NCIT

NAGARJUNA COLLEGE of Information Technology (Affiliated to T.U) A destination for quality education....

Nagarjuna ICT Club

Nagarjuna Information Communications Technology Club
(NCIT)

Integrating Technical Analysis with Predictive Models

A Study of Support and Resistance with LSTM and ARIMA Model in Stock Trading

Prepared by:
[Ashesh B. Bajracharyal]
[28980/078]

Department of Bachelor of Science in Computer Science and IT
Nagarjuna College of IT (NCIT)

Date:
[August 11, 2024]

Abstract

This study seeks to evaluate the effectiveness of a Support and Resistance trading strategy by implementing comprehensive back-testing strategy. The historical data of Banking Index from mid 2019 to present is used for the study. The main purpose of this research is to assess the worth of the Support and Resistance trading strategy by evaluating the profitability and its reliability. The strategy identifies support levels when a consecutive red candles making lower lows is followed by three green candles with higher lows, indicating a potential reversal point. Similarly, for identifying resistance level roles of green and red candles are inversed. The study yielded performance metrics such as profit and loss, win rate, profit rate, peak value, final value. Two ML Models LSTM Model and ARIMA Model were also used for Predictive analysis. Comparison between two models and their accuracy were also examined. The study concludes that while support and resistance strategy demonstrate potential profit, it is not without constraints so can be enhanced by integrating different other trading strategies.

Keywords: *Support, Resistance, Back-testing, Trading, Banking, Red candles, Green candles, Lower lows, Higher lows, Win rate, Profit rate, LSTM, ARIMA, Predictive analysis*

Contents

List of Figures

List of Tables

1 INTRODUCTION

In today's fast paced financial market, the ability to analyze the market effectively and to predict the movement of prices is crucial for successful and profitable trading. With the availability of sophisticated data, analysis tools and historical market data, Traders and investors are provided with a unique opportunities to refine their trading strategies and improve their decision-making processes. This research seeks to evaluate the comprehensive analysis of a trading strategy based on the principles of support and resistance, utilizing historical stock data to backtest the strategy's effectiveness.

Figure 1: Illustration of Support and Resistance Levels

The primary aim of this study is to backtest the trading strategy based on support and resistance and evaluate it's profitability. In the context of trading, Support and resistance are the fundamental concept of technical analysis to anticipate potential reversal points in the price movements. These levels are identified by analyzing historical price data and determined by labeling the points from where price has struggled to move beyond (resistance) or move below (support). Therefore, Support and resistance levels are crucial in technical analysis, serving as a means of understanding to aware the traders and investors for making the informed decisions about entering and exiting trade.

To achieve the objectives of this research, the study implements a combination of technical analysis and backtesting methodologies. Historical stock data is extracted from a reputable financial data provider, covering a data of Banking Index of NEPSE (Nepal Stock Exchange) from mid 2019 to the present. The data is then meticulously preprocessed, addressing any missing values, null values and outliers to ensure the accuracy and reliability of the analysis. After the data preprocessing, Support and resistance levels are identified with a feature engineering integrated with a unique logic. Thereby, Backtesting process involves simulating trades based on these identified levels, using statistical analysis to assess the performance of the strategy. Contrast to the backtesting of past historical data, machine learning models such as LSTM model and ARIMA model is also used for the predictive future analysis.

In the tailing of successful trading strategies and decision-making, it is important to comprehend the various factors that influences a price movement in financial stock market. This research focuses on analyzing historical stock data to evaluate the performance of trading strategy based on support and resistance. In this study, we explore a specific approach in identifying a support and resistance levels: a support level is determined when a sequence of consecutive red candles is followed by three consecutive green candles, suggesting a potential price reversal and indicating that price will not move below that point. Conversely, resistance is identified when a sequence of green candles is followed by three consecutive red candles, indicating that price will not go beyond that particular point.

This study aims to uncover the key patterns and correlation that can significantly impact the trading outcome. The analysis seeks to assess the particular strategy's effectiveness and identify areas where it may be optimized for improved performance. After identifying the support and resistance levels using the specified candlestick patterns, buy and sell signals were generated based on these levels. A buy signal was triggered when the price approached and respected a support level, indicating a potential upward reversal. Conversely, a sell signal was initiated when the price neared and respected a resistance level, suggesting a potential downward movement.

The evaluation of the effectiveness of the trading strategy is invoked using backtesting method. With the fictional initial investment amount, Backtesting is performed systematically over the selected period of time, with trades executed based on generated signals to evaluate the profitability and to draw down the other performance metrics such as profit and loss, earn percentage, win rate to assess the strategy's gainfulness and risk management. The outcomes of the backtesting provided insights into the strategy's reliability and potentiality for generating consistent returns under various market conditions.

This research primarily conducted on past historical data to evaluate the performance of trading strategy based on support and resistance, extends to further predictive future analysis. This involved leveraging advanced machine learning and statistical models to forecast the future price movements and enhance the strategy's effectiveness. Long Short-Term Memory (LSTM) which is a type of Recurrent Neural Network (RNN) particularly suited for time series forecasting, is implemented to predict future stock prices based on historical trends. Additionally, Auto-Regressive Integrated Moving Average (ARIMA) widely used for statistical method of time series analysis is applied to generate predictions. The comparison is done between the accuracy of outcome of LSTM Model with the outcome of ARIMA Model.

The outcomes of the analysis will be meticulously visualised to ensure the clear and comprehensive understanding of the strategy's performance. The identified support and resistance levels will be graphically represented on stock price charts, clearly marking the points where buy and sell signals are generated. The results of backtesting including key performance metrics will also be showcased to learn about the potentiality of the analysis. Additionally, The predictive outcomes of LSTM and ARIMA model will also be visualized to interpret the insights of the analysis. By presenting the analysis visually, the research ensured that the findings were not only precise but also easily interpretable, facilitating informed decision-making.

In summary, This research explores a trading strategy that identifies support and resistance levels using specific candlestick patterns. Buy and sell signals are generated at these levels, and the strategy is backtested with historical stock data, starting with an initial investment to evaluate its profitability and risk management. Additionally, predictive analysis is conducted using LSTM and ARIMA models to forecast future price movements and enhance the strategy's effectiveness. All outcomes, including support and resistance levels, buy/sell signals, predictive forecasts, and backtesting results, are visualized to provide clear and actionable insights into the strategy's performance.

1.1 PROBLEM STATEMENT

The project seeks to address the following key issues related to evaluation of trading strategy based on support and resistance:

1. **Identification of Effective Trading Signals:** Determine how well the strategy identifies support and resistance levels using specific candlestick patterns and generates reliable buy and sell signals based on these levels.

2. **Assessing Historical Performance:** Evaluate the profitability and risk management of the strategy through backtesting with historical stock data, analyzing metrics such as profit and loss, win rate and draw down.

3. **Predicting Future Price Movements:** Utilize advanced predictive models, including LSTM networks and ARIMA model to forecast future price movements and enhance the strategy's performance.

4. **Visualizing Outcomes:** Assess how effectively the strategy's outcomes, including identified levels, generated signals, predictive forecasts, and backtesting results are visualized to provide clear and comprehensive insights.

5. **Optimizing Strategy Performance:** Identify the areas for improvement in the strategy based on backtesting and predictive analysis results, and suggest refinements to enhance its effectiveness and adaptability in various market conditions.

1.2 PROJECT OBJECTIVE

1. **To backtest the support and resistance trading strategy** by analyzing historical stock data, generating buy and sell signals based on identified support and resistance levels, and evaluating the strategy's performance in terms of profitability, win rate, and risk management.

2. **To predict future stock movements** by employing advanced predictive models, including Long Short-Term Memory (LSTM) networks and Auto regressive Integrated Moving Average (ARIMA), to enhance the strategy and assess its potential for forecasting future price trends.

1.3 SCOPE AND LIMITATIONS

1.3.1 SCOPE

This project focuses on evaluating a trading strategy based on support and resistance levels with the following objectives:

1. **Data Analysis:** The project involves collecting and analyzing historical stock data to identify support and resistance levels, generate buy and sell signals, and evaluate the strategy's effectiveness.

2. **Methodological Approach:** It utilizes backtesting to measure the strategy's historical performance and applies predictive models, such as Long Short-Term Memory (LSTM) networks and Auto regressive Integrated Moving Average (ARIMA), to forecast future stock price movements.

3. **Objective:** The goal is to offer valuable insights that assist traders in making informed decisions, improving their strategies, and potentially maximizing their trading profits through precise signal generation and accurate price predictions.

1.3.2 LIMITATIONS

The project is subject to the following constraints:

1. **Data Limitations:** The analysis is constrained by the availability of historical stock data, which may limit the scope and depth of the backtesting and predictive analysis.

2. **Data Quality:** Variability in data quality, such as inconsistencies or gaps, may impact the reliability of the identified support and resistance levels and the predictions made by the models.

3. **Integration Challenges:** There may be difficulties in effectively integrating the backtesting results with the predictive analysis, potentially impacting the overall evaluation and effectiveness of the trading strategy.

1.4 PROJECT APPLICATIONS

The insights gained from this trading strategy evaluation have several practical applications:

1. **Trading Strategy Enhancement:** Refine and optimize trading strategies based on identified support and resistance levels and performance metrics to improve profitability and risk management.

2. **Predictive Analysis:** Utilize predictive models, such as LSTM and ARIMA, to anticipate future stock price movements, enabling more informed trading decisions and better preparation for market trends.

3. **Signal Generation:** Apply the buy and sell signals generated from the strategy to guide trading actions and potentially enhance decision-making processes in real-time trading environments.

4. **Market Analysis:** Use the findings to better understand market behavior and dynamics, facilitating more effective market analysis and strategic planning.

5. **Risk Management:** Leverage the backtesting results and predictive insights to develop robust risk management strategies, minimizing potential losses and maximizing returns.

2 Literature Review

The literature on the relationship between equity prices and aggregate investment highlights several key findings. Tobin's Q theory suggests that higher equity prices lead to increased investment due to the more favorable market value-to-replacement cost ratio. As European financial markets integrate, equity price correlations among countries strengthen, influencing similar investment behaviors and economic cycles. Studies on monetary policy indicate that changes in policy significantly affect stock prices and, consequently, investment decisions. Vector auto-regressive (VAR) models are commonly used to analyze these dynamics, showing how equity price shocks propagate through the investment channel. Overall, increased market integration and financial dynamics play crucial roles in shaping investment patterns and economic development across European countries. [1].

The study by Chung and Bellotti (2021) explores the behavior and effectiveness of support and resistance (SR) levels in financial time series, revealing their role as temporary price barriers that can reverse price trends. Their research introduces a heuristic algorithm designed to discover and evaluate SR levels within intraday price series. The findings indicate that SR levels with a higher number of price bounces are more likely to act as barriers for future price movements, highlighting the predictive power of these levels. The study also observes a decay in the effectiveness of SR levels over time, suggesting that their ability to influence price trends diminishes as they age. This research challenges the notion that SR levels can be fully explained by AR(1) processes or stationary models, underscoring their contribution to the temporary predictability and stationarity in financial time series. The insights from this paper emphasize the importance of SR levels in understanding price dynamics and improving trading strategies.[2].

This paper investigates the effectiveness of the support and resistance strategy in international financial markets, highlighting its role in identifying critical price levels where buying and selling pressures historically converge. By analyzing historical price actions and market trends, the study evaluates how this strategy can predict future price movements and enhance trading decisions. Utilizing a probability algorithm and the Sharpe ratio model across six financial markets from June 13, 2018, to June 13, 2023, the research finds that the support and resistance strategy provides significant benefits compared to a buy-and-hold approach, particularly in markets such as the JSE and DAX. The study emphasizes the importance of setting entry points slightly above support levels and exit points near resistance levels, along with employing stop-loss orders to manage risk effectively.[3].

The literature highlights the long-standing importance of support and resistance lines in stock trading, with traders traditionally relying on these lines as crucial technical indicators for determining buy and sell points. However, the challenge of accurately identifying these imaginary lines remains significant due to their complex and inconsistent nature. Recent studies have explored the use of automated methods, such as evolutionary optimization algorithms like Particle Swarm Optimization (PSO), to identify these lines, aiming to enhance trading decisions. The findings suggest that while optimized support and resistance lines can identify buy-sell points, their performance alone may not significantly outperform traditional Buy & Hold strategies. This indicates the need for integrating these automated lines with other technical and fundamental indicators within a comprehensive trading system to potentially achieve better results. The preliminary nature of these findings underscores the importance of further research to validate the effectiveness of automated support and resistance lines in diverse market conditions. [4].

3 METHODOLOGY

3.1 Waterfall Model

The waterfall model used in a project involves six key stages: Data Collection and Preprocessing with EDA, Support and Resistance Strategy Implementation, Backtesting Framework, Predictive Analysis Model, and Deployment of ML Model.

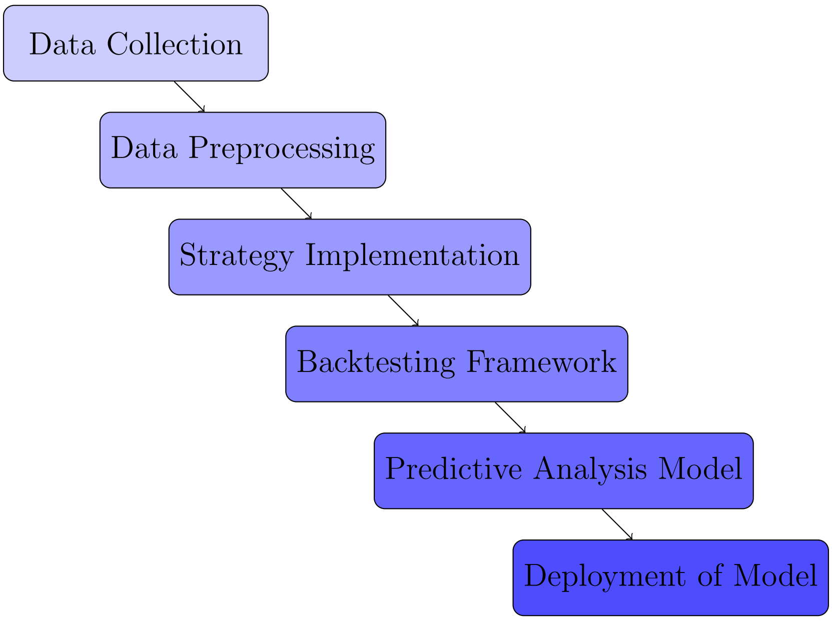

Figure 2: Waterfall Model Stages for Trading Strategy Development

In this project, we have employed the waterfall model for our analysis. This approach involves a systematic, step-by-step process where each phase builds upon the previous one. The waterfall model used in our project consists of following main stages:

1. **Data Collection:** This phase involves acquiring historical stock data from various reliable sources, including financial databases. The collected data encompasses features such as Symbol, Date, Open, High, Low, Close, Percent Change, and Volume. Ensuring the data's accuracy and relevance is crucial for effective analysis. The dataset should be comprehensive, covering a sufficient time period to capture various market conditions.

2. **Data Preprocessing and Exploratory Data Analysis (EDA):** After data collection, the next step is to preprocess the data to prepare it for analysis. In the phase of data cleaning, missing values

are handled by imputing categorical data with mode and numerical data with mean or median based on their susceptibility of outliers. Outliers are detected using visualizations such as box plots and statistical tests like Kolmogorov-Smirnov, Shapiro-wilk, Jarque-Bera test for normality. Histogram and QQ Plot is also applied for testing of normality. Methods like Interquartile Range (IQR) or Z-score are used for outlier detection. Feature engineering is applied in case of adding technical indicators like RSI and computing moving averages.

3. **Strategy Implementation:** This stage involves implementing the trading strategy, which focuses on support and resistance levels. The strategy is encoded using historical data to identify key support and resistance levels, generate buy/sell signals and simulate trading decisions based on these signals.

4. **Backtesting Framework:** Develop and apply a backtesting framework to evaluate the effectiveness of the trading strategy. This involves using historical data to simulate trading scenarios and measure the performance of the strategy. Key metrics such as return rate, win/loss ratio, win rate, equity peak, equity final are analyzed to assess the strategy's success.

5. **Predictive Analysis Model:** Integrate machine learning models like LSTM and ARIMA to enhance the trading strategy. These models are trained on historical data to forecast future stock movements and with the integration of buy and sell signals with future stock movements it maximizes the profitability. Model performance is evaluated using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) and also different plots of validation loss, residuals is also visualized to select the most effective model for predicting market trends.

6. **Deployment of Model:** The final stage involves deploying the selected machine learning model into a production environment. This includes setting up the necessary infrastructure to run the model in real-time, integrating it with trading systems and providing actionable insights. The deployment process ensures that the model is effectively operational and capable of making real-time trading decisions.

3.2 System Design and Implementations

3.2.1 Data Collection:

The dataset collected for this research focuses on stock market data and includes various attributes relevant to financial analysis and trading strategy development. Each record in the dataset corresponds to a trading day and encompasses the following features:

Table 1: Detailed description of dataset features for stock market analysis.

Feature	Type	Description
Symbol	Nominal	Stock or Index ticker symbol (e.g., BANKING, TSLA)
Date	Date	Specific trading date for the record
Open	Numeric	Opening price of the stock on the given date
High	Numeric	Highest price of the stock during the trading day
Low	Numeric	Lowest price of the stock during the trading day
Close	Numeric	Closing price of the stock on the given date
Percent Change	Numeric	Percentage change in the stock's price from the previous trading day
Volume	Numeric	Total number of shares traded on the given date

Figure 3: Banking Index Chart from mid2019 to present

3.2.2 Data Preprocessing and EDA

In the data preprocessing and exploratory data analysis (EDA) phase, we undertook several critical steps to ensure the quality and usability of the dataset:

1. **Handling Missing Values:** We first addressed missing values in the dataset. For numerical variables, missing values were imputed using the mean or median. The choice between mean and median was based on the data's sensitivity to outliers; the median was used if the data exhibited significant skew or outliers. For categorical variables, missing values were filled with the mode, which is the most frequently occurring value. Data cleaning is also applied such as commas and other symbols were removed in numerical data.

2. **Outlier Detection and Handling:** Outliers were identified through both visual and statistical methods. Visual plots, such as box plots, were used to detect potential outliers. Statistically, the Kolmogorov-Smirnov test and graphs such as histogram and QQ plot was employed to assess the normality of the data. For non-normally distributed data, the Interquartile Range (IQR) method was used for outlier detection. For normally distributed data, the Z-score method was applied. Outliers were removed if necessary, and any irrelevant columns were also eliminated to refine the dataset.

3. **Feature Engineering:** We calculated the RSI to measure the speed and change of price movements, providing insights into overbought or oversold conditions. We created moving average features (e.g., Simple Moving Average and Exponential Moving Average) to smooth out price data and identify trends over different time periods.

4. **Saving the Filtered Dataset:** After removing outliers and irrelevant columns, the cleaned dataset was saved. This dataset, now refined and free from unnecessary elements, is ready for use in machine learning models and further analysis.

5. **Data Transformation:** We proceeded with transforming the data to prepare it for machine learning models. Numerical features were standardized using the Standard Scaler to ensure that all features contribute equally to the model. Min Max scaler is used to scale the data and sequences are created. This step ensures that numerical data are represented appropriately making the data ready for modeling.

3.2.3 Strategy Implementation

In this section, we outline the process of implementing the trading strategy based on support and resistance levels, and how buy and sell signals were generated.

1. **Generating Support and Resistance Levels:**

- **Support Levels:** Support levels were identified by analyzing historical price data to find points where the stock price consistently found a lower bound. This involved scanning for multiple instances where the stock price touched a low point but failed to break through it, indicating strong buying pressure.
- **Support Function:** The support function is designed to detect potential support levels by analyzing the low prices of candlesticks within a specific window of time. For each candlestick, the function checks if the low price of the current and preceding candlesticks is lower than or equal to the previous one, indicating that the price is consistently finding a floor. If this condition is satisfied for all candlesticks in the specified range (`n1` periods before and `n2` periods after the current candlestick), the function confirms a support level.

```python
def support(df1, l, n1, n2):  # n1 and n2 are the number of candles before and after the current candle
    for i in range(l - n1 + 1, l + 1):
        if df1['Low'][i] > df1['Low'][i - 1]:
            return False
    for i in range(l + 1, l + n2 + 1):
        if df1['Low'][i] < df1['Low'][i - 1]:
            return False
    return True
```

Figure 4: Support function to generate support level

- **Resistance Levels:** Resistance levels were similarly identified by finding points where the stock price encountered a consistent upper bound, indicating strong selling pressure. These levels were determined by analyzing price data to locate multiple instances where the stock price failed to surpass a certain high point.

```python
def resistance(df1, l, n1, n2):  # n1 and n2 are the number of candles before and after the current candle
    for i in range(l - n1 + 1, l + 1):
        if df1['High'][i] < df1['High'][i - 1]:
            return False
    for i in range(l + 1, l + n2 + 1):
        if df1['High'][i] > df1['High'][i - 1]:
            return False
    return True
```

Figure 5: Resistance function to generate resistance level

- **Resistance Function:** The resistance function identifies potential resistance levels by analyzing the high prices of candlesticks. It checks whether the high price of the current and preceding candlesticks is higher than or equal to the previous one, suggesting that the price is consistently hitting a ceiling. The function confirms a resistance level if this condition is met for all candlesticks in the defined window (`n1` periods before and `n2` periods after the current candlestick).

- **Plotting Support and Resistance Lines:** After creating the candlestick chart, the identified support and resistance levels are plotted as horizontal lines across the chart. These lines provide a visual reference for where the price has historically found support or resistance.

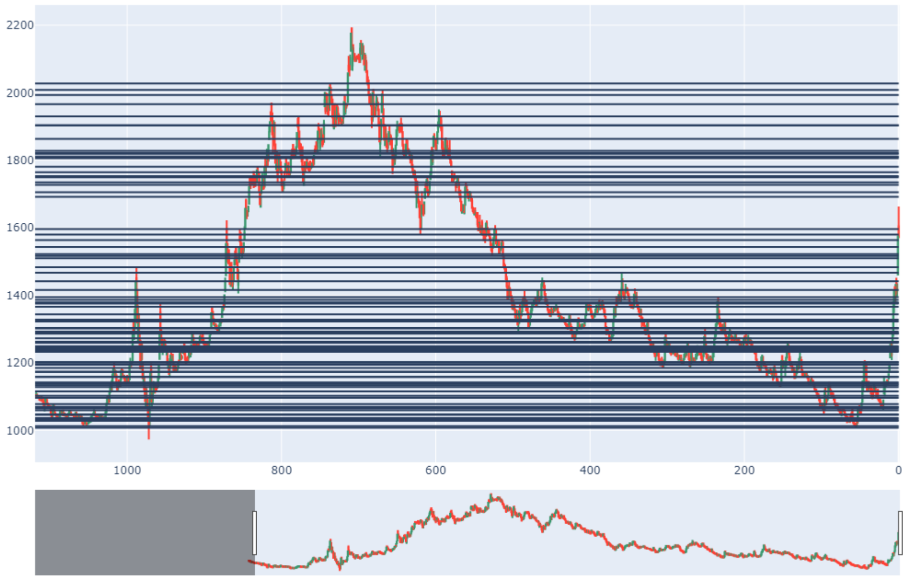

Figure 6: Lining of Support and Resistance Level

- **Visualization of Support and Resistance Levels with Color-Coded Lines:** Enhanced Visualization of Support and Resistance Levels: Green for Support, Red for Resistance.

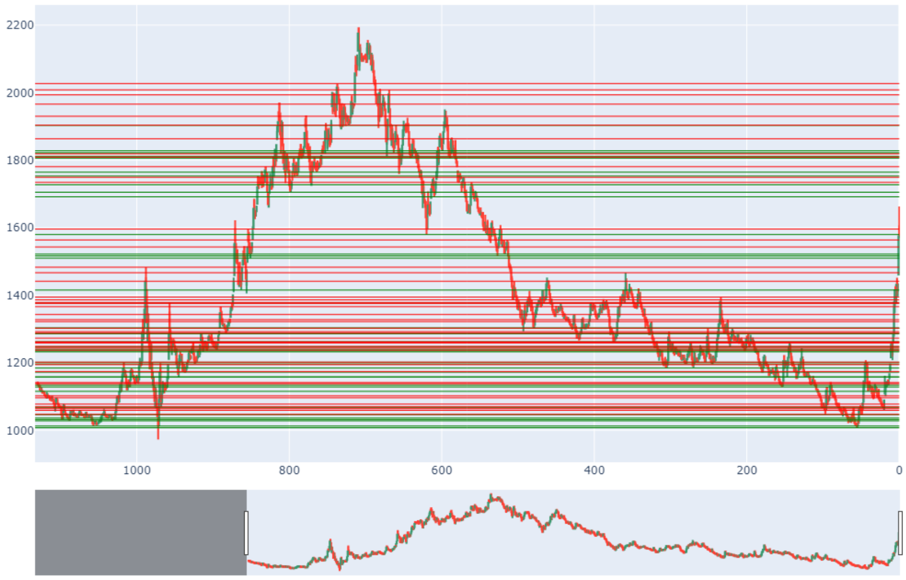

Figure 7: Lining of Support and Resistance level with green and red lines

2. **Generating Buy and Sell Signals**

   - **Buy Signals:** If the current price is close to any of the identified support levels (`ss`), a buy signal is generated. This is determined by the `closeSupport` function, which checks if the current price is within the defined tolerance of any support level. If this condition is met, `signal[row]` is set to `2`, indicating a buy signal.
   - **Sell Signals:** If the current price is close to any of the identified resistance levels (`rr`), a sell signal is generated. This is determined by the `closeResistance` function, which checks if the current price is within a certain tolerance (e.g., `150e-5`) of any resistance level. If the condition is met, `signal[row]` is set to `1`, indicating a sell signal.

```python
n1=2
n2=2
backCandles=30
signal = [0] * length

for row in range(backCandles, len(df)-n2):
    ss = []
    rr = []
    for subrow in range(row-backCandles+n1, row+1):
        if support(df, subrow, n1, n2):
            ss.append(df['Low'][subrow])
        if resistance(df, subrow, n1, n2):
            rr.append(df['High'][subrow])
    #!!!! parameters
    if (closeResistance(row, rr, 150e-5) ):#and df.RSI[row]<30
        signal[row] = 1
    elif(closeSupport(row, ss, 150e-5)):#and df.RSI[row]>70
        signal[row] = 2
    else:
        signal[row] = 0
```

Figure 8: Generating Buy and Sell Signals

3.2.4 Backtesting Framework:

In this section, a backtesting framework was created to backtest the trading strategy and draw down the performance metrics.

1. **Defining the Strategy Class:** The strategy is implemented using a class derived from 'Strategy':

   class MyCandlesStrat(Strategy):
   def init(self):
   super().init()
   self.signal1 = self.I(SIGNAL)

2. **Defining the Strategy Logic:** The 'next' method is called on each iteration of the backtest, processing each data point:

   def next(self):
   super().next()
   if self.signal1 == 2:
   sl1 = self.data.Close[-1] - 95
   tp1 = self.data.Close[-1] + 50

```
self.buy(sl=sl1, tp=tp1)
elif self.signal1 == 1:
sl1 = self.data.Close[-1] + 95
tp1 = self.data.Close[-1] - 50
self.sell(sl=sl1, tp=tp1)
```

3. **Signal Handling:** When a buy signal (2) is generated, a buy order is placed with a stop loss (`sl1`) set 95 units below the current close price and a take profit (`tp1`) set 50 units above the current close price. When a sell signal (1) is generated, a sell order is placed with a stop loss (`sl1`) set 95 units above the current close price and a take profit (`tp1`) set 50 units below the current close price.

4. **Running the Backtest:** The 'Backtest' class is used to run the strategy on historical data, providing performance metrics to evaluate its effectiveness.

```
bt = Backtest(df, MyCandlesStrat, cash=1000000, commission=.002)
stat = bt.run()
print(stat)
```

```
Start                                  0.0
End                                 1130.0
Duration                            1130.0
Exposure Time [%]                66.136163
Equity Final [$]             2211605.71432
Equity Peak [$]              2362072.24492
Return [%]                      121.160571
Buy & Hold Return [%]           -27.795464
Return (Ann.) [%]                      0.0
Volatility (Ann.) [%]                  NaN
Sharpe Ratio                           NaN
Sortino Ratio                          NaN
Calmar Ratio                           0.0
Max. Drawdown [%]               -29.280282
Avg. Drawdown [%]                -3.335728
Max. Drawdown Duration               322.0
Avg. Drawdown Duration           18.519231
# Trades                              89.0
Win Rate [%]                      67.41573
Best Trade [%]                    7.327852
Worst Trade [%]                 -10.106232
Avg. Trade [%]                    0.840899
Max. Trade Duration                   71.0
Avg. Trade Duration               8.382022
Profit Factor                     2.064211
Expectancy [%]                    0.899243
SQN                               2.691434
_strategy                   MyCandlesStrat
_equity_curve                      Equi...
_trades                   Size  EntryB...
dtype: object
```

Figure 9: Outcomes of Backtesting

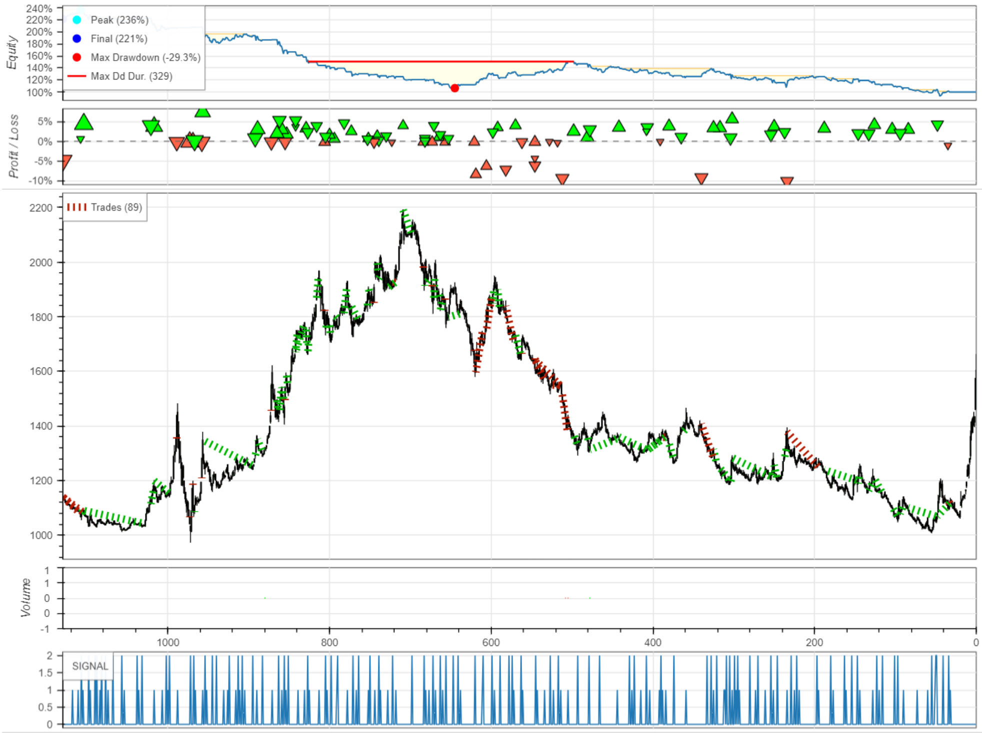

Figure 10: Buy and Sell Signals

3.2.5 Machine Learning Model Implementation

In this project, we evaluated two machine learning models to predict the future price movements. Each model was selected for its unique approach and suitability and for evaluating which model is best suited for predictive analysis. Below, we provide an overview of each model, including theoretical explanations and mathematical formulations.

1. **LSTM Model**

   (a) **Theory:** Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed to handle sequential data and learn long-term dependencies more effectively than traditional RNNs.

   (b) **Mathematical Formulation:** The forget gate determines which parts of the cell state to discard. It is computed as follows:

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

   where:

   - W_i and W_C are weight matrices for the input gate and candidate cell state, respectively.

- b_i and b_C are biases.
- tanh is the hyperbolic tangent activation function.

(c) **Applications:** LSTM (Long Short-Term Memory) networks are widely used for tasks involving sequential data due to their ability to capture long-term dependencies and mitigate the vanishing gradient problem. Applications of LSTMs include natural language processing, where they are employed for language modeling, machine translation, and sentiment analysis, as well as time series forecasting.

2. **ARIMA Model**

(a) **Theory:** The ARIMA (Auto Regressive Integrated Moving Average) model is a popular statistical approach for analyzing and forecasting time series data. It combines three components: auto regression (AR), which uses the dependency between an observation and several lagged observations; differencing (I), which involves differencing the raw observations to make the time series stationary; and moving average (MA), which models the dependency between an observation and a residual error from a moving average model applied to lagged observations.

(b) **Mathematical Formulation:** The ARIMA model is defined mathematically by combining its three components: auto regression (AR), differencing (I), and moving average (MA):

$$\phi(B)\Delta^d X_t = \theta(B)\varepsilon_t$$

where:

- $\phi(B)$ is the autoregressive polynomial of order p,
- Δ^d is the differencing operator of order d,
- $\theta(B)$ is the moving average polynomial of order q,
- ε_t represents the white noise or error term.

(c) **Applications:** The ARIMA (AutoRegressive Integrated Moving Average) model is widely used in time series forecasting to predict future values based on past observations. Its application spans various domains, including finance, economics, and engineering. For instance, in finance, ARIMA models are employed to forecast stock prices, interest rates, and economic indicators by analyzing historical data and identifying patterns and trends.

3.2.6 Machine Learning model Evaluations

(a) **Comparison of Machine Learning Models**

The table below summarizes the performance of both machine learning models used in this project. The performance is evaluated based on training and testing mean squared error and mean absolute error.

Table 2: Comparison of LSTM and ARIMA Models

Model	Training MSE	Training MAE	Testing MSE	Testing MAE
LSTM	0.0003	0.014	0.0001	0.0112
ARIMA	1503.99	13.908	19225.4194	121.6486

(b) **Backtesting after LSTM Implementation**

```
Start                                         0.0
End                                        1130.0
Duration                                   1130.0
Exposure Time [%]                       66.136163
Equity Final [$]                    2322273.56286
Equity Peak [$]                     2480281.71726
Return [%]                             121.168911
Buy & Hold Return [%]                  -27.795464
Return (Ann.) [%]                             0.0
Volatility (Ann.) [%]                         NaN
Sharpe Ratio                                  NaN
Sortino Ratio                                 NaN
Calmar Ratio                                  0.0
Max. Drawdown [%]                      -29.276661
Avg. Drawdown [%]                       -3.335676
Max. Drawdown Duration                      322.0
Avg. Drawdown Duration                  18.519231
# Trades                                     89.0
Win Rate [%]                             67.41573
Best Trade [%]                           7.327852
Worst Trade [%]                        -10.106232
Avg. Trade [%]                           0.840899
Max. Trade Duration                          71.0
Avg. Trade Duration                      8.382022
Profit Factor                            2.064211
Expectancy [%]                           0.899243
SQN                                       2.69188
_strategy                          MyCandlesStrat
_equity_curve                             Equi...
_trades                          Size   EntryB...
dtype: object
```

Figure 11: Backtesting outcome after LSTM Model

(c) **Plotting of Actual vs Predicted Price:**
Below are the plot of actual vs predicted price for the machine learning models used in this project:

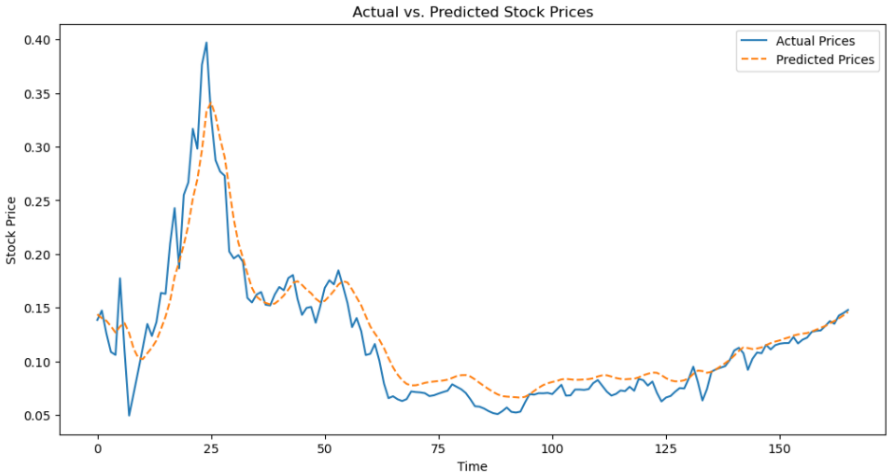

Figure 12: Actual vs Predicted price from LSTM Model

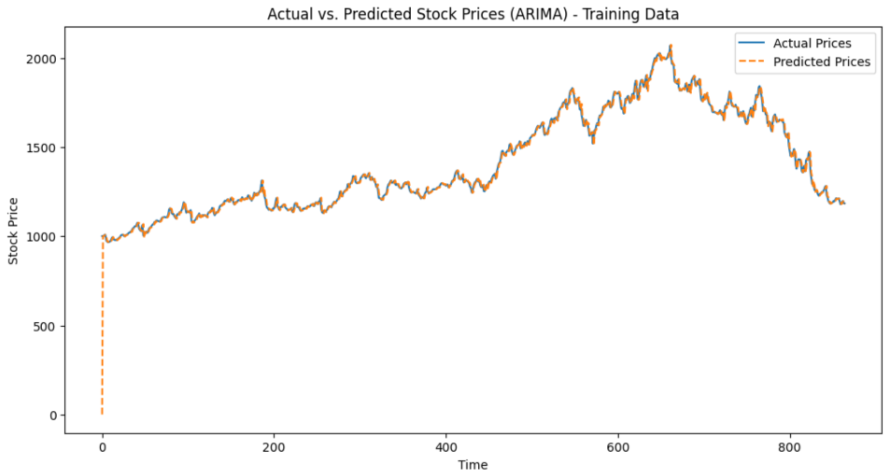

Figure 13: Actual vs Predicted price from ARIMA Model

(d) **Other plots to visualize the outcome of Models:**
Below are some other plots which showcases the residuals, validation loss and other outcomes of the model.

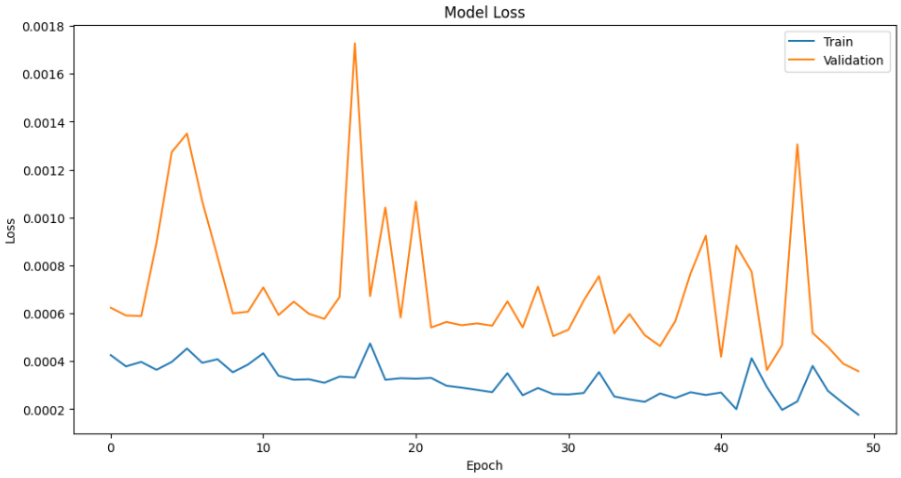

Figure 14: Model Loss

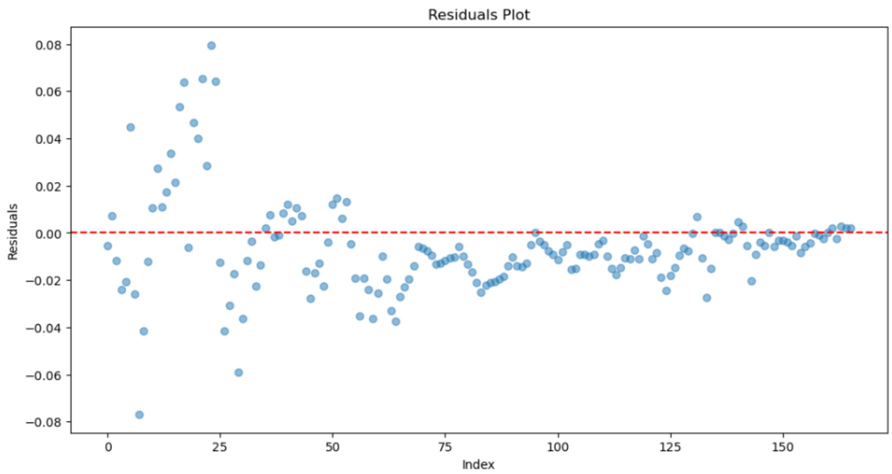

Figure 15: Residual plots

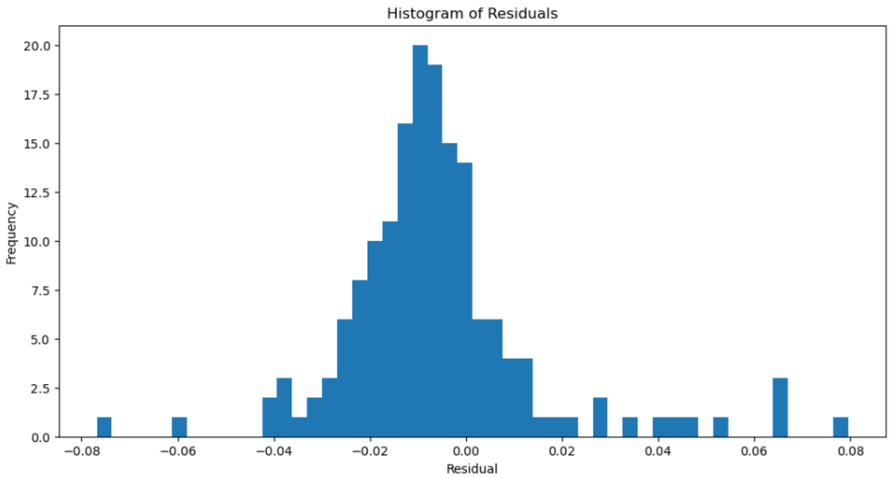

Figure 16: Histogram of Residuals

4 Deliverable Outputs

The output of this research project demonstrates the practical application of technical analysis and machine learning techniques in trading strategies. The focus was on generating support and resistance levels, creating actionable buy and sell signals, and evaluating the effectiveness of these signals through backtesting. Additionally, predictive analysis was performed using LSTM and ARIMA models to further enhance trading strategies.

In the initial phase, after the datasets were collected and preprocessed, we calculated support and resistance levels based on historical price data. These levels were instrumental in generating buy and sell signals, which were then rigorously backtested to assess their performance. The backtesting results showed promising outcomes, indicating that the support and resistance levels effectively guided trading decisions.

The subsequent phase involved predictive analysis using LSTM and ARIMA models. The LSTM model, with its ability to capture complex temporal patterns, achieved a significantly smaller error compared to the ARIMA model. Consequently, the LSTM model was chosen for final backtesting, which yielded favorable results, affirming its robustness in predicting future price movements. In contrast, the ARIMA model demonstrated substantial error, limiting its utility in this context.

The evaluation of the models was conducted using Mean Squared Error (MSE) and Mean Absolute Error (MAE) on both training and testing datasets. These metrics provided a comprehensive understanding of model performance, with MSE capturing the average squared difference between predicted and actual values, and MAE reflecting the average absolute differences. To further assess the models' effectiveness, various plots were visualized. The "Actual vs Predicted" plot illustrated the alignment of model predictions with actual values, offering a clear visual representation of accuracy. The "Validation Loss" plot tracked the model's performance over epochs, highlighting how well the model generalized to unseen data. Residual plots and histograms of residuals were used to analyze the distribution and patterns of prediction errors, ensuring that residuals were randomly distributed and not indicative of any underlying biases. These visualizations collectively provided a robust evaluation framework, confirming the superiority of the LSTM model in capturing price trends and demonstrating the limitations of ARIMA in this context.

Overall, this research provides valuable insights into the power of support and resistance levels in trading. By integrating these levels with advanced predictive models, the research offers traders a sophisticated tool for making informed trading decisions. The successful implementation of LSTM highlights its potential for enhancing trading strategies, while the limitations of ARIMA emphasize the importance of model selection in trading applications.

5 Project Task Timeline Schedule/Gantt Chart and Usecase Diagram

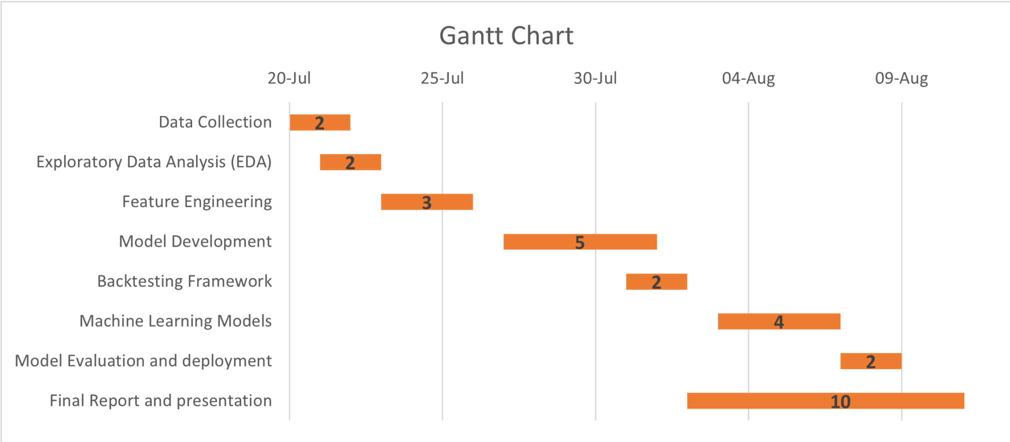

Figure 17: Project Task Timeline Schedule

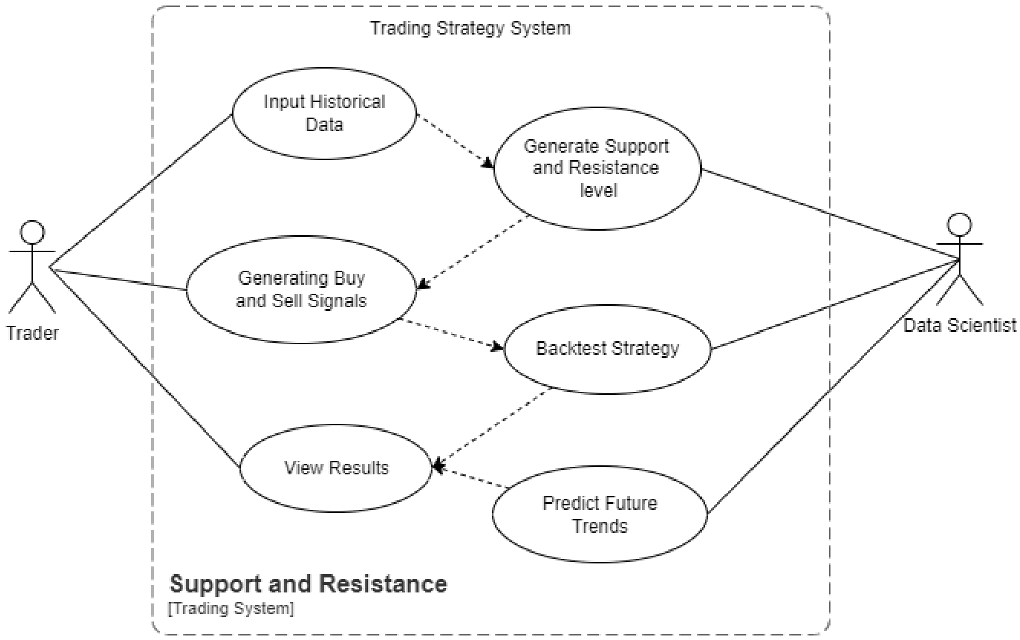

Figure 18: Project Use Case Diagram

6 REFERENCES

[1] H. Manh, Duong, B. Siliverstovs, Manh, and H. Duong, "The stock market and investment," Jul. 2006.

[2] K. Chung and A. Bellotti, *Evidence and behaviour of support and resistance levels in financial time series*, Jan. 2021.

[3] S. Tabot, "Investigating the merits of support and resistance strategy: Evidence from international financial markets," *Journal of economic and social development*, vol. 10, pp. 75–80, Sep. 2023.

[4] E. O. Yıldırım, M. Uçar, and A. M. Özbayoğlu, "Evolutionary optimized stock support-resistance line detection for algorithmic trading systems," in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, 2019, pp. 1–6. DOI: 10.1109/UBMYK48245.2019.8965471.

7 APPENDIX

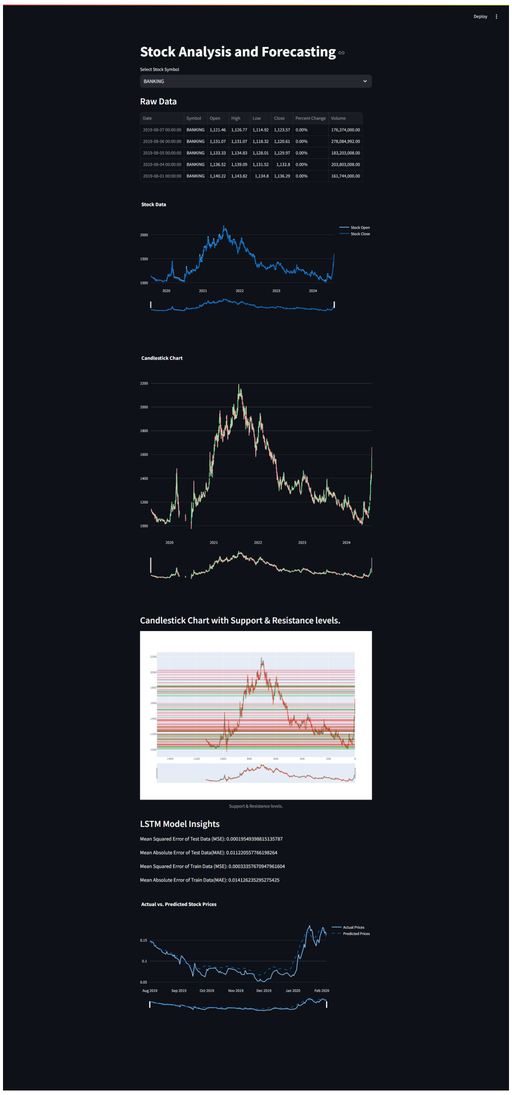

Figure 19: Support and Resistance Detection with Prediction