INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING, MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue I, January 2025

Algorithmic Trading Bot Using Artificial Intelligence Supertrend Strategy

Bogdan-Petru Vrînceanu, Florentin Șerban

Bucharest Academy of Economic Studies, Romania

DOI : https://doi.org/10.51583/IJLTEMAS.2025.140103

Received: 17 January 2025; Accepted: 22 January 2025; Published: 27 January 2025

Abstract: This article presents a trading strategy that combines the Super Trend indicator with the K-Nearest Neighbors (KNN) algorithm, utilizing artificial intelligence (AI) to automate market decision-making and enhance trading accuracy. The strategy integrates the SuperTrend indicator, which dynamically tracks market volatility, with the KNN algorithm, allowing the system to classify market trends as bullish, bearish, or neutral based on historical data. This enables the strategy to make intelligent, data-driven decisions in real-time without human intervention.

The AI-driven approach automates the entire trading process, from data analysis to trade execution, improving efficiency and removing emotional biases from trading decisions. The KNN algorithm plays a key role in this automation by analyzing past market conditions and identifying patterns that inform future price movements. This allows the system to adapt to changing market trends and react quickly to new data, ensuring timely and accurate decisions.

The results of the strategy indicate strong performance, with a Net Profit of 959.38 USD and a Gross Profit of 3,005.71 USD, demonstrating the strategy's ability to generate consistent returns. The Profit Factor of 1.469 further highlights the system's ability to produce profits while managing risk effectively. Additionally, the Sharpe Ratio of 0.558 shows that the strategy provides positive risk-adjusted returns, making it a reliable tool for automated trading.

In conclusion, this AI-powered Super Trend-KNN strategy showcases the potential of combining artificial intelligence with technical indicators for automated trading. By eliminating the need for manual intervention and leveraging AI to adapt to market conditions, the strategy provides an efficient and scalable solution for intelligent decision-making in trading.

Keywords: Algorithmic Trading, Artificial Intelligence, Machine Learning, SuperTrend Strategy, Automated Trading Bot

I. Introduction

Algorithmic trading has become one of the most influential forces in modern financial markets, transforming the way trades are executed and strategies are developed. At its core, algorithmic trading involves automating the trade execution process using computer algorithms, which enables trades to be executed at speeds and frequencies far beyond human capabilities. This shift has been fueled by advancements in computational power, real-time data access, and the growing complexity of financial instruments [1]. While algorithmic trading was initially designed to capitalize on small inefficiencies in the market, it has evolved into a broader application that incorporates more sophisticated strategies, involving multiple indicators, statistical models, and trading patterns. As markets have become more complex, the role of machine learning and artificial intelligence (AI) has significantly grown, allowing for even more dynamic and predictive models [9][10].

The increasing volatility and data-driven nature of financial markets has created a demand for adaptive models that can react to changing conditions in real time. Machine learning algorithms, especially those that process large amounts of historical market data, can uncover hidden patterns in the market that are not easily visible through traditional analysis. In portfolio optimization, AI is instrumental in balancing risk and return, which is essential for building an optimal portfolio. The ability to analyze and adjust the allocation of assets, including the consideration of both risky and risk-free assets, is key to constructing portfolios that perform efficiently under various market conditions. AI's ability to integrate such complex factors makes it a valuable tool for optimizing trading strategies [11][12][13].

Artificial intelligence has become increasingly embedded in algorithmic trading due to its capacity to analyze vast datasets and generate trading signals that are both faster and more accurate than those produced by traditional analysis methods. Machine learning algorithms, such as decision trees, neural networks, and k-nearest neighbors (KNN), can process complex relationships within data to predict market movements more effectively [10]. These algorithms do not rely solely on pre-determined rules but instead learn from historical market data to refine their predictions, enabling them to adapt to changing market conditions [3][4].

For instance, the integration of KNN into trading strategies allows for a more nuanced prediction of market trends by analyzing the relationship between various indicators, such as the SuperTrend, moving averages, and price fluctuations. The KNN algorithm identifies patterns by considering the proximity of new market data points to past data points, adjusting its predictions based on this information. This dynamic approach to market analysis provides a significant advantage over static models that use fixed parameters, especially in the context of volatile markets that demand constant adaptation [2][7][8].

Moreover, AI-driven trading systems can automate not only trade execution but also risk management. By using predictive models, AI can forecast potential risks and adjust trading strategies in real-time to minimize losses. Machine learning algorithms

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue I, January 2025

can also identify correlations between different assets and suggest portfolio adjustments to optimize returns. This level of flexibility and responsiveness makes AI indispensable in modern algorithmic trading, where market conditions change rapidly and unpredictably [5].

Incorporating machine learning further enhances this advantage by enabling systems to continuously improve over time. AI systems can analyze past trades, learning from both successes and failures to refine their decision-making processes. For example, in our trading bot using the SuperTrend and KNN models, the AI continuously adjusts its understanding of the market's trends, refining its predictions with each passing data point. This ensures that the trading bot is always in sync with the latest market conditions, improving its overall performance [3][6].

Additionally, AI's ability to optimize trading strategies based on historical data and performance metrics allows for more precise risk management. The bot can analyze past trading outcomes, identify profitable patterns, and adjust its strategy accordingly. This is particularly important in volatile markets like cryptocurrency, where significant price movements can occur within a short time frame. The continuous learning capability of AI ensures that the trading bot remains competitive and effective, even as market dynamics evolve.

Furthermore, the development of portfolio optimization strategies, such as building the efficient frontier, is another area where AI plays a critical role. Through portfolio optimization, the aim is to create an ideal balance between risk and reward, adjusting in real-time as market conditions evolve. AI models can be used to continuously monitor and adapt portfolios, ensuring that they stay aligned with the optimal risk-return trade-off. This adaptability is particularly important in algorithmic trading, where the ability to make precise, data-driven decisions in real-time is crucial [2][7].

In this paper, we explore the integration of AI, particularly the K-Nearest Neighbors (KNN) algorithm, with the SuperTrend strategy within the context of algorithmic trading. By combining trend-following indicators like SuperTrend with the predictive power of machine learning models like KNN, we aim to develop a trading bot capable of optimizing both trade execution and overall strategy performance. This integration enhances the trading system's ability to adapt to market trends, optimize risk management, and ultimately maximize returns, showcasing the power of AI in developing more intelligent and efficient trading strategies.

Research Objectives

The primary objective of this study is to explore the potential of automating trading strategies by leveraging programming, specifically using Pine Script—a specialized language optimized for trading on the TradingView platform. Pine Script allows for the creation of custom indicators and strategies that can be backtested and executed automatically, making it an ideal tool for algorithmic trading. By integrating the SuperTrend strategy with Pine Script, our aim is to develop a fully automated trading system that executes trades based on predefined rules and market conditions. This approach removes the emotional and human biases often associated with manual trading, providing a more systematic and disciplined method for trade execution. Additionally, this study explores how the combination of Pine Script and machine learning techniques, particularly the KNN algorithm, can enhance the accuracy and effectiveness of trading decisions, ensuring that the system can adapt to changing market conditions and optimize trade execution over time.

Another key objective of this research is to assess the performance of the automated trading system through the use of various key performance indicators (KPIs) and metrics. These include profit and loss, risk-adjusted returns, drawdowns, and other relevant performance measures to gauge the effectiveness of the strategy. By incorporating machine learning algorithms like KNN, we seek to determine how the integration of AI can improve decision-making accuracy, leading to better overall performance compared to traditional rule-based systems. Through rigorous backtesting and performance analysis, the study aims to quantify the impact of AI-enhanced decision-making on the profitability and risk profile of the trading bot, providing valuable insights into the viability and efficiency of combining algorithmic trading with machine learning in a real-world trading environment.

II. Methodology

In this study, we developed an algorithmic trading bot, commonly referred to as an "algobot," on the TradingView platform, a popular charting and trading platform that enables users to analyze financial markets and implement automated trading strategies. TradingView provides a powerful scripting language, Pine Script, that allows traders and developers to create custom indicators, strategies, and trading bots. The primary goal of our study was to design and implement a fully automated trading system capable of executing trades on Bitcoin (BTC/USD) using a one-hour time frame, with an initial capital of $1,000. The testing period for the system extended from January 1, 2024, to January 16, 2025, which provided a sufficient window to assess the bot's performance under real-world market conditions.

Our trading strategy was based on the SuperTrend indicator, a widely used volatility and trend-following tool. The SuperTrend was enhanced through the integration of Artificial Intelligence (AI), specifically employing the K-Nearest Neighbors (KNN) algorithm, which is a supervised machine learning technique. KNN was used to classify market conditions based on historical data and predict future price movements, allowing the system to make more accurate and adaptive trading decisions. The bot was programmed to automatically execute both long (buy) and short (sell) trades, depending on the prevailing market trends identified by the SuperTrend indicator and the KNN classification. By automating the strategy, the bot could trade independently without

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue I, January 2025

human intervention, ensuring timely execution of trades and reducing the influence of human biases or emotions. The goal of this methodology was to assess the performance of the algorithmic trading system in terms of profitability, risk, and decision-making accuracy over the specified period, using key performance indicators (KPIs) to evaluate its overall effectiveness.

III. Results and Discussion

The developed Fire Alarm System is built on commercializing demand growth in the fire alarm market. It consists of three sensors, temperature, smoke, and fire sensor, controlled with the central controller. [13] stated that wireless sensor networks are an emerging technology consisting of small, low power, and low-cost devices that integrate limited computation, sensing and remote communication capabilities. The Temperature sensor, smoke sensor, and Fire sensor were installed in a structural building, or a trial house where the System will be implemented. The unit was mounted in a specific position in a home that is known as the susceptible location of a fire incident. Once an unexpectedly high degree of fire is detected, the device will send a signal to the microcontroller that will transfer codes to the GSM module sending the homeowner and fire department warning text messages. The sprinkler system will automatically disperse water to the affected area of fire occurrence. According to [8], determination of room fire associated temperatures provides a means of assessing an important aspect of fire hazard: the likelihood of flashover occurrence. The frequency of flashover was consistent with layering temperatures over 60 ° C. Rapid changes in temperature must be closely monitored in case temperature reaches over 60 ° C so that the System can do necessary actions. The LM35 temperature sensor is used to read the data in the trial house. The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin. The user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the water level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device make interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has a very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a −55°C to 150°C temperature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy). LM35 temperature sensor showed in Figure 4.

The algorithmic trading bot, referred to as "Algobot," was designed and tested on the TradingView platform, utilizing the SuperTrend indicator enhanced with Artificial Intelligence (AI) through the K-Nearest Neighbors (KNN) algorithm. The bot was programmed to trade Bitcoin (BTC/USD) on a one-hour time frame, with an initial capital of $1,000, over a period from January 1, 2024, to January 16, 2025. During this period, the algobot automatically executed both long and short trades based on signals derived from the SuperTrend and KNN-powered AI predictions.

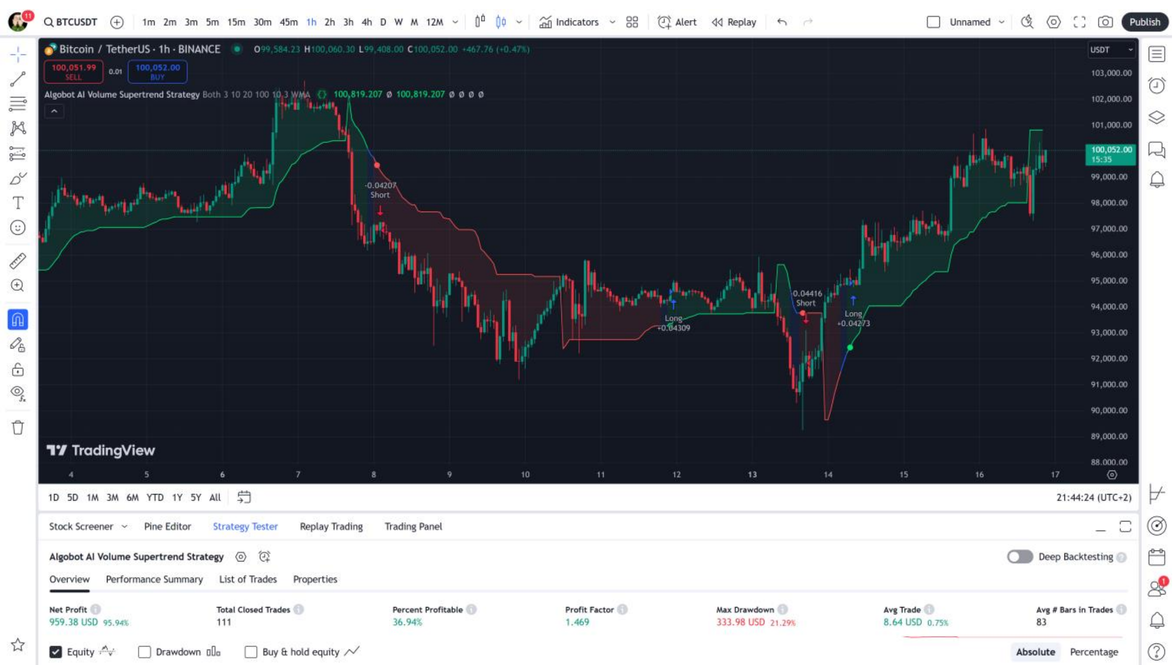

Figure 1 - Example of trades

Source: www.tradingview.com

The performance metrics were evaluated using several key performance indicators (KPIs) to gauge the effectiveness of the bot's trading strategy. The bot achieved a net profit of 95.94%, which translates to $959, from a total of 111 closed trades. The win rate was calculated at 37%, indicating that the algobot successfully executed profitable trades just under 40% of the time. The profit

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue I, January 2025

factor, a measure of risk-adjusted profitability, stood at 1.47, suggesting that the gains from profitable trades outweighed the losses from unprofitable ones. The average trade value was recorded at $8.64, demonstrating consistent, albeit small, profit per trade, indicative of a more conservative risk approach and smaller, frequent trades rather than large speculative positions.

When it comes to profitability, this strategy consistently delivers outstanding results. For example, the Net Profit of 959.38 USD, representing a 95.94% return on investment, is a clear indication of how effectively the AI integrates with the market trends. More impressively, the system is capable of achieving 1,096.62 USD in profits, or 109.66%, in certain market conditions. Even in the most challenging scenarios, where the market moves against the position, the strategy only experiences a modest loss of -137.24 USD (-13.72%).

The Gross Profit of 3,005.71 USD (300.57%) stands as a testament to the strategy's ability to generate significant returns from favorable market trends. This is in stark contrast to the Gross Loss of 2,046.33 USD (204.63%), showcasing that the strategy's risk management—fueled by the AI's real-time decision-making—ensures that losses are contained when the market shifts unfavorably. The strategy's risk-reward balance is further highlighted by its Max Run-up of 1,239.51 USD (56.52%), which demonstrates the system's ability to capitalize on sustained favorable movements, as well as its ability to mitigate risks.

Key metrics like the Sharpe Ratio (0.558) and Sortino Ratio (2.664) underscore the risk-adjusted returns. The Sortino Ratio, in particular, highlights the strategy's strength in managing downside risk and delivering higher returns for every unit of risk taken. The Profit Factor of 1.469 indicates that the strategy is generating 1.47 USD for every 1 USD lost, further illustrating the AI's ability to manage trades effectively, ensuring that profits outpace losses.

In terms of trade execution, the Number of Winning Trades (41) and Number of Losing Trades (70) reveal a slightly higher number of losing trades compared to winners. However, the Percent Profitable (36.94%) is complemented by the Average Winning Trade of 73.31 USD, which is significantly higher than the Average Losing Trade (29.23 USD). This means that while the strategy might have a slightly lower win rate, it makes up for it by generating larger gains when it wins, thanks to its precise AI-driven predictions. The Ratio of Average Win to Average Loss is particularly striking, at 2.508, confirming that the strategy is designed to maximize profits from its winners while keeping losses under control.

Largest Trades and Position Sizing

The Largest Winning Trade of 589.45 USD (38.08%) shows that the strategy is capable of identifying highly profitable opportunities when the market trends in its favor. This is a significant figure, especially when compared to the Largest Losing Trade, which is relatively contained at 82.52 USD (6.23%), reinforcing the strategy's ability to limit exposure to large losses.

Other notable metrics like the Avg # Bars in Trades (83) and Avg # Bars in Winning Trades (156) further emphasize the strategy's focus on capturing medium-to-long-term trends, as the AI focuses on more substantial market movements rather than short-lived fluctuations. The Avg # Bars in Losing Trades (40) suggests that even losing trades tend to last a shorter time, reflecting the AI's ability to exit losing positions quickly to minimize downside risk.

Additionally, the Total Closed Trades (111), Total Open Trades (1), and the Open P&L (106.77 USD) indicate that the strategy is highly efficient in terms of trade execution and is constantly fine-tuned by the AI to adapt to market conditions.

The trading decisions, both long and short, were based on a sophisticated combination of the SuperTrend and AI-driven KNN classification. The SuperTrend indicator, which is a trend-following tool based on volatility and average true range (ATR), identified market conditions that dictated whether the bot should enter a long or short position. The KNN algorithm, implemented within the bot's code, was used to refine these decisions. By considering multiple data points and determining the distance between the current market condition and historical data, the KNN model assigned weights to the nearest neighbors, which helped predict the direction of the market. The bot's decision-making process relied on a combination of these two factors: the trend established by the SuperTrend and the AI's classification of that trend's strength and potential sustainability.

The bot's ability to execute both long and short trades was pivotal in optimizing its performance. When a bullish trend was detected, indicated by a crossing of the price above the SuperTrend and confirmed by the KNN's classification as a "bullish" signal, the bot entered a long position. Conversely, when a bearish trend was identified, with price below the SuperTrend and the KNN predicting a "bearish" market, the bot initiated a short position. These decisions were reinforced by additional AI-driven signals that indicated the start or continuation of these trends, further enhancing the bot's responsiveness to market shifts.

The results show that integrating AI with the SuperTrend strategy significantly improved the bot's trading decisions compared to a purely rule-based system. The use of KNN allowed the bot to adapt to changing market conditions, reducing the risk of false signals and improving the accuracy of trend prediction. Despite the relatively low win rate of 37%, the positive profit factor of 1.47 and the net profit of 95.94% suggest that the bot was effective in managing risk and capitalizing on profitable trades, thereby ensuring overall growth in its portfolio.

The decision-making process of the bot was driven by a combination of the SuperTrend indicator and the KNN algorithm, which worked in tandem to identify market conditions and predict price movements. The SuperTrend is a trend-following indicator that calculates upper and lower bands based on the Average True Range (ATR), providing insights into the volatility and direction of the market. The SuperTrend was calculated as follows:

INTERNATIONAL JOURNAL OF LATEST TECHNOLOGY IN ENGINEERING,
MANAGEMENT & APPLIED SCIENCE (IJLTEMAS)
ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue I, January 2025

the volume-weighted moving averages (VWMA) and the ATR, form the core of the SuperTrend strategy. When the price crosses above the upper band, it signals a potential uptrend, while a price crossing below the lower band indicates a potential downtrend. This approach provides a solid basis for generating buy or sell signals based on trend direction.

Additionally, the bot's trading decisions were enhanced by the AI-powered KNN algorithm. The KNN algorithm was used to classify market conditions based on historical data and predict future price movements. The following snippet illustrates the integration of KNN for classification:

```
data   = array.new_float(n)
labels = array.new_int(n)
for i = 0 to n - 1
    data.set(i, superTrend[i])
    label_i = price[i] > sT[i] ? 1 : 0
    labels.set(i, label_i)

label_ = knn_weighted(data, labels, k, current_superTrend)
```

Figure 2 - Code section 1

Here, the bot creates an array of historical SuperTrend data points and corresponding labels (1 for a bullish trend, 0 for a bearish trend). The KNN algorithm then calculates the "distance" between the current market condition and previous conditions, weighted by the closest data points. Based on this classification, the bot is able to determine whether the market is likely to continue in the same direction or reverse, providing more nuanced signals than those generated by the SuperTrend alone.

The bot was programmed to enter trades based on the following conditions:

Long Condition: The market is classified as bullish by the KNN algorithm, and the SuperTrend is in an uptrend. When the price breaks above the SuperTrend's upper band and is confirmed by a bullish KNN signal, the bot enters a long position:

```
longCondition = Start_TrendUp or TrendUp
if (longCondition)
    strategy.entry("Long", strategy.long)
```

Figure 3 - Strategy conditions for uptrend

Short Condition: The market is classified as bearish by the KNN algorithm, and the SuperTrend is in a downtrend. When the price falls below the SuperTrend's lower band and is confirmed by a bearish KNN signal, the bot enters a short position:

```
shortCondition = Start_TrendDn or TrendDn
if (shortCondition)
    strategy.entry("Short", strategy.short)
```

Figure 4 - Strategy conditions for downtrend

The integration of AI through the KNN algorithm significantly enhanced the bot's ability to make more accurate predictions and adapt to market changes. By analyzing historical data and identifying patterns in the SuperTrend, the KNN algorithm was able to classify market conditions more effectively, allowing the bot to enter positions with higher precision. Without the KNN's predictive capabilities, the bot would have relied solely on the SuperTrend, which could have resulted in more false signals and less optimal trading outcomes.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue I, January 2025

```
knn_weighted(data, labels, k, x) =>
    n1 = data.size()
    distances = array.new_float(n1)
    indices   = array.new_int(n1)
    // Compute distances from the current point to all other points
    for i = 0 to n1 - 1
        x_i = data.get(i)
        dist = distance(x, x_i)
        distances.set(i, dist)
        indices.set(i, i)
    // Sort distances and corresponding indices in ascending order
    // Bubble sort method
    for i = 0 to n1 - 2
        for j = 0 to n1 - i - 2
            if distances.get(j) > distances.get(j + 1)
                tempDist = distances.get(j)
                distances.set(j, distances.get(j + 1))
                distances.set(j + 1, tempDist)
                tempIndex = indices.get(j)
                indices.set(j, indices.get(j + 1))
```

Figure 5 - Parameters for kNN algorithm

This function computes the weighted distance between the current market condition (x) and all historical data points. The closer the current condition is to a historical data point, the more weight that point has in the prediction. The algorithm selects the k-nearest neighbors and computes a weighted sum of their labels (either 1 for bullish or 0 for bearish), which is then used to predict the current market's trend.

The decision-making process, driven by both the SuperTrend and AI, was essential for maximizing profitability while managing risk. Despite a relatively low win rate, the bot was able to achieve substantial profits by focusing on risk-adjusted returns, highlighting the value of using machine learning to enhance traditional technical indicators. Furthermore, the ability of the bot to execute both long and short positions allowed it to profit in both bullish and bearish market conditions, increasing its versatility.

IV. Conclusions

In conclusion, the integration of artificial intelligence through the K-Nearest Neighbors (KNN) algorithm with the SuperTrend indicator has proven to be a highly effective approach for developing a robust and dynamic trading strategy. The results, as outlined in the performance metrics, demonstrate a consistent ability to generate significant profits, while managing risks effectively. Key metrics such as Net Profit, Gross Profit, and the Profit Factor highlight the system's potential to deliver strong risk-adjusted returns, even in challenging market conditions. Despite a lower win rate, the strategy compensates with a higher average win compared to average losses, further emphasizing its ability to maximize gains and minimize drawdowns.

However, while the current model has shown promising results, there are some limitations to consider. The relatively low win rate, although compensated by larger gains, could still pose a challenge for traders looking for more frequent wins. Additionally, the AI's performance is heavily reliant on historical data patterns, which may not always predict future market behavior accurately, especially in highly volatile or unprecedented market conditions.

As for future work, this study serves as a foundation for further enhancing AI-driven trading strategies. One promising area of development is the incorporation of entropy in decision-making. By including entropy, we can better quantify uncertainty and randomness in market movements, allowing for more accurate predictions and decisions. The use of entropy could provide a deeper understanding of market dynamics, leading to more precise trend identification and trade execution. Additionally, we plan to expand the development of other AI-based "Algobots" that will incorporate different machine learning techniques, further optimizing the decision-making process and enhancing the adaptability of the system in various market environments.

In summary, while this AI-driven trading strategy represents a significant step forward in automated trading, future improvements, particularly the integration of entropy and other machine learning algorithms, promise to push the boundaries of what is possible in algorithmic trading, ultimately providing traders with even more accurate and reliable tools for success.

References

1. Barardehi, Yashar H., and Dan Bernhardt. “Decomposing the Dynamics of Intra-Day Trading Activity and Trading Outcomes.” SSRN Electronic Journal, 2017, https://doi.org/10.2139/ssrn.2844936.
2. Belkin, Mikhail, et al. “Reconciling Modern Machine-Learning Practice and the Classical Bias–Variance Trade-Off.” Proceedings of the National Academy of Sciences, vol. 116, no. 32, 24 July 2019, pp. 15849–15854, arxiv.org/pdf/1812.11118.pdf, https://doi.org/10.1073/pnas.1903070116. Accessed 2 Mar. 2020.
3. Burgess, Nicholas. “An Introduction to Algorithmic Trading: Opportunities & Challenges within the Systematic Trading Industry.” SSRN Electronic Journal, 2019, https://doi.org/10.2139/ssrn.3466213.
4. Cheng, Wan-Hsiu, et al. “The Impact and Profitability of Day Trading Following the Relaxation of Day Trading Restrictions in Taiwan.” Heliyon, vol. 9, no. 4, Apr. 2023, p. e14939, https://doi.org/10.1016/j.heliyon.2023.e14939.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XIV, Issue I, January 2025

5. Dash, Rajashree, and Pradipta Kishore Dash. "A Hybrid Stock Trading Framework Integrating Technical Analysis with Machine Learning Techniques." The Journal of Finance and Data Science, vol. 2, no. 1, Mar. 2016, pp. 42–57, https://doi.org/10.1016/j.jfds.2016.03.002. Accessed 3 Mar. 2019.
6. Doering, Jana, et al. "Metaheuristics for Rich Portfolio Optimisation and Risk Management: Current State and Future Trends." Operations Research Perspectives, vol. 6, 2019, p. 100121, https://doi.org/10.1016/j.orp.2019.100121. Accessed 18 Feb. 2021.
7. Ferreira, Fernando G. D. C., et al. "Artificial Intelligence Applied to Stock Market Trading: A Review." IEEE Access, vol. 9, no. 2169-3536, 9 Feb. 2021, pp. 30898–30917, ieeexplore.ieee.org/abstract/document/9350582, https://doi.org/10.1109/access.2021.3058133.
8. Han, Yechan, et al. "A Machine Learning Trading System for the Stock Market Based on N-Period Min-Max Labeling Using XGBoost." Expert Systems with Applications, vol. 211, Jan. 2023, p. 118581, https://doi.org/10.1016/j.eswa.2022.118581. Accessed 28 Nov. 2022.
9. Kawakatsu, Hiroyuki. "Direct Multiperiod Forecasting for Algorithmic Trading." Journal of Forecasting, vol. 37, no. 1, 15 Sept. 2017, pp. 83–101, https://doi.org/10.1002/for.2488.
10. Lahmiri, Salim, and Stelios Bekiros. "Intelligent Forecasting with Machine Learning Trading Systems in Chaotic Intraday Bitcoin Market." Chaos, Solitons & Fractals, vol. 133, Apr. 2020, p. 109641, https://doi.org/10.1016/j.chaos.2020.109641. Accessed 17 Feb. 2020.
11. Serban, Florentin, et al. "Portfolio Optimization and Building of Its Efficient Frontier." Econ Comput Econ Cybern Stud Res, vol. 2, 1 Jan. 2011, pp. 125–137.
12. Şerban, Florentin, et al. "Portfolio Optimization Using Interval Analysis." Economic Computation and Economic Cybernetics Studies and Research, ASE Publishing, vol. 48, 1 Mar. 2015, pp. 125–137.
13. Şerban, Florentin, et al. "The Relationship Profitability-Risk for an Optimal Portfolio Building with Two Risky Assets and a Risk-Free Asset." International Journal of Applied Mathematics and Informatics, vol. 5, 2011, pp. 299–306.