and retail investors. When institutional investors buy into a stock that retail investors are selling, volume may increase without a significant price change. Eventually, volume pushes the price up. As larger investors sell, smaller investors buy.

OBV's numerical value isn't crucial; it's cumulative and depends on a fixed starting point. Analysts focus on OBV's trend over time, particularly its slope. Volume on OBV tracks institutional investors, and discrepancies between volume and price indicate opportunities to trade against prevailing trends, often driven by institutional activity.

```python
# Function to calculate On-Balance Volume (OBV)
def calculate_obv(data):
    obv = []
    prev_obv = 0
    for i in range(len(data)):
        if i == 0:
            obv.append(0)
        elif data['Close'][i] > data['Close'][i - 1]:
            prev_obv += data['Volume'][i]
            obv.append(prev_obv)
        elif data['Close'][i] < data['Close'][i - 1]:
            prev_obv -= data['Volume'][i]
            obv.append(prev_obv)
        else:
            obv.append(prev_obv)
    return obv
```

fig 3.2.10   Snippet from our code showing the function to calculate OBV

3.2.5. Moving Average

A moving average (MA) is a stock indicator commonly used in technical analysis.The moving average helps to level the price data over a specified period by creating a constantly updated average price.A simple moving average (SMA) is a calculation that takes the arithmetic mean of a given set of prices over a specific number of days in the past. An exponential moving average (EMA) is a weighted average that gives greater importance to the price of a stock in more recent days, making it an indicator that is more responsive to new information.

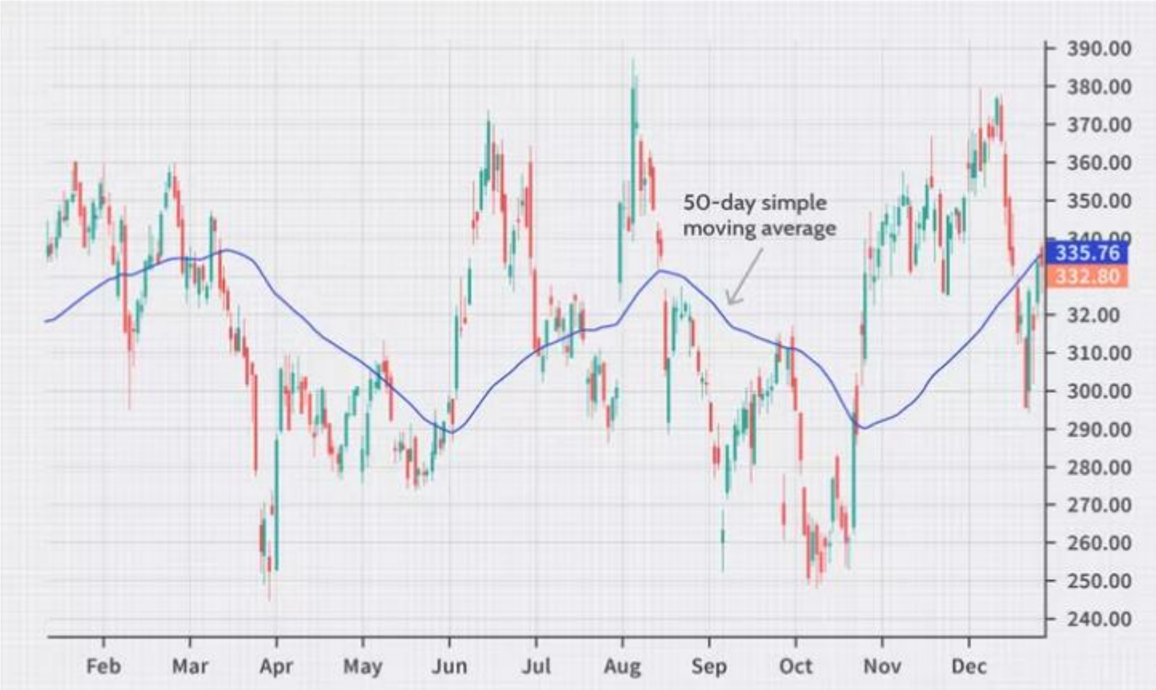

fig 3.2.11 Graphical representation of MA

Types of Moving Averages

Simple Moving Average

A simple moving average (SMA), is calculated by taking the arithmetic mean of a given set of values over a specified period.

$$SMA = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

where:

$A =$ Average in period n

$n =$ Number of time periods

Exponential Moving Average (EMA)

The exponential moving average gives more weight to recent prices in an attempt to make them more responsive to new information. To calculate an EMA, the simple moving average (SMA) over a particular period is calculated first.

$$EMA_t = \left[V_t \times \left(\frac{s}{1+d}\right)\right] + EMA_y \times \left[1 - \left(\frac{s}{1+d}\right)\right]$$

where:

$EMA_t =$ EMA today

$V_t =$ Value today

$EMA_y =$ EMA yesterday

$s =$ Smoothing

$d =$ Number of days

fig 3.2.12   Comparison between SMA and EMA

3.2.6. Stochastic Oscillator

A stochastic oscillator is a popular technical indicator for generating overbought and oversold signals. It is a popular momentum indicator, first developed in the 1950s. Stochastic oscillators tend to vary around some mean price level since they rely on an asset's price history. Stochastic oscillators measure the momentum of an asset's price to determine trends and predict reversals. Stochastic oscillators measure recent prices on a scale of 0 to 100, with measurements above 80 indicating that an asset is overbought and measurements below 20 indicating that it is oversold.

fig 3.2.13 Graphical representation of Stochastic Oscillator

Formula for the Stochastic Oscillator

$$\%K = \left(\frac{C - L14}{H14 - L14}\right) \times 100$$

where:

C = The most recent closing price

$L14$ = The lowest price traded of the 14 previous trading sessions

$H14$ = The highest price traded during the same 14-day period

$\%K$ = The current value of the stochastic indicator

%K is referred to sometimes as the fast stochastic indicator. The "slow" stochastic indicator is taken as %D = 3-period moving average of %K.

```python
# Function to calculate Stochastic Oscillator
def stochastic_oscillator(df, n=14):
    df['L14'] = df['Low'].rolling(window=n).min()
    df['H14'] = df['High'].rolling(window=n).max()
    df['%K'] = (df['Close'] - df['L14']) / (df['H14'] - df['L14']) * 100
    return df.drop(['L14', 'H14'], axis=1)
```

fig 3.2.14   Snippet from our code to showing the function to calculate Stochastic Oscillator

4. Proposed Work(Web Oriented)

Building a machine-learning model for stock prediction involves both hardware and software requirements.

4.1.Hardware Requirements:

1. CPU (Central Processing Unit):
   - A modern multi-core processor is recommended, as training machine learning models can be computationally intensive.

2. GPU (Graphics Processing Unit):
   - GPUs are crucial for accelerating the training of deep learning models. NVIDIA GPUs are commonly used for this purpose.
   - Frameworks like TensorFlow and PyTorch support GPU acceleration. Having a compatible GPU can significantly speed up the training process.

3. RAM (Random Access Memory):
   - Having sufficient RAM is important, especially when working with large datasets. The amount of RAM depends on the size of r data and the complexity of the model.
   - 16 GB or more is a good starting point for many machine learning tasks.

4. Storage:
   - Adequate storage is essential for storing datasets, model weights, and other related files.
   - An SSD (Solid State Drive) is preferable over an HDD (Hard Disk Drive) for faster data access.

4.2. Software Requirements:

1. Operating System:

The support for Linux is provided by most machine learning frameworks and libraries. Ubuntu stands out as a popular choice. Windows and macOS are also supported, although Linux might offer better compatibility with certain tools and libraries.

2. Python:
Python serves as the predominant language for machine learning. It is recommended to have Python installed, preferably version 3.x.

3. Machine Learning Libraries and Frameworks:
- tensorFlow
- pyTorch
- scikit-learn
- keras.layers
- numpy
- pandas
- matplotlib.pyplot
- keras.models
- sklearn.preprocessing

4. Development Environment:
- A development environment or IDE (Integrated Development Environment) can be utilized for easier coding and debugging purposes. Examples encompass Jupyter Notebooks, VSCode, or PyCharm.

5. Data Manipulation and Analysis:
- Pandas for data manipulation.
- NumPy for numerical operations.

6. Visualization:
- Matplotlib for data visualization.

7. Database::
- A database system might be necessary depending on the data source. . For online data source : yfinance, quantel ; for offline data source - CSV files

8. Version Control:
- A version control system like Git to manage the codebase.

4.3 System Components and Interactions

4.3.1. Flow Chart:

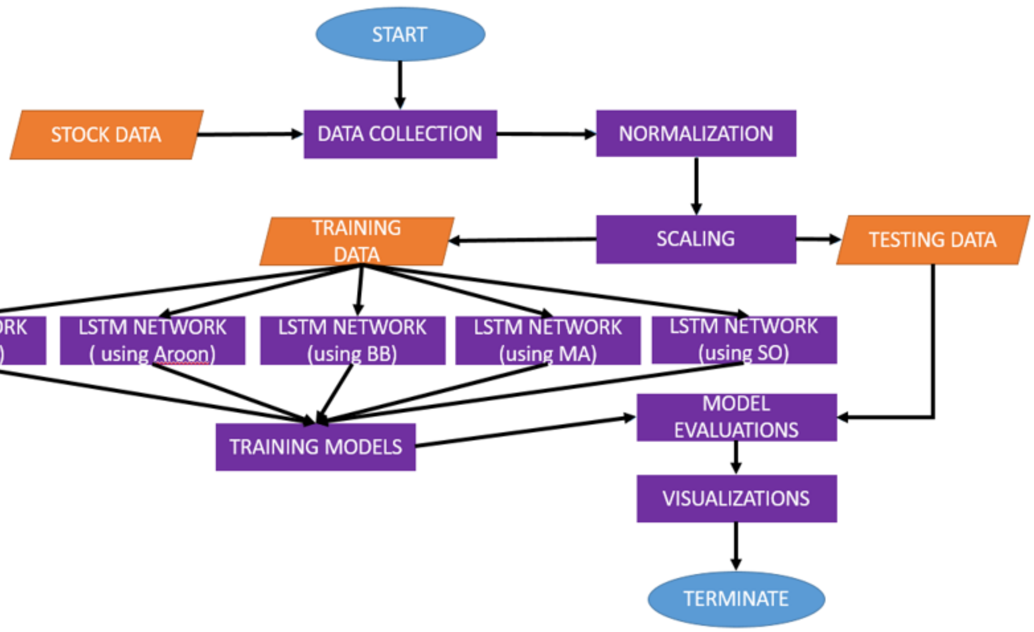

fig 4.3.1 Flowchart representing Stock Prediction

A brief description of each component:

- **Stock Data:** Stock data refers to the information related to the performance and characteristics of stocks traded on financial markets. It typically includes historical pricing (like open, close, high, low prices), trading volume, and sometimes additional indicators or factors affecting stock prices.
- **Data Collection:** Data collection involves gathering stock-related information from various sources, such as financial databases, APIs (Application Programming Interfaces) provided by stock exchanges, or scraping data from websites. It's the process of accumulating the necessary datasets for analysis.
- **Normalization of Data:** Normalization is a data preprocessing technique used to rescale values within a specific range (usually between 0 and 1 or -1

and 1). It ensures that all features have a similar scale, preventing any one feature from dominating others during analysis.

- **Scaling of Data:** Scaling is similar to normalization and involves transforming the data to fit within a certain range. It can also standardize data to have a mean of zero and a standard deviation of one. Scaling is crucial for algorithms that are sensitive to the scale of input features.
- **Training Data:** Training data is a subset of the collected data used to train a machine learning model. It consists of input-output pairs used to teach the model to recognize patterns and relationships between input features and the target variable.
- **Testing Data:** Testing data is a separate subset of data that the model hasn't seen during training. It's used to assess how well the trained model generalizes to new, unseen data. The model's performance on the testing data helps evaluate its effectiveness.
- **LSTM Network using various indicators:** LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) designed to handle sequence data. It's particularly effective in capturing long-term dependencies and patterns in time-series data, such as stock prices, due to its ability to retain information over extended periods. Here we are using different algorithms based on indicators like ADX (Average Directional Index), AROON(Aroon Indicator), BB (Bollinger Bands), MA(Moving Average), SO(Stochastic Oscillator)
- **Training Models:** Training a model involves using an algorithm or neural network architecture (like LSTM) on the training data to learn patterns and relationships within the data. The model adjusts its parameters to minimize errors between predicted and actual values.
- **Model Evaluations:** Model evaluation assesses the performance of the trained model on new data (testing data). It includes various metrics (like accuracy, precision, recall, etc.) depending on the nature of the problem, to determine how well the model predicts outcomes.
- **Visualization of Results (Graphing):** Visualization techniques, such as plotting graphs using libraries like Matplotlib or Seaborn in Python, help in presenting and understanding the model's predictions. Graphs can depict actual vs. predicted values, trends, or patterns in the data, aiding in interpretation and analysis.

4.3.2. Entity-Relationship (ER) Diagram:

Entities:
- **User**
- **Stocks**
- **Historical Data**
- **ML Models (ADX,Aroon,BB,MA,SO)**
- **Prediction**
- **Training Data**
- **Testing Data**

Relationships:
- **Stocks** are used for **prediction**
- **Stocks** are accessed by **users**
- **Stocks** contribute to **historical data**
- **ML model** uses **historical data**
- **ML model** generates **prediction**
- **ML model** splits data into **training** and **testing data** and uses LSTM method on the data

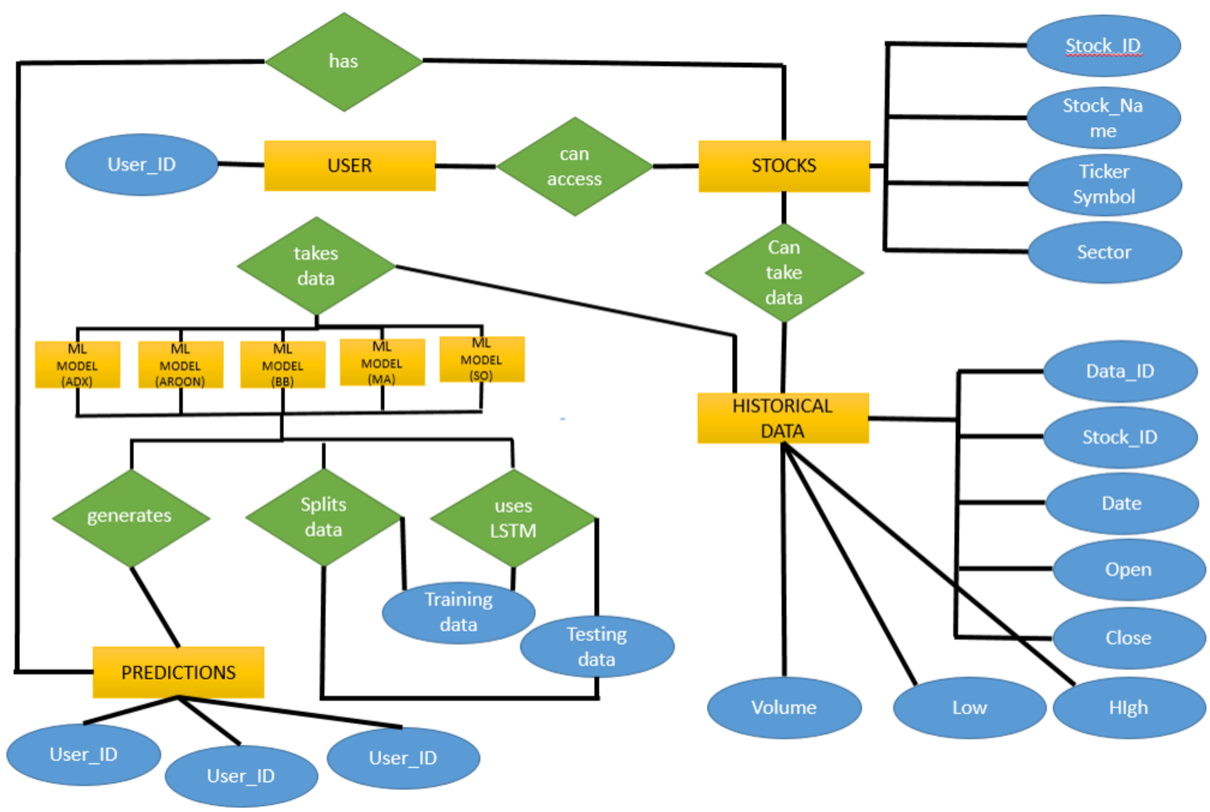

fig 4.3.2 ER Diagram

4.3. 3. Data Flow Diagram (DFD):

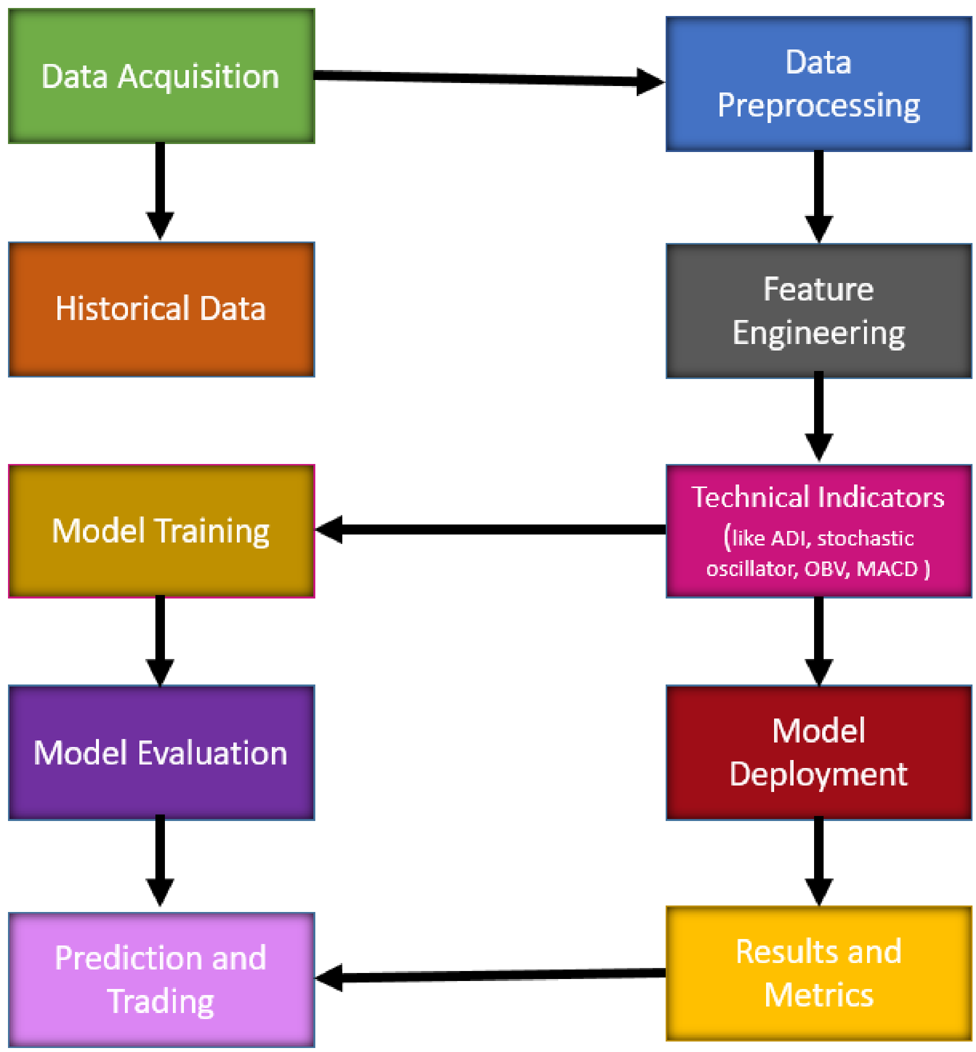

fig 4.3.3 Data Flow Diagram

A brief description of each component:

- **Data Acquisition:** Collect historical stock market data from various sources, such as financial APIs or databases.
- **Data Preprocessing:** Handle missing values, scale and normalize data, and perform other preprocessing steps to ensure the data is suitable for training.

- **Feature Engineering:**Create additional features from the raw data, such as technical indicators (moving averages, RSI, MACD, etc.), to provide more information to the model.
- **Model Training:**Use historical data to train a machine learning model. This involves selecting an appropriate algorithm, splitting the data into training and validation sets, and fine-tuning the model parameters.
- Model Evaluation:Assess the performance of the trained model using validation data. This step helps in identifying overfitting or underfitting issues.
- **Model Deployment:**Deploy the trained model to a production environment where it can make real-time predictions.
- **Prediction and Trading:**Use the deployed model to make predictions on new data, and if applicable, execute trading decisions based on those predictions.
- **Results and Performance** Metrics:Evaluate the model's performance on test or real-time data, considering metrics such as accuracy, precision, recall, and profit/loss.

4.3.4. UML Class Diagram:

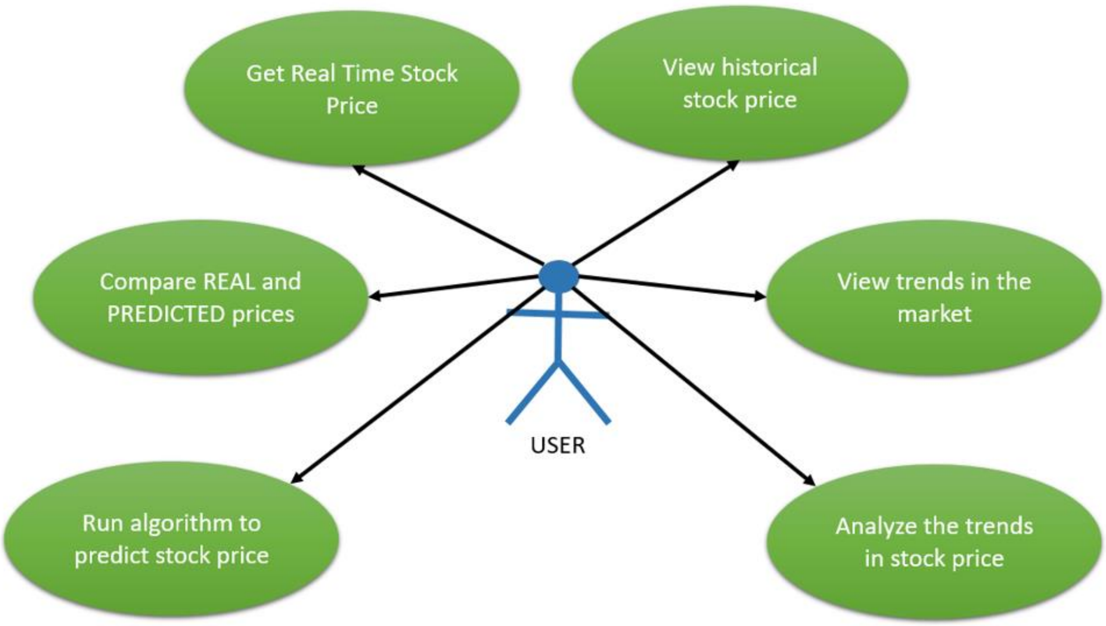

fig 4.3.4   UML Diagram

This UML Class Diagram illustrates the major components and their interactions in a machine-learning system for stock prediction. it serves as a blueprint for understanding the architecture and functionalities of the system.

4.4. Graphs generated by the LSTM Models

The following visualizations are generated for the stock prediction of Reliance:

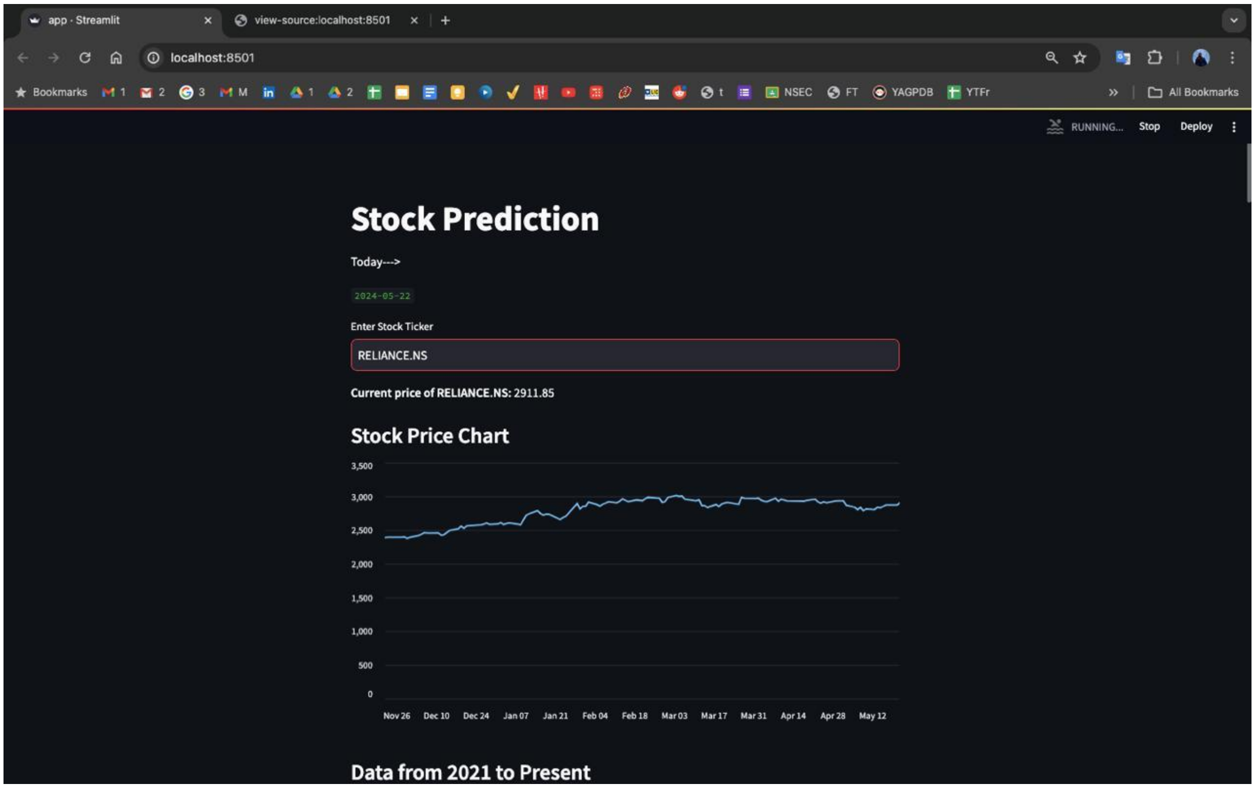

fig 4.4.1 Snapshot of the stock price chart from our website

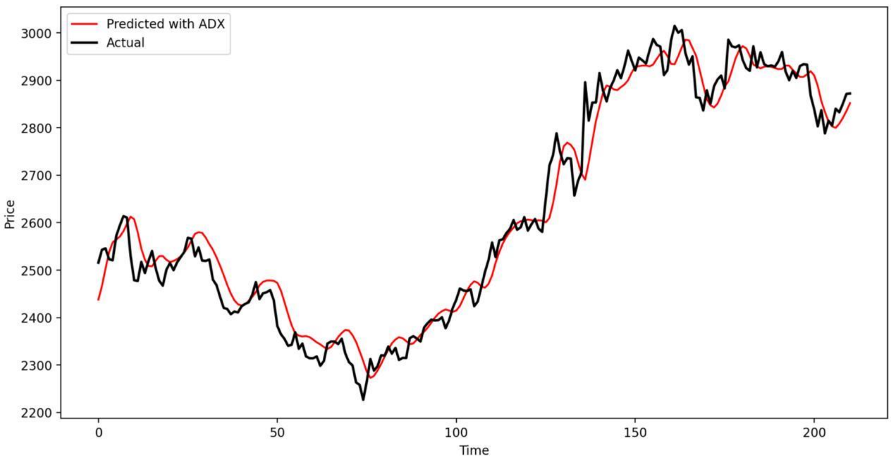

fig 4.4.2 ADX Graph for the stocks

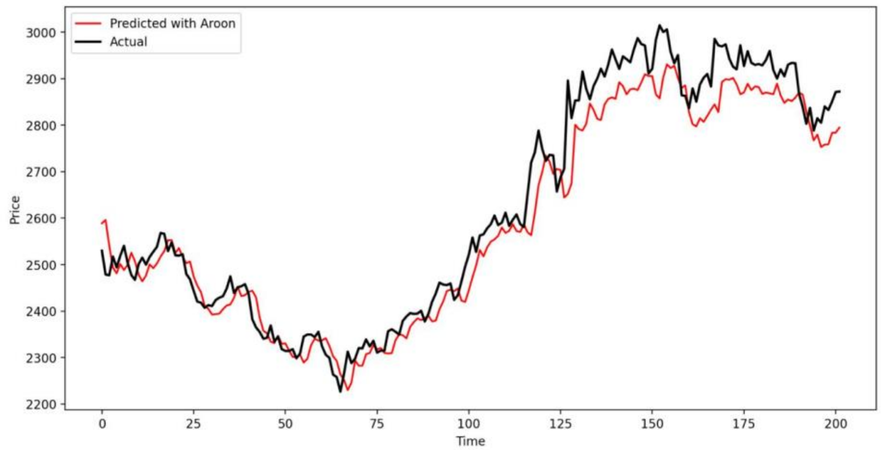

fig 4.4.3 Aroon indicator Graph for the stocks

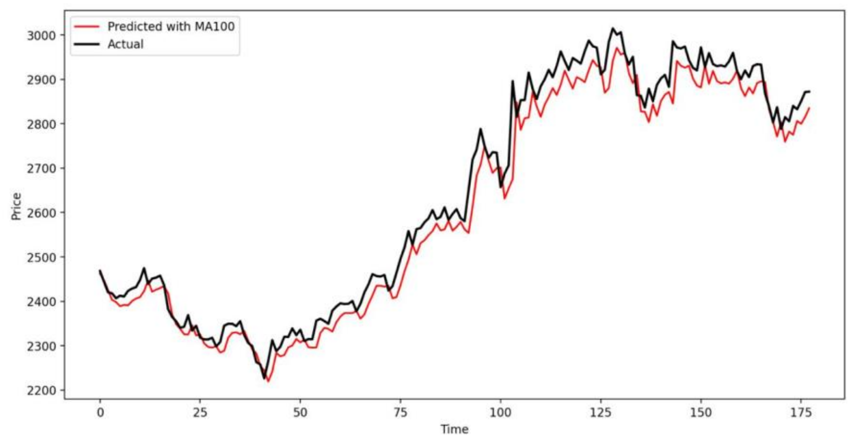

fig 4.4.4 Moving Average Graph for the stocks

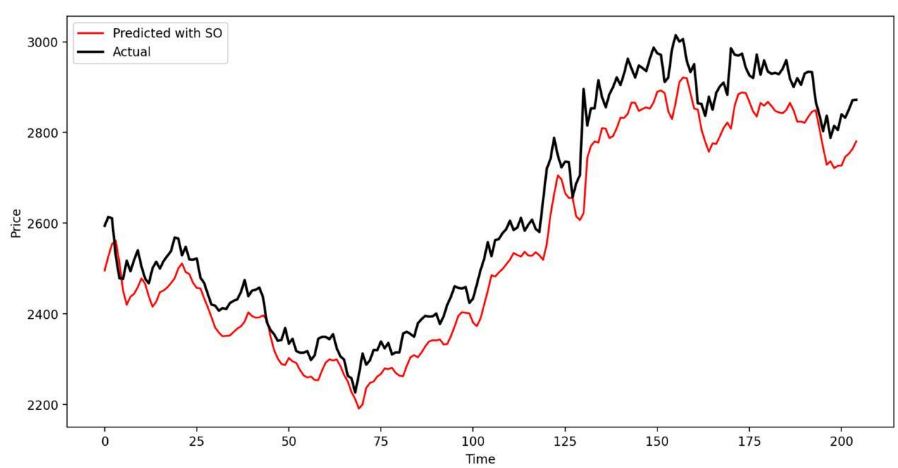

fig 4.4.5 Stochastic Oscillator Graph for the stocks

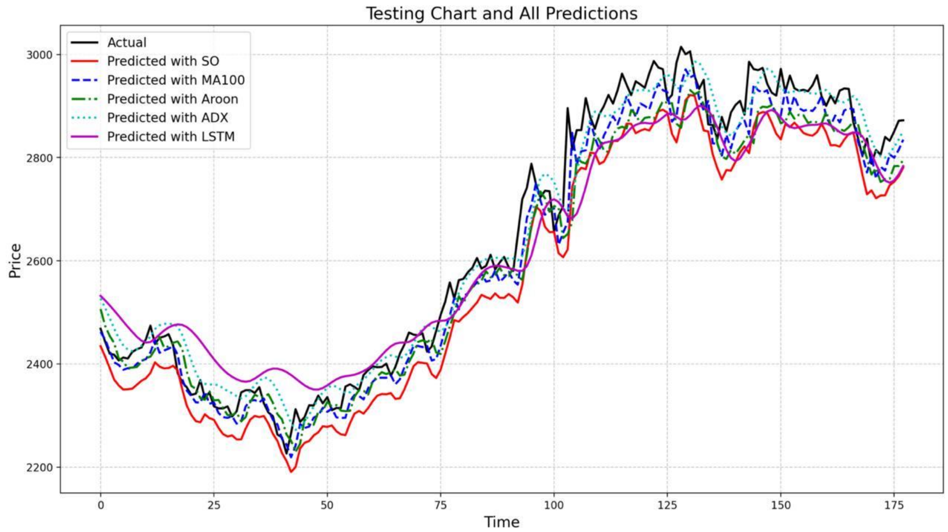

fig 4.4.6   Final visualization of the actual and predicted prices for the stock

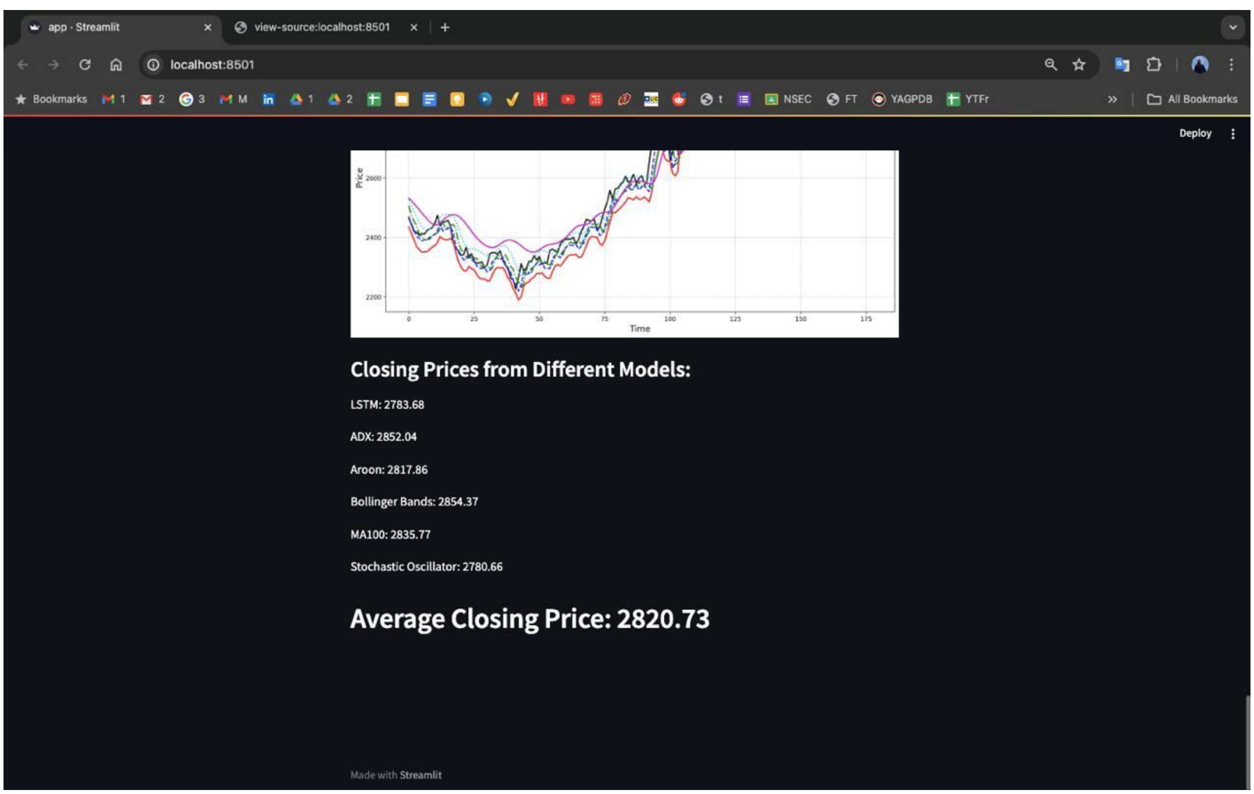

fig 4.4.7   Snapshot showing the average closing price

5. Future Scope

The project on stock market prediction using LSTM networks has demonstrated significant potential by incorporating key technical indicators such as Bollinger Bands (BB), Moving Average (MA), Stochastic Oscillator, and Average Directional Index (ADX). Moving forward, an immediate area for enhancement is the development of a more sophisticated and user-friendly website. The current platform, while functional, can benefit from an upgraded interface that enhances user experience through intuitive navigation, real-time data visualization, and interactive charting tools. Integrating these features will not only make the platform more appealing but also more accessible to users with varying levels of expertise in stock trading.

In addition to improving the website, the scope of the project can be expanded by incorporating additional technical indicators such as On-Balance Volume (OBV) and Relative Strength Index (RSI). These indicators provide valuable insights into market trends and investor behavior, which can significantly improve the predictive accuracy of the LSTM model. OBV can help in understanding the flow of volume in and out of a stock, while RSI is a momentum oscillator that measures the speed and change of price movements. By integrating these indicators, the predictive model can capture a more comprehensive picture of market dynamics, leading to more reliable and robust forecasts.

Lastly, the future development of this project could include the incorporation of advanced machine learning techniques and hybrid models to enhance prediction accuracy further. Exploring ensemble methods that combine LSTM with other deep learning architectures or traditional machine learning algorithms could provide a more nuanced understanding of market trends. Additionally, incorporating sentiment analysis from news articles and social media can offer a more holistic view of the factors influencing stock prices. Continuous research and development in these areas will ensure that the predictive model remains cutting-edge and highly competitive in the fast-evolving field of stock market analysis.

6. Discussion and Conclusion

In summary, this project signifies a groundbreaking fusion of cutting-edge technology and meticulous research methodologies in the domain of stock market prediction. By amalgamating advanced machine learning algorithms with a user-centric web interface, this endeavor transcends the boundaries of traditional prediction methodologies, offering a sophisticated yet accessible tool for investors and analysts.

The pivotal aspect of this project lies in its multifaceted approach to comprehensively understand and predict stock market movements. Through an exhaustive consideration of diverse factors influencing stock prices, encompassing historical data analysis, sentiment analysis, and macroeconomic indicators, our model achieves a nuanced understanding of the intricate fabric that weaves stock market dynamics. This depth of analysis empowers the model to detect elusive patterns and subtle trends, presenting a holistic view that surpasses conventional analytical techniques.

Moreover, the adaptive nature of our machine learning model stands as a testament to its relevance amidst the ever-evolving financial landscape. Its ability to continually learn and adapt to changing market conditions positions it as a robust and forward-looking tool for navigating the uncertainties inherent in stock trading.

Furthermore, the impact of this project transcends academic boundaries, extending into the practical realm of real-world applications. By equipping stakeholders with a sophisticated predictive tool, this project contributes tangibly to the arena of informed decision-making in stock investments.

As the financial markets persist in their evolution, the scalability and adaptability of our model serve as a cornerstone for future developments in predictive analytics. This project not only adds depth to academic discourse but also provides a practical and invaluable asset for stakeholders in their pursuit of understanding and harnessing the volatility of the stock market.

7. References

1. L. Zhang, F. Zhang, Y. Lai, "Stock Price Prediction Using Machine Learning: A State-of-the-Art Review.",Journal Name: IEEE Access Volume(Issue): 8, 2020,Page Range:1- 41
2. Y. Chen, Y. Zhang, D. Huai,"A Machine Learning Approach to Stock Price Prediction Based on Technical Analysis.",Journal Name: International Journal of Economics, Commerce and Management Volume(Issue): 3(2), 2015,Page Range: 1-11
3. A. Dash, B. S. Behera"Stock Market Prediction Using Machine Learning Algorithms.",Journal Name: IOP Conference Series: Materials Science and Engineering,Volume(Issue): 225, 2017,Page Range: 1-31
4. V. Atanasov, C. Pirinsky, and Q. Wang, "The efficient market hypothesis and investor behavior," 2018.
5. D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, svm and lstm for stock market," in 2019 Amity International Conference on Artificial Intelligence (AICAI). IEEE, 2019, pp. 228–234.
6. Patel J, Shah S, Thakkar P, Kotecha K. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert systems with applications. 2015;42(1):259-68.
7. Vui CS, Soon GK, On CK, Alfred R, Anthony P. A review of stock market prediction with Artificial neural network (ANN). In: 2013 IEEE international conference on control system, computing and engineering. IEEE; 2013. p. 477-82.
8. Patel J, Shah S, Thakkar P, Kotecha K. Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications. 2015;42(4):2162-72.
9. Vargas MR, De Lima BS, Evsukoff AG. Deep learning for stock market prediction from financial news articles. In: 2017 IEEE international conference on computational intelligence and virtual environments for measurement systems and applications (CIVEMSA). IEEE; 2017. p. 60-5.
10. Ding X, Zhang Y, Liu T, Duan J. Deep learning for event-driven stock prediction. In: Twenty-fourth international joint conference on artificial intelligence; 2015.

11. https://www.investopedia.com/