International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Automated Options Trading Using AI

Prajwal S[1], Nandeesh[2], Niharika K Murthy[3], Niharika N[4]
[1][2][3][4] UG Students, DEPARTMENT OF ELECTRONICS OF ELECTRONICS AND COMMUNICATION,
BMS COLLEGE OF ENGINEERNIG, Bangalore, India

Abstract

Algorithmic trading is a method of executing orders using automated pre-programmed trading instructions accounting for variables such as time, price, and volume. This type of trading attempts to leverage the speed and computational resources of computers relative to human traders.

With the help of an algo trading computer program will automatically monitor the stock price and place the buy and sell orders when the defined conditions are met. The trader no longer needs to monitor live prices and graphs or put in the orders manually. The algorithmic trading system does this automatically by correctly identifying the trading opportunity.

This is the first in a series of articles dealing with machine learning in asset management. Asset management can be broken into the following tasks: (1) portfolio construction, (2) risk management, (3) capital management, (4) infrastructure and deployment, and (5) sales and marketing. This article focuses on portfolio construction using machine learning. Historically, algorithmic trading could be more narrowly defined as the automation of sell-side trade execution, but since the introduction of more advanced algorithms, the definition has grown to include idea generation, alpha factor design, asset allocation, position sizing, and the testing of strategies. Machine learning, from the vantage of a decision-making tool, can help in all these areas.

1. INTRODUCTION

Algorithmic trading, also known as algo trading or automated trading, is the use of computer algorithms to execute trades in financial markets. It involves the use of predefined sets of rules and instructions that determine when and how to place orders, without the need for human intervention. The history of algo trading can be traced back several decades, with significant developments occurring over time. Here is a brief overview:

Early Beginnings (1970s-1980s): In the 1970s, financial institutions started using computers to automate certain aspects of trading. The introduction of electronic exchanges in the 1980s provided a platform for algorithmic trading to flourish. Institutional investors and large financial firms were the primary adopters of early algorithmic trading systems.

Rise of High-Frequency Trading (1990s-2000s):

The development of faster and more sophisticated computer systems in the 1990s facilitated the rise of high-frequency trading (HFT). HFT strategies aim to exploit small price discrepancies and execute trades within microseconds or even nanoseconds. Advancements in telecommunications infrastructure, such as direct market access (DMA), further accelerated the growth of algo trading.

Algorithmic Trading Goes Mainstream (2000s-2010s): As technology became more accessible, algo trading expanded beyond institutional players to encompass hedge funds and proprietary trading firms. The proliferation of algorithmic trading was fueled by increased computing power, advanced trading platforms, and improved data availability. Various algorithmic trading strategies emerged, including statistical arbitrage, trend following, mean reversion, and execution algorithms. Regulatory Challenges and Flash Crashes (2010s): The "Flash Crash" of May 6, 2010, highlighted the risks associated with algorithmic trading when abnormal market conditions occur. Regulatory bodies, such as the U.S. Securities and Exchange Commission (SEC), responded with increased oversight and the implementation of new regulations.

Regulations, such as the Markets in Financial Instruments Directive II (MiFID II) in Europe, aimed to improve market transparency and reduce potential risks.

Machine Learning and Artificial Intelligence (2010s-2020s):

The use of machine learning and artificial intelligence (AI) techniques gained prominence in algorithmic trading.

These technologies enable traders to analyze vast amounts of data, identify patterns, and make more informed trading decisions. Machine learning algorithms can adapt and improve over time, allowing for the development of more sophisticated trading strategies.

Ongoing Developments: Algorithmic trading continues to evolve, with ongoing advancements in technology, data analytics, and market infrastructure. The integration of alternative data sources, such as social media sentiment analysis and satellite imagery, is enhancing trading strategies. Market participants are also exploring the potential of blockchain technology and cryptocurrencies in algorithmic trading. Overall, the history of algo trading showcases its progression from early computerized trading to the present-day complex systems that leverage advanced technologies and data analysis techniques. It has become an integral part of financial markets, providing efficiency and liquidity while posing challenges that require ongoing regulatory scrutiny.

2. LITERATURE REVIEW

1. Hongyang Yang, Xiao-Yang Liu, Christina Dan Wang , “FinGPT: Open-Source Financial Large Language Models” Large language models (LLMs) have shown the potential of revolutionizing natural language processing tasks in diverse domains, sparking great interest in finance. Accessing high-quality financial data is the first challenge for financial LLMs (FinLLMs). While proprietary models like BloombergGPT have taken advantage of their unique data accumulation, such privileged access calls for an open-source alternative to democratize Internet-scale financial data. In this paper, we present an open-source large language model, FinGPT, for the finance sector.
2. Ruibo Chen, Wei Li, Zhiyuan Zhang, Ruihan Bao, Keiko Harimoto, Xu sun “Stock Trading Volume Prediction with Dual-Process Meta-Learning” Volume prediction is one of the fundamental objectives in the Fintech area, which is helpful for many downstream tasks, e.g., algorithmic trading. Previous methods mostly learn a universal model for different stocks. However, this kind of practice omits the specific characteristics of individual stocks by applying the same set of parameters for different stocks. On the other hand, learning different models for each stock would face data sparsity or cold start problems for many stocks with small capitalization.

3. Tudor-Vlad Pricope," Deep Reinforcement Learning in Quantitative Algorithmic Trading: A Review" We can look at the stock market historical price series and movements as a complex imperfect information environment in which we try to maximize return - profit and minimize risk. This paper reviews the progress made so far with deep reinforcement learning in the subdomain of AI in finance, more precisely, automated low-frequency quantitative stock trading.
4. V. Gupta and E. Kumar, "Review on Machine Learning Techniques for International Trade Trends Prediction" Analyzing the economic trends and knowing the future value of important economic variables makes a country more efficient in country's economic planning and developing policies. This can be achieved by increased application of machine learning more effectively and accurately.
5. Elior Nehemya, Yael Mathov, Asaf Shabtai, Yuval Elovici," Taking Over the Stock Market: Adversarial Perturbations Against Algorithmic Traders" . Stock market traders utilize machine learning models to predict the market's behavior and execute an investment strategy accordingly. However, machine learning models have been shown to be susceptible to input manipulations called adversarial examples. Despite this risk, the trading domain remains largely unexplored in the context of adversarial learning. In this study, we present a realistic scenario in which an attacker influences algorithmic trading systems by using adversarial learning techniques to manipulate the input data stream in real time. The attacker creates a universal perturbation that is agnostic to the target model and time of use, which, when added to the input stream, remains imperceptible.

3. EXSING SYSTEM

They obtain the optimal strategy, defined as the strategy that minimizes the expected cost of trading under their assumptions (and considering only market orders), via Dynamic Programming and Stochastic Control respectively. Extensions to the original approach that model additional market phenomena via stochastic processes include. Another branch of the literature focuses instead on the modelling of the order book directly again via stochastic modelling and solved via stochastic control techniques. The main drawback of the stochastic modelling paradigm is model risk, i.e., the choice of model depends on stringent assumptions, which may not hold in reality, on the nature of the market phenomena being modelled. For example, assumptions such as the independence of returns and linear market impact functions lead to the naïve TWAP (placing equally sized orders over the execution) being the only optimal strategy, which is difficult to justify across all markets/assets in a real setting. Furthermore, the chosen functional forms of the stochastic processes used in models often lead to unavailable analytic solutions that only allow for approximations, costly parameter calibrations, and/or A Modular Framework for RL Optimal Execution complex optimization problems which can become computationally intractable and/or run into convergence challenges.

4. Proposed System

Brief introduction to Quantitative algorithmic trading.

- Quantitative algorithmic trading, also known as quant trading or quantitative trading, is a specific approach to algorithmic trading that emphasizes the use of quantitative analysis and mathematical models to guide trading decisions. It involves the application of statistical and mathematical techniques to identify trading opportunities, develop trading strategies, and automate the execution of trades.

- In quantitative algorithmic trading, traders rely on a systematic and data-driven approach to generate trading signals and make informed decisions. They typically employ a combination of historical data analysis, statistical modeling, and quantitative research to identify patterns, correlations, and anomalies in the financial markets.
- Key components of quantitative algorithmic trading include:
- Data Analysis: Traders collect and analyze vast amounts of historical and real-time market data, including price data, volume, market depth, and other relevant indicators. This data is used to identify patterns, trends, and relationships that can be exploited for trading opportunities.
- Quantitative Models: Quant traders develop mathematical models, often using statistical techniques and machine learning algorithms, to capture market dynamics and generate trading signals. These models may include trend-following models, mean reversion models, statistical arbitrage models, and volatility models, among others.
- Risk Management: Quantitative trading strategies incorporate risk management measures to control and mitigate potential losses. Position sizing, stop-loss orders, and portfolio diversification techniques are commonly used to manage risk.
- Backtesting and Optimization: Quant traders extensively backtest their strategies using historical data to evaluate their performance and optimize parameters. This process involves simulating trades based on past market conditions to assess profitability and risk characteristics.
- Execution Algorithms: Quant traders develop and implement algorithms for trade execution, aiming for efficient and timely order placement and execution. Execution algorithms can consider factors such as market impact, liquidity, and transaction costs to optimize trade execution.

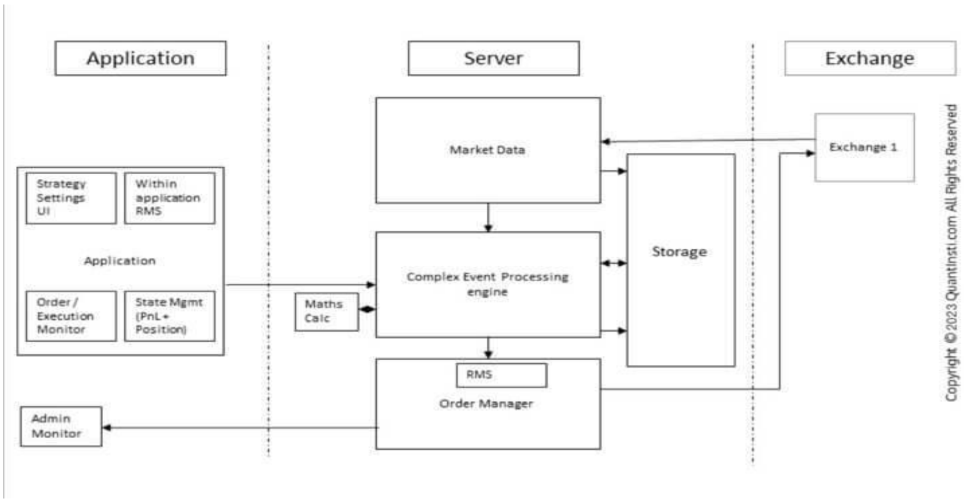

Figure 1Block diagram

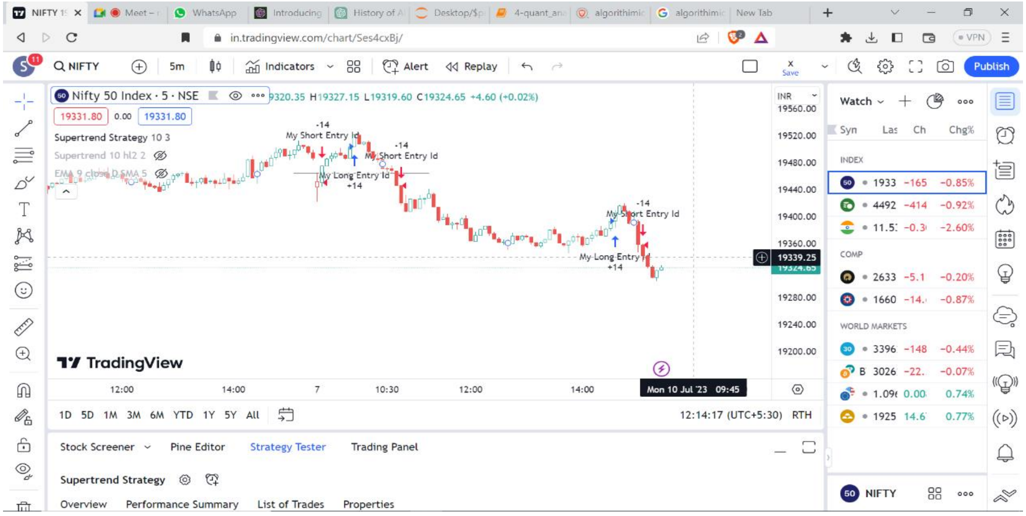

Figure 2 Generating Trade signal

If you want to go beyond stocks, mutual funds or bonds in your portfolio, buying options could be a good fit. Options trading gives you the right or obligation to buy or sell a specific security on a specific date at a specific price. An option is a contract that's linked to an underlying asset, e.g., a stock or another security.
Options contracts are good for a set period, which could be as short as a day or as long as a couple of years. When you buy an option, you have the right to trade the underlying asset, but you're not obligated to. If you decide to do so, that's called exercising the option. When you sell an option, you have an obligation to fulfill the contract. Selling options is where things get more complicated, and you could be at risk of losing an unlimited amount.

What is buying a put?

A put option is the opposite of a call option. Instead of having the right to buy an underlying security, a put option gives you the right to sell it at a fixed strike price (think of this as putting the underlying security away from you). Put options also have expiration dates. The same style rules (i.e., American or European) apply when you can exercise them.

Put options example

For example, say you buy a put option for 100 shares of ABC stock at $50 per share with a premium of $1 per share. Prior to the option's expiration date, the stock's price drops to $25 per share. If you exercise your option, you could still sell the 100 shares at the higher $50 per share price, and your profit would be $25 x 100 (less the $1 per share premium) for a total of $2,400.On the flip side, if the stock's price rises, you'll be out your premium, plus any commission.

What is buying a call?

A call option gives you the right to buy an underlying security at a designated price within a specific period (think of it as calling the underlying security to you.) The price you pay is called the strike price. The end date for exercising a call option is called the expiration date. Call options can be American-style or European-style. With American-style options, you can buy the underlying asset any time up to the expiration date. European-style options only allow you to buy the asset on the expiration date.

Call options example

For example, say you buy a call option for 100 shares of ABC stock with a premium of $3 per share, but you're hoping for a price increase this time. Your call option contract allows you to buy shares at $50 each. Meanwhile, the stock's price climbs to $100 apiece. You could effectively use a call option contract to buy that stock at a discount, saving yourself $4,700 ($50 x $100, minus the $3 per share premium).If the stock's price dropped and the option contract expired, you'd still be out the premium cost of $3 per share.

How to trade options in simple steps

Consider the following steps to trade options:

Open a trading account: Opening a brokerage account differs from opening an options trading account, especially if you plan to trade on margin(borrowing money from your broker to trade). Options trading brokers may want to see your investment objectives, trading experience, personal financial information and types of options to trade. Choose the options contract you'd like to trade: There's a huge variety you can choose from, so do your research on different strategies and stocks, make sure you're aware of all the disclosures and decide on the risks you're willing to take before choosing a path.

Select your strike price: Buying an option only works to your advantage if the stock price closes the option "in the money." The strike price refers to the price at which the underlying security can be bought or sold (exercised) in your options contract. Make your trade: Finally, pay the premium and broker commission and take ownership of the contract.

Figure 3 Parts of stock options quote

What is the advantage of trading options?

Cost-effective Trade

Options provide an investor with much flexibility in leveraged capital bets on the direction of a stock, irrespective of how it turns out. In this way, equities can earn profit even if they aren't performing well.

Hedging

Hedging in options trading means establishing a strategy to balance the risk of price swings in a future or equity position. Options can hedge Long-term stock investments at a reasonable cost.

Leveraged gains

Options give a chance to receive additional gains with little investment. It refers to creating much more significant potential from very little money.

What are the disadvantages of trading options?

Greater Commissions

Options provide more significant commissions according to Market direction prediction than a standard stock commission.

Complex Operations

For novice investors, Options and Strategies are convoluted.

Time Decay

Options run out of their time worthlessly. This is the resultant effect of the option's time-sensitive nature.
Time-sensitive nature
According to the market's prediction, there are higher chances of losing the entire investment ion since the contract is for a short period.

5. Working System

Fetching market data from data vendors

Fetching market data from data vendors is a common practice in algorithmic trading and quantitative analysis. Data vendors provide a wide range of financial data, including historical price data, real-time market data, fundamental data, news feeds, and more. Here's an overview of the process involved in fetching market data from data vendors:

Research and Select Data Vendor: Start by researching and selecting a reputable data vendor that offers the specific types of data you require. Consider factors such as data quality, reliability, coverage, pricing, and any specific requirements or preferences you may have.

Subscribe to the Data Service: Once you've chosen a data vendor, you typically need to subscribe to their data service. This involves signing up for an account, agreeing to their terms and conditions, and selecting the specific data packages or subscriptions that meet your needs. There may be different pricing plans based on the types and frequency of data you require.

Obtain Access Credentials: After subscribing to the data service, you will usually receive access credentials, such as API keys or login credentials, which allow you to authenticate and access the vendor's data.

Integration with API or Data Feed: Data vendors often provide APIs (Application Programming Interfaces) or data feed protocols that enable you to retrieve data programmatically. You will need to integrate your trading or analysis software with the vendor's API or data feed to fetch the desired market data.

Request and Retrieve Data: Once your software is integrated with the data vendor's API or data feed, you can start requesting and retrieving the market data you require. This may involve making API calls, sending specific data requests, or subscribing to real-time data streams, depending on the vendor's data delivery mechanisms.

1. **Data pre-processing**

   Data Cleaning:

   Handling Missing Values: Identifying and dealing with missing data points, which may involve imputation techniques such as mean, median, or regression-based imputation.
   Removing Duplicates: Identifying and removing duplicate records from the dataset to avoid

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

duplication biases.

Handling Outliers: Identifying and addressing extreme values or outliers that may significantly affect the analysis result.

2. **Quantitative analysis.**

Normal distribution, also known as the Gaussian distribution or bell curve, is a probability distribution that is symmetric around its mean. It is one of the most common and widely used probability distributions in statistics and quantitative analysis. The normal distribution is characterized by its bell-shaped curve, which is symmetrical and centered around the mean.

3. **Visualization**

Figure 4.10 Visualizing close price vs predicted price

As above observation close price is almost equal to the predicted price, and this gives insight that the probability of market closing near the predicted price is quite descent and hence adopting this strategy in real world market scenario
is like taking a calculated risk.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

4. **Writing pine script for the custom strategies**

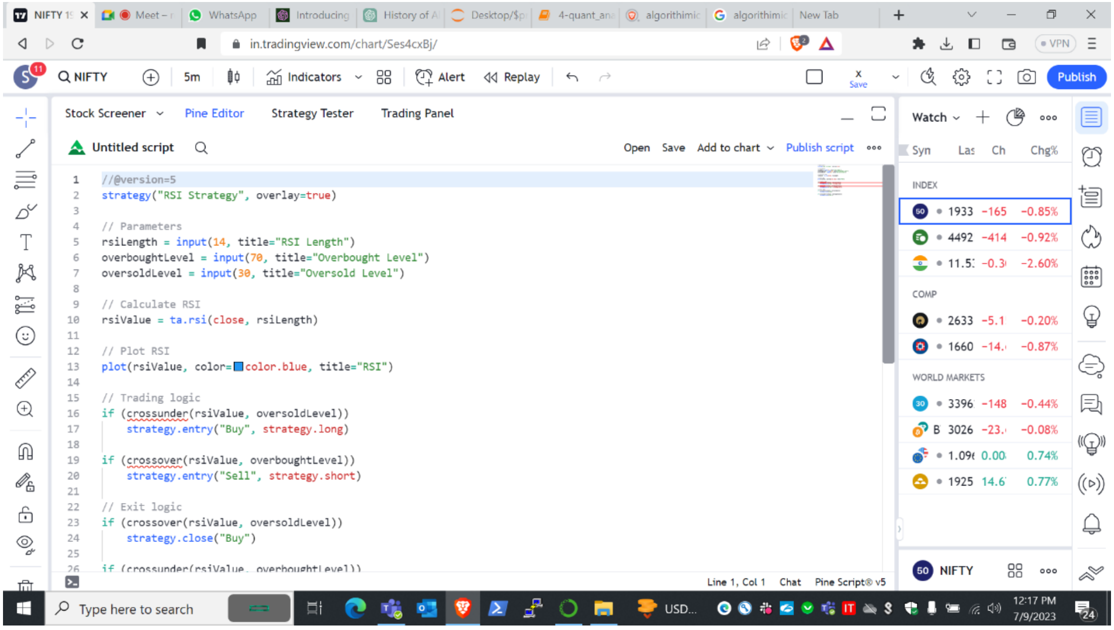

Figure 4.11 Pictorial representation of adding strategy in trading terminal

5. **Integrating strategies with broker terminal using API**

Steps involved in integrating strategy in broker terminal

Understand the Broker's API: Familiarize yourself with the broker's API documentation, which will outline the available functions, endpoints, and protocols for interacting with their trading platform.

Choose a Programming Language or Platform: Select a programming language or platform that is compatible with the broker's API. Common choices include Python, Java, C++, or trading platforms like MetaTrader or TradeStation.

Authenticate and Establish Connection: Set up authentication to access the broker's API. This typically involves generating API keys or tokens provided by the broker. Establish a connection between your trading strategy and the broker's server using the appropriate connectivity protocols (e.g., WebSocket, REST API).

Implement Trading Logic: Write the code for your trading strategy or algorithm, incorporating the necessary buy/sell rules, risk management, position sizing, and order placement logic. This code will interact with the broker's API to execute trades based on your strategy's signals or criteria.

Test and Debug: Before deploying your strategy to live trading, thoroughly test and debug your code in a simulated or paper trading environment provided by the broker. This allows you to validate the performance and behavior of your strategy without risking real capital.

Deploy and Monitor: Once you are satisfied with the testing results, deploy your trading strategy to live trading. Monitor the execution and performance of your strategy, making any necessary adjustments or optimizations as you observe its behavior in real-time market conditions.

Risk Management and Error Handling: Implement appropriate risk management measures, such as setting stop-loss orders or applying position limits, to manage potential losses. Handle errors and exceptions gracefully, ensuring that your trading strategy responds appropriately to unexpected situations or connectivity issues

6. Automating the trade

Steps involved in automating the trade:

- Define Your Trading Strategy: Clearly define the rules and parameters of your trading strategy. This includes determining the entry and exit criteria, risk management rules, position sizing, and any other relevant factors.

- Choose an Algo Trading Platform or API: Select an algo trading platform or broker that provides automation capabilities. Ensure the platform supports the markets and instruments you wish to trade. Alternatively, use a broker's API to connect your algorithm to their trading infrastructure.

- Develop or Code Your Algorithm: Implement your trading strategy in code. Use a suitable programming language such as Python, Java, or C++. Write the logic for order placement, position management, risk management, and any other necessary components of your algorithm.

- Connect to the Trading Platform or API: Establish a connection between your algorithm and the algo trading platform or broker's API. Set up authentication and obtain the required access credentials or API keys. This allows your algorithm to interact with the trading platform or broker's infrastructure.

- Test and Validate Your Algorithm: Backtest your algorithm using historical market data to assess its performance and validate its effectiveness. Ensure the algorithm generates expected results and meets your predefined criteria. Make adjustments and refinements as needed.

- Paper Trading or Simulation: Deploy your algorithm in a simulated or paper trading environment. This allows you to execute trades using virtual or simulated funds, replicating real market conditions without risking actual capital. Monitor and evaluate the algorithm's performance in this controlled environment.

- Risk Management and Error Handling: Implement risk management measures within your algorithm, such as setting stop-loss orders, position sizing limits, or other risk controls. Handle errors and exceptions gracefully to ensure your algorithm responds appropriately to unexpected situations.

- Deploying Live Trading: Once you have tested and refined your algorithm, deploy it for live trading with real capital. Ensure you have sufficient funds in your trading account to support your trading strategy. Monitor the performance of your algorithm in real-time market

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

conditions.

Financial machine learning research can loosely be divided into four streams. The first concerns asset price prediction where researchers

attempt to predict the future value of securities using a machine learning methodology. The second stream involves the prediction of hard or soft financial events like earnings surprises, regime changes, corporate defaults, and mergers and acquisitions. The third stream entails the prediction and/or estimation of values that are not directly related to the price of a security, such as future revenue, volatility, firm valuation, credit ratings, and factor quantiles. The fourth and last stream comprises the use of machine learning techniques to solve traditional optimization and simulation problems in finance like optimal execution, position sizing, and portfolio optimization. The first three streams are concerned with the creation of trading strategies,

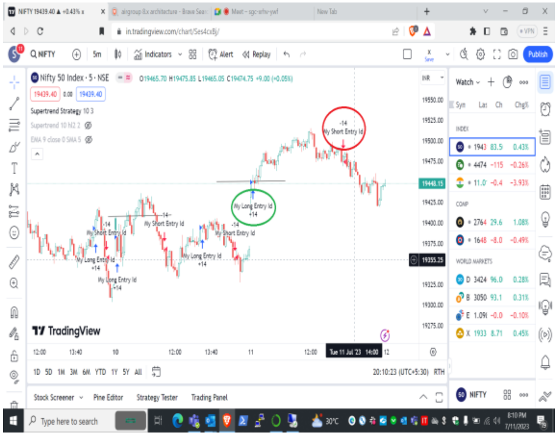

Figure 3 How trades are executed in live market

Figure 4 Profit visualization

6. Advantages

Automated option trading using AI offers a multitude of advantages. First, it enables traders to execute trades swiftly and efficiently, leveraging market opportunities in real-time. Second, AI algorithms make data-driven decisions, analyzing extensive market data to enhance trade execution and improve efficiency. By identifying patterns and trends, AI systems can provide valuable insights that might not be apparent to human traders.

AI-powered systems can be backtested using historical data, allowing traders to evaluate performance, optimize strategies, and gain valuable insights into potential risks and rewards. This process facilitates the development of robust and profitable trading strategies. Furthermore, automated trading systems can monitor the market 24/7, executing trades even when traders are not actively engaged. This ensures the capture of opportunities that arise outside regular trading hours, reducing the chance of missing out on profitable trades.

Scalability is another advantage of automated option trading using AI. AI algorithms can handle large volumes of trades and multiple strategies simultaneously, providing flexibility and diversification across various markets and instruments. The ability to adapt quickly to market changes is also an asset, allowing AI algorithms to adjust strategies in response to evolving market conditions.

AI-based systems offer improved speed and accuracy, reducing the likelihood of missed trading opportunities. They can process complex calculations and analyze multiple variables simultaneously, enhancing decision-making capabilities. Additionally, AI can uncover hidden insights and correlations in the market, enabling traders to identify profitable trading opportunities that might be overlooked by human traders.

Automated option trading saves time by reducing the need for manual intervention. AI algorithms can process information faster than human traders, leading to quicker trade execution. With their capacity to handle large data sets, AI systems excel at analyzing vast amounts of information to identify potential trades.

Finally, continuous evaluation and monitoring of AI-based trading strategies are crucial to ensure their effectiveness. By constantly monitoring performance, traders can make necessary adjustments and improvements, ultimately enhancing the reliability and profitability of their automated option trading strategies.

7. Conclusion

Technological adoption within portfolio management moves fast, and over the decades we have seen technologies come and go. It is likely that this cycle in quantitative finance will persist and that it also applies to machine learning in asset management, with one caveat: Machine learning is also practically revolutionary; instead of just maximizing alpha, it also minimizes overheard costs. Machine learning is already having large economic effects on many financial domains, and it is poised to grow further. Advanced machine-learning models present myriad advantages in flexibility, efficiency, and enhanced prediction quality.

We have paid special attention to how machine learning can be used to improve various types of trading strategies. We started by identifying important components to asset management in the context of machine learning, one of which is portfolio construction, which itself was divided into trading and weight optimization sections. The trading strategies were classified according to their respective machine learning frameworks.

8. Reference

[1] Hongyang Yang, Xiao-Yang Liu, Christina Dan Wang , *“FinGPT: Open-Source Financial Large Language Models”* ,Vol.03, DOI: 10.1011/s.2022.2.344

[2] Ruibo Chen, Wei Li, Zhiyuan Zhang, Ruihan Bao, Keiko Harimoto, Xu sun *“Stock Trading Volume Prediction with Dual-Process Meta-Learning”,* Vol.01, DOI: 10.1011/s.2022.2.344

[3] Tudor-Vlad Pricope*,”Deep Reinforcement Learning in Quantitative Algorithmic Trading: A Review”*,Vol.02, Page 357-90, 2021 DOI: 10.36648/jscp.2022.3.003

[4]V. Gupta and E. Kumar, *"Review on Machine Learning Techniques for International Trade Trends Prediction,"* 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 544-547, doi: 10.1109/ICAC3N53548.2021.9725585.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 07 | July - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

[5] Elior Nehemya, Yael Mathov, Asaf Shabtai, Yuval Elovici,*" Taking Over the Stock Market: Adversarial Perturbations Against Algorithmic Traders"*: Universal adversarial perturbations. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1765–1773 (2019).
[6] Market making and the changing structure of the securities industry. Lexington Books, Lexington, Mass, 1985.
[7] Nasdaq nordic algorithm offering. ttps://www.nasdaq.com/solutions/execution-algorithms. Accessed: 2022-05-18.
[8] Aurélien Alfonsi, Antje Fruth, and Alexander Schied. Optimal execution strategies in limit order books with general shape functions. arXiv e-prints, page arXiv:0708.1756, August 2007.
[9] Robert Almgren. Optimal execution with nonlinear impact functions and trading-enhanced risk. Applied Mathematical Finance, 10:1 – 18, 2003. [5] Robert Almgren. Optimal trading in a dynamic market. 2008.
[10] Robert Almgren and Neil Chriss. Optimal execution of portfolio transactions. Journal of Risk, pages 5–39, 2000. [7] Olle Dahlén Axel Rantil. Optimized trade execution with reinforcement learning. Master's thesis, Linköping University | Department of Computer and Information Science, 2018.
[11] Wenhang Bao and Xiao-yang Liu. Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis. arXiv e-prints, page arXiv:1906.11046, June 2019.
[12] Wenhang Bao and Xiao-yang Liu. Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis. arXiv e-prints, page arXiv:1906.11046, June 2019.
[13] Erhan Bayraktar and Michael Ludkovski. Liquidation in Limit Order Books with Controlled Intensity. arXiv e-prints, page arXiv:1105.0247, May 2011. [11] Dimitris Bertsimas and Andrew W. Lo. Optimal control of execution costs. Journal of Financial Markets, 1:1–50, 1998