International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 10 | Mar 2024 www.irjet.net p-ISSN: 2395-0072

Algorithmic Trading with an API

[1]Gaurangi Raul, [2]Riya Jadhav, [3]Tejas Kamble, [4]Kshitija Satpute

[1,2,] Dept. of CSE (AI&ML), Smt. Indira Gandhi College of Engineering, Navi Mumbai, Maharashtra, India

---

Abstract - *Algorithmic trading, uses pre-programmed instructions to automate trade executions. This research paper investigates the development of an algorithmic trading system using the Kotak API, focusing on the Relative Strength Index (RSI) strategy.*

Through APIs, such as WebSocket traders can seamlessly connect their algorithms to brokerage platforms, allowing for real-time data access, order management, and trade execution with high precision and speed.

Key Words: Kotak Neo API, Yahoo Finance, RSI algorithm

1.INTRODUCTION

The financial markets have witnessed a significant shift towards automation with the rise of algorithmic trading (algo-trading). This approach leverages computer programs to execute trades based on pre-defined rules and technical indicators, offering advantages like speed, accuracy, and the ability to remove emotions from trading decisions.

This research paper explores the development of an algorithmic trading system specifically designed for the Kotak Securities platform. We will focus on utilizing the Relative Strength Index (RSI) strategy, a popular momentum oscillator, to identify potential trading opportunities.

The RSI analyzes recent price movements to gauge whether a stock is overbought or oversold, providing valuable insights for buy and sell decisions. By integrating the Kotak API and implementing the RSI strategy, this research aims to develop a robust algorithmic trading system.

The following sections will delve into the details of Kotak API integration, RSI indicator calculation, the design of the RSI-based trading strategy, back testing and evaluation methodologies, and a paper trading simulation for live market testing. Through this comprehensive approach, we will assess the feasibility and potential benefits of algorithmic trading using the Kotak API and the RSI strategy.

1.1 API

A trading API (Application Programming Interface) acts as a bridge between your trading platform and your custom-built application or software. It allows programmatic access to features like:

Market Data: Real-time or historical data on stock prices, volumes, order book depth, and other market information.

Order Management: Ability to place buy and sell orders, modify existing orders, and cancel orders electronically.

Portfolio Management: Access to your current holdings, account balance, and transaction history.

1.2 KOTAK API

Kotak Mahindra Bank offers APIs relevant to algorithmic trading:

1. **Kotak Neo Trade API (offered by Kotak Securities):** This API caters specifically to algorithmic trading and provides access to features like:

   Real-time market data: Get live updates on stock prices, volumes, and other market information.

   Order placement and management: Programmatically place buy and sell orders, modify existing orders, and cancel orders.

   Live market feeds: Integrate real-time market data into your trading application

2. **Y Finance**

   Yahoo Finance is generally considered a reliable source for historical stock data. Yahoo Finance allows users to download historical price, dividend, and split data for most stocks in daily, weekly, or monthly formats. This data can be downloaded as a CSV file to be used offline. In summary, Yahoo Finance stands out as a preferred choice for casual investors and traders seeking quick access to news, quotes, and market data, making it a valuable tool for those looking for a user-friendly interface, advanced research capabilities, and portfolio tracking features

2.LITERATURE SURVEY

Algorithmic trading bots and price prediction are being used a lot in the Indian stock market. These bots use special instructions to make trades faster and more accurately than humans. Price prediction helps traders decide when to buy or sell stocks by using different techniques like looking at past patterns or using computers to learn from data. Kotak Securities, a company in India, has made a tool called Kotak Neo API that helps traders use these bots and make trades easily. People have done studies to see how well these bots

and price prediction work in India, and they have found that it can be successful, but it also depends on the specific bot, the market conditions, and how much risk the trader is willing to take.

There is an increasing trend of research into the efficiency of algorithmic trading and price prediction in the Indian markets. A study titled as Algorithmic Trading in Indian Stock Market examines different algorithmic trading strategies used in India. The study tells the importance of algorithm design and market conditions in the performance of algorithmic trading strategies. Another study tells that algorithmic trading can indeed be profitable. However, there is some risk associated that one needs to be aware of.

Kotak Neo Trade is a powerful APIs that can help you develop your own trading Terminal and implement your strategies easily and intuitively. This APIs give you open access to live market feeds, orders, live positions, and a lot more. It has a user-friendly interface and it works well with various platforms therefore, it is convenient for both individual and institutional investors

One of the emerging areas in algorithmic trading is called reinforcement learning. Reinforcement learning algorithms learn in doing; interacting with the environment and receiving feedback. Early work on algorithmic trading by **Harris (2003)** and **Chan (2009)** set the foundational principles, emphasizing the role of automation in improving the efficiency of trade execution and minimizing the market impact of large orders. These studies highlighted how algorithmic trading systems could optimize execution strategies by reducing human intervention, particularly in volatile markets. While these studies discussed the benefits of automated trading, they lacked a direct focus on APIs, as APIs had not yet become central to the infrastructure of financial markets.

The importance of APIs became more apparent as trading strategies grew more sophisticated and market conditions more dynamic. **Kissell (2014)** brought attention to the role of APIs in algorithmic trading, describing them as the bridge that connects trading algorithms with data sources and brokerage systems. This research underscored how APIs enable automated systems to access real-time data, place orders, and monitor trades without requiring human input. Kissell further differentiated between types of APIs, such as RESTful APIs, which allow for specific requests like historical data retrieval, and WebSocket APIs, which facilitate real-time data streaming. His work was instrumental in showing how different API protocols could cater to different trading needs, such as high-frequency trading or more traditional long-term strategies.

Narayan (2016) delved deeper into the application of APIs in high-frequency trading (HFT), where speed is crucial for success. Narayan demonstrated that APIs with low latency—those that can process data and execute orders within milliseconds—are essential for HFT algorithms. He also explored how modern APIs allow traders to exploit short-lived arbitrage opportunities, a strategy that requires extremely fast reaction times. His study provided evidence that APIs significantly enhance the execution of such time-sensitive trades, contributing to the growing body of research that emphasizes the role of technology in modern trading.

In 2019, Li Xucheng and Peng Zhihao proposed a new algorithmic trading model combining artificial intelligent into it. While this study demonstrated the profitability of applying reinforcement learning algorithms towards algorithmic trading, albeit marginal, net profit.

Ye and Dejun (2018) developed event-based algorithmic trading strategies involving the stock news, earnings reports, and sentiment changes, all leading to stock price motion. According to the authors, they put forward a trading system that applies moving average, searches for candlestick patterns in search of reversal points and places orders automatically. They however stressed on the importance of contingency planning, risk management and the complexity associated therein while accurately predicting the market reaction.

Recent studies have continued to explore the cutting-edge applications of APIs in algorithmic trading. **Shah (2020)** focused on the growing trend of cloud-based APIs, which allow traders to access computational power and data storage remotely, thus reducing the infrastructure costs traditionally associated with algorithmic trading. Shah argued that cloud-based APIs enable greater scalability, allowing traders to access multiple markets simultaneously and run more complex algorithms without the need for expensive on-premise hardware. **Gupta (2021)** took this a step further by investigating AI-driven APIs that provide predictive analytics and sentiment analysis, tools that are increasingly being used to enhance algorithmic decision-making.

Technical analysis is amongst the oldest and popular techniques for price prediction. This involves studying the past market prices, the chart patterns to detect the trends and forecast the prices movements in the future.

Technical analysts use moving averages, support and resistance levels and a variety of other technical indicators to understand the possible market directions for trading which will help the traders during buying and selling. Whereas technical analysis considers historical data on price, fundamental analysis is a more thorough examination of the company's strengths and weaknesses in order to discover the actual value. It is based on considering different financial ratios like earnings, revenue, the debt levels and the expertise of the management in order to get information about the company's prospects in the future.

3. PROPOSED STRATEGY [RSI]

This strategy utilizes the Relative Strength Index (RSI) to identify potential buy and sell signals within the Kotak algorithmic trading environment.

Core Functionality:

- **RSI Calculation:** The strategy will calculate the RSI value for a chosen security using historical price data retrieved via the Kotak Neo Trade API. Libraries like TA-Lib or pandas-ta can be used for this purpose.
- **Overbought/Oversold Thresholds:** The strategy will define specific RSI thresholds to identify overbought (above a certain value) and oversold (below a certain value) conditions in the security. Common RSI thresholds are 70 for overbought and 30 for oversold, but these can be customized based on back testing and risk tolerance.
- **Trading Signals:**

  Buy Signal: If the RSI falls below the oversold threshold, the strategy will generate a buy signal, indicating a potential buying opportunity.

  Sell Signal: Conversely, if the RSI rises above the overbought threshold, the strategy will generate a sell signal, prompting the sale of the security.

Additional Considerations:

- **Stop-Loss Orders:** The strategy should incorporate stop-loss orders to limit potential losses if the price moves against the anticipated direction. Stop-loss orders can be placed automatically using the Kotak Neo Trade API.
- **Exit Strategy:** While RSI provides entry and exit signals based on overbought/oversold conditions, additional technical indicators or price action confirmation might be implemented for a more robust exit strategy.
- **Back testing and Optimization:** The strategy's effectiveness should be rigorously evaluated through back testing on historical data. This helps refine the RSI thresholds, identify potential weaknesses, and optimize the strategy for the chosen market and timeframe.

Fig.1.RSI index

Benefits:

- **Automated Trading:** The strategy automates trade execution based on pre-defined RSI signals, removing emotions from the decision-making process.
- **Trend Identification:** RSI can help identify potential trends by analyzing the speed and magnitude of price movements.
- **Risk Management:** Stop-loss orders can be integrated to limit potential losses.

4.METHODOLOGY

Development Environment Setup:

Programming Language: Choose a suitable programming language like Python for its extensive libraries and community support for algorithmic trading.

Libraries: Install necessary libraries for data manipulation (pandas), technical analysis (TA-Lib or pandas-ta), API integration (requests), and potentially charting (Matplotlib).

Kotak Neo Trade API Integration: Obtain API credentials from the Kotak Developer Connect portal and integrate the API within your code using the provided libraries.

Data Acquisition:

Security Selection: Define the security (stock, ETF) you want to trade and ensure it's supported by the Kotak Neo Trade API.

Historical Data Retrieval: Utilize the Kotak Neo Trade API to retrieve historical price data for the chosen security. This data will be used for RSI calculation and back testing.

RSI Calculation:

Formula Implementation: Implement the RSI formula within your code using libraries like TA-Lib or pandas-ta. The formula considers the average of recent gains and losses to gauge the momentum and identify overbought/oversold conditions.

RSI Threshold Definition: Set specific RSI thresholds to define overbought (typically above 70) and oversold (typically below 30) zones. These thresholds can be adjusted based on back testing results and risk tolerance.

Trading Strategy Design:

Signal Generation: Based on the calculated RSI values and defined thresholds, develop code to generate buy and sell signals.

Order Management: Integrate order placement functionalities with the Kotak Neo Trade API to automatically execute buy and sell orders based on generated signals.

Stop-Loss Integration: Implement stop-loss orders to limit potential losses if the price moves against the anticipated direction. Stop-loss orders can be placed automatically using the Kotak API.

Back testing and Evaluation:

Historical Data Simulation: Utilize the retrieved historical price data to simulate the trading strategy's performance over a specific period. This helps assess profitability, win rate, and risk metrics.

Performance Analysis: Analyze the back testing results to evaluate the strategy's effectiveness. Identify potential weaknesses and optimize parameters like RSI thresholds based on the findings.

Paper Trading Simulation (Optional):

Simulated Environment: Once the strategy is back tested and refined, consider a paper trading simulation. This involves running the strategy with virtual funds in a simulated market environment offered by some brokers. This provides a real-world testing ground without risking actual capital.

Performance Monitoring: Continuously monitor the strategy's performance during paper trading and make further adjustments if necessary.

Live Deployment:

Real Market Execution: After thorough testing and refinement, the strategy can be deployed with real capital. However, proceed with caution and start with a small amount to assess live market performance.

Risk Management: Always prioritize risk management by setting appropriate stop-loss orders and position sizing strategies.

Ongoing Monitoring and Refinement:

Market Conditions: Algorithmic trading strategies require ongoing monitoring and adaptation as market conditions evolve.

Performance Tracking: Regularly track the strategy's performance and make adjustments as needed based on real-time data and market changes.

5. WORKFLOW

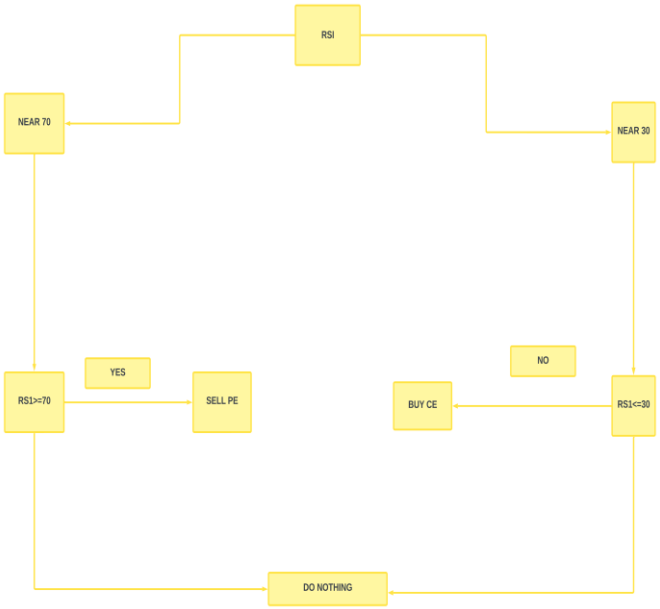


Fig.2 Algorithm workflow

Data Acquisition and token generation:

Integrating with exchange and market data APIs to fetch real-time market information. Data acquisition in an algorithmic trading bot involves gathering real-time market data from various sources such as cryptocurrency exchanges and market data aggregators. This data includes price, volume, and other relevant market information necessary for the bot to analyze and identify potential trading opportunities Additionally, the data acquisition process may involve historical market data analysis to evaluate the performance of trading strategies before live execution in the markets.

These tokens serve as authentication credentials that allow the bot to securely connect to the exchange APIs and execute

trades on behalf of the trader. By generating and utilizing these tokens, the trading bot gains the necessary permissions to access real-time market data, place buy and sell orders, and implement trading strategies effectively within the exchange environment

Fetch historical data:

To fetch historical data from Yahoo Finance, you can use Python libraries like yfinance to access financial information and historical market data. By utilizing the Ticker function from yfinance, you can retrieve various data sets such as financial ratios, historical prices, and key metrics for specific companies listed on Yahoo Finance. Additionally, you can specify the period of time for which you want historical data, whether it's for a specific date range or a predefined period like 1d, 5d,1mo,6mo,1y, and more.

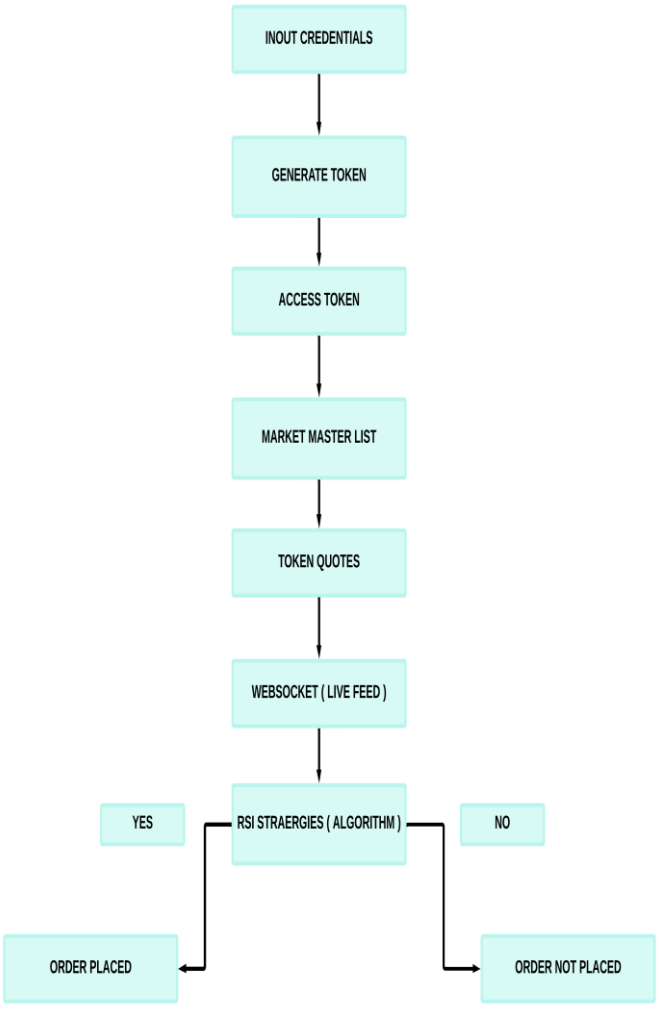


Fig.3 Functional workflow

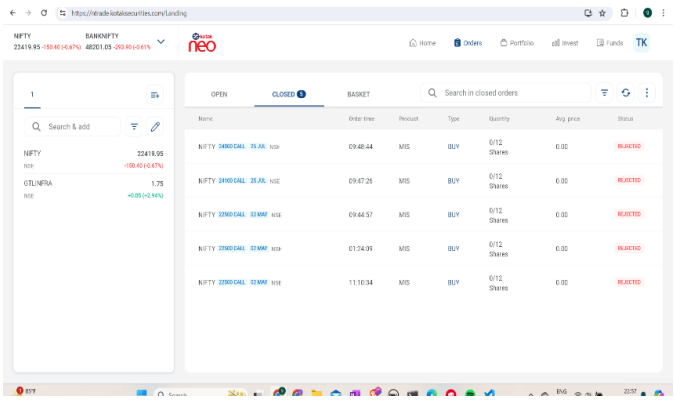
Strategy Implementation:

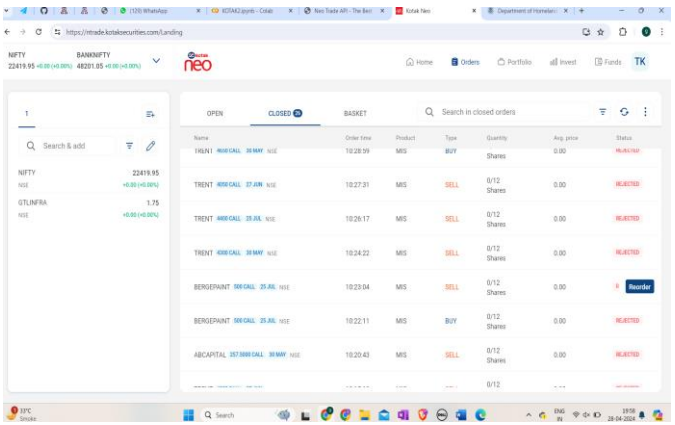
The RSI is a technical analysis indicator that measures the momentum of a price movement. It is often used in combination with other indicators, such as the Stochastic Oscillator, to generate buy and sell signals. Back testing and optimization are crucial steps in developing an RSI-based trading bot. The bot's performance should be evaluated using historical market data to ensure the trading strategy is profitable before deploying it in live trading. In summary, the RSI strategy can be effectively implemented in an algorithmic trading bot to automate the trading process and potentially capitalize on market opportunities.

Monitoring and Analysis

A successful algorithmic trading bot relies on adaptive strategies that can detect patterns in vast amounts of data. These strategies govern the bot's buying and selling decisions. In summary, monitoring and analyzing orders placed by an algorithmic trading bot involve real-time visualization, surveillance for compliance and risk management, constant monitoring of market activities, and the implementation of adaptive trading strategies.

6. RESULT





7. CONCLUSIONS

This research explored the development of an algorithmic trading system utilizing the Kotak Neo Trade API and the Relative Strength Index (RSI) strategy. The project focused on integrating the API for real-time market data access and order execution. We discussed the RSI indicator calculation and its use in generating buy and sell signals based on overbought/oversold conditions. The methodology outlined back testing for strategy optimization and risk management through stop-loss orders. While WebSocket's were mentioned as a potential future technology for real-time data communication, the initial focus remained on the core RSI strategy and the Kotak API functionalities. Overall, this project provided a framework for building an algorithmic trading system using the Kotak Neo Trade API and the RSI strategy, emphasizing the importance of back testing, optimization, and risk management practices before deploying real capital.

8. FUTURE SCOPE

I. Machine Learning and Artificial Intelligence (AI): Integrating machine learning algorithms into trading strategies can lead to more sophisticated pattern recognition, risk assessment, and potentially superior performance. AI could analyze vast datasets and identify complex market relationships that traditional indicators might miss.

II. High-Frequency Trading (HFT): HFT strategies leverage advanced algorithms and hardware to execute trades at ultra-fast speeds, capitalizing on fleeting market inefficiencies. As technology continues to evolve, HFT is expected to become even more prevalent, demanding ever-faster and more efficient algorithmic trading systems.

III. Regulation and Transparency: As algorithmic trading grows, regulatory frameworks will need to adapt to ensure fairness, transparency, and market stability. Algorithmic transparency and potential biases within these systems will be crucial areas of discussion.

IV. Democratization of Algorithmic Trading: Traditionally, algorithmic trading was the domain of hedge funds and large institutions. Advancements in technology and user-friendly platforms could democratize algorithmic trading, allowing individual investors to participate with their own strategies.

9. REFERENCES

Algorithmic Trading:

- "Advances in Financial Machine Learning" by Marcos Lopez de Prado (Book)
- "Quantitative Trading: Building Your Own Algorithmic Trading Business" by Ernest Chan (Book)
- https://www.investopedia.com/articles/trading/11/automated-trading-systems.asp

Kotak Neo Trade API:

- While official public documentation might be limited, try exploring resources within the Kotak Developer Connect portal after registering: https://www.kotak.com/en/open-banking.html

REST API Integration with Python:

- "Python for Algorithmic Trading" by Yves Hilpisch (Book)
- https://realpython.com/lessons/rest-apis-ninja/

Technical Analysis Indicators:

- "Technical Analysis of the Financial Markets" by John Murphy (Book)
- https://www.investopedia.com/terms/r/rsi.asp

Backtesting Algorithmic Trading Strategies:

- "Backtesting Strategies for Algorithmic Trading" by Yves Hilpisch (Book)
- https://www.youtube.com/watch?v=EO6mp3bJWhs