# Trading Auto-bot for Enhanced Financial Decision Making


Sumana Sinha
*Department Of Information Science And Engineering*
*Nitte Meenakshi Institute Of Technology*
Bengaluru, Karnataka
sumana.sinha@nmit.ac.in

Aaron Mackenzie
*Department Of Information Science And Engineering*
*Nitte Meenakshi Institute Of Technology*
Bengaluru, Karnataka
ammisquith@gmail.com

Ullal Akshatha Nayak
*Department Of Information Science And Engineering*
*Nitte Meenakshi Institute Of Technology*
Bengaluru, Karnataka
akshatha.n@nmit.ac.in

Ashwin Sridhar
*Department Of Information Science And Engineering*
*Nitte Meenakshi Institute of Technology*
Bengaluru, Karnataka
ashwinsridhar777@gmail.com



***Abstract*—This paper develops an automated trading system based on the backtesting of strategies on the Trader's View platform, which simulates multiple market scenarios. Machine learning models KNN and KNN with PSO is used for backtesting to measure the performance based on risk-adjusted returns, drawdown, and profit consistency. These selections were also incorporated into the Zerodha API using the KNN-PSO model, which can connect with real markets for the automated execution of trades and showed a considerable outperformance in return on investment and risk management. The model in real time was greatly comparable with what had been in the back-tested result, thus evolving accordingly with changing market conditions. This work is promising specially in terms of short term trading for the improvement of profitability and decision-making as an insight for individual traders and institutions and proved by comparing with LSTM and KNN-PSO ensemble model.**



***Index Terms*—K nearest neighbor (KNN), Zerodha API, Particle swarm optimization (PSO), Long Short-term memory (LSTM), Algorithmic Trading, Automated Trading Systems, Kite connect.**


## I. Introduction

A trading auto-bot, also known as an automated trading system or algorithmic trading bot, is a sophisticated software program designed to execute transactions in financial markets without human intervention. By following predefined rules and algorithms, these bots make trading decisions based on real-time market data, technical indicators, and trading strategies. This automated approach offers several key advantages over conventional manual trading methods, particularly in terms of speed, efficiency, and accuracy.The primary advantage of a trading auto-bot is its ability to execute trades at high speed by rapidly analyzing vast amounts of market data. This reduces the likelihood of missing profitable opportunities, as the system can place trades within fractions of a second. Auto-bots are also capable of scanning multiple financial instruments and time frames simultaneously, allowing them to monitor market conditions in real-time and identify specific patterns or signals that align with predetermined strategies. Another significant benefit is the removal of emotional biases from the decision-making process. Human traders can be influenced by emotions such as fear, greed, or hesitation, leading to inconsistent and potentially harmful trading decisions. In contrast, auto-bots operate strictly according to set algorithms, ensuring objective, consistent, and emotion-free trading execution. This results in improved decision-making based solely on data and strategy, rather than human impulses. In addition to executing trades, auto-bots are programmed to manage open positions using predefined risk management rules, such as take-profit and stop-loss levels. This automated risk management helps to protect trades and reduce potential losses. As a result, trading auto-bots enhance objectivity, consistency, and efficiency in trading activities. However, it is essential to thoroughly backtest and optimize these bots before deploying them in live markets. Ongoing monitoring and periodic adjustments are also crucial to ensure that the auto-bot adapts to changing market conditions and continues to perform effectively over time.

## II. Objectives

In this work we have used the Zerodha kite connet API get access of historical and real-time market data use KNN and KNN PSO Model to automate the trading system in real-time for discovering market opportunities. The objectives to be achieved are listed below:

- **Timely and Efficient Execution:** Rapid and timely execution of trades based on the conditions of the markets to eliminate delays and maximize opportunities.
- **Elimination of Human Error and Emotional Bias:** o reduce errors which may happen due biased and emotional decision biases like fear or greed by implementing

pre-programmed algorithms, thereby trading with consistency and without subjectivity.
- **Successful Risk Management:** Putting in place robust risk management strategies, which include the use of risk-reward ratios, position sizing, stop-loss orders, and other risk-remediation techniques that will protect your capital and exposure.
- **Scalability:** Autobot is able to maximize profit potential and reduce dependence on any one market thus achieving scalability, and robustness through trading a variety of asset classes.
- **Adaptability to Transition of Market:** Helps the automation system identify changing market conditions and switches trading strategies that may change over time.
- **Extended Performance Measurement:** The performance is measured by tracking the outcomes of trading; monitors key performance indicators such as returns, drawdowns, risk-adjusted returns, and hence affords to continuously improve the system.

The rest of the paper consist of the literature survey in III section, section IV talks about methodology followed Result section section V and the conclusion section VI.

## III. LITERATURE SURVEY

Algo-trading eliminates the influence of human emotions on trading activities, which makes trading more systematic and market liquidity in addition to providing opportunities for profit for the trader and in this section some of the research works are discussed. The paper by Vaibhav Aggarwal et al. [1], they talks about how trading with algorithms has changed the traditional way of monitoring the financial markets and making decisions, both for investors and researchers. It has also established connections with high-speed data processing strategies in the financial field, resulting in various applications. They included high-frequency trading (HFT), arbitrage, ML, and AI, to mention a few of these systems that have contributed to efficiency and liquidity within markets. However, there are also some drawbacks.

The research work by M. E. Aloud et al. [2] leverages prior research in algorithmic trading, specifically reinforcement learning and modeling of the events of directional change. RL previously has been applied to stock trading, while few works tried to incorporate dynamic thresholds in DC for market state representation. The current study enhances the RL-based trading methodologies with advanced state representation and decision-making for less volatile returns. This paper [3] is based presents the study of traditional approaches, such as TTRs or machine learning models, predict stock trends by static strategies and introduced a more advanced reinforcement learning algorithms for decision-making by dynamically adapting to the market conditions using deep Q-networks. This paper enhances the models by incorporating both past and future stock trends, including a Gated Recurrent Unit to make more accurate trading decisions.

This work by Salvatore Carta1 et al. [4], come up with a multi-layer and multi-ensemble method for deep learning and deep reinforcement learning-based stock trading. The suggested approach optimizes trading decisions using hundreds of convolutional neural networks (CNNs) and a reinforcement learning (RL) meta-learner. Their model performs better than the existing machine learning-based approaches and traditional approaches, by considering the terms of return on investment and risk management, and experimental findings on real-world data. The results underline the profitability and resilience of combining deep learning approaches with technical analysis. Authors have improved the model performance even further, by enabling the model to predict future studies to investigate the effects of adding whole financial news content and hyper-parameter optimization. Thus providing insightful analysis of how cutting-edge machine learning techniques are applied in financial trading.

[12] The authors also use intensive experimentation to illustrate how looking at the combination of some technical indicators together with market returns may improve trading strategy performance. Testing against real and simulated market data has been emphasized as a very important step throughout this article. This is to ascertain the reliability and flexibility of the trading strategy to adapt to the fluctuating financial markets. The research contributes to the ongoing evolution of the financial markets by bringing out the automation of the trading decisions and generation-testing of the automated trading strategies. It gives viable information to the academics and traders who would like to have an edge in trading by using technology and artificial intelligence. The paper by Yuxiang Huan at el., [7] highlights the difficulties these robots encounter in actual markets as it explores the creation and testing of autonomous traders in the sector of finance. It offers a way to evaluate FX trading methods and give reasonable expectations about how well the strategies will perform. The study highlights the challenge of developing trustworthy trading methods and makes the argument that many strategies that have been published are unreliable means of making investments. The suggested approach can be used to choose possible robots and set minimum timeframes and specifications for test runs. The article also discusses the variety of AI methods applied in the banking industry, ranging from straightforward concepts to intricate machine learning plans. Overall, the study sheds light on the difficulties in creating trustworthy autonomous trading strategies, the significance of practical testing procedures, and the variety of AI approaches applied in the finance industry. The results of this study can help autonomous traders progress and enhance their effectiveness in actual market settings. The paper [14] delves into the fascinating realm of automated trading systems and agent-based technology, exploring their potential to revolutionize financial markets. The authors highlight the advantages of electronic trading systems and autonomous trading agents, including lower costs, faster execution, and the elimination of human intermediaries, leading to increased efficiency and profitability. The paper by Medha Mathur et al. elaborates on an Autobot algorithmic trading bot that supports an automated trading system using Random Forest a machine-

learning model. The Autobot has been designed for short term trading by checking the market conditions and performing trades efficiently, thereby cutting down on manual intervention and transaction costs. Here, financial knowledge is combined with machine learning to enhance profitability and trading performance. Backtesting of the various strategies showed that the bot could make proper decisions. Thus it would be of great benefit to traders it in making automated and optimized trade executions [9]. In this per we have referred papers based on algorithmic trading [11], [10], [15].

### A. Limitations of existing models

The literature survey gives us insight into the areas to be taken care of.

- As algorithmic trading is concerned achieving stability is a major issue. To guarantee an exhaustive evaluation a framework is needed that monitors market circumstances from various data sources and reliably gauges trading bot performance.
- Overfitting of the historical data is a major issue in most cases the algorithms may show great performance in the training and testing stages, but usually decrease when faced with fresh market conditions or previously unheard-of data.
- Managing intricacies with scalability is a challenge while managing massive data and transaction volumes. The complexity of the algorithmic trading model is increased by an amalgamation of many data sources, such as news, social, and financial data.
- The self-governing trading algorithms spawn ethical and regulatory issues, namely market manipulation, insider trading, and adherence to financial regulations. Maintaining market integrity requires trading algorithms to stay within moral and legal bounds.

## IV. METHODOLOGY

The general methodology for building this model describes how the KNN and KNN with PSO can be incorporated with Zeroda API to build an algorithmic trading system to automate real-time trading with risk management which involves requirements gathering, system design, implementation, integration, and maintenance.

### A. Collection of Data and Feature Engineering

The Zerodha API access is checked before for utilizing trading functions. Additionally, it evaluated to check whether trading strategies should be tested and anticipated using a backtesting platform like TradingView. Specific performance metrics and trade data were identified to assess the efficacy of various trading strategies.

### B. Model Design and Training

The real-time data through Zerodha Kite Connect is fed to the KNN and KNN with PSO and trained based on historical data. Splitting of the data set for training and test sets is done in 80 20 ratio. The KNN-PSO model is then trained using the historical training data sets and the optimal number of neighbors k is tuned by using techniques such as cross-validation or Particle Swarm Optimization (PSO). The evaluation tests the trained model on the test set to ascertain its accuracy and ability to predict price movements in real-time. The KNN-PSO is compared with an ensembled model consisting of KNN-PSD and Long short-term memory LSTM [8].

### C. Risk Management Integration

- Implementation of Stop-Loss and Trailing Stops:As long as the KNN model predicts some movement in stock price, then stop loss automatically gets generated and placed via Zerodha API. Stop-Loss: A stop-loss order is placed at a predefined percentage (e.g., 2%) below the entry price to limit potential losses in case the market moves unfavorably.
  Trailing Stop: As the stock price moves in a favorable direction-say up-the trailing stop automatically adjusts downward to lock in profits while limiting downside risk.
- Position Sizing Based on Risk: Position size for each trade is calculated based on trader's risk tolerance and the stop-loss distance and is given as:
  $$PositionSize = \frac{AccountBalance * RiskperTrade}{Stop - LossDistance} \quad (1)$$
  The calculated position size1 is then forwarded to the Zerodha API for the buy or sell orders. The result is such that the portfolio stays within the risk management rules, not through building up positions in highly volatile times.
- Tracking Maximum Drawdown This is to say that the system tracks continuously the maximum drawdown, which is the maximum loss seen from peak to trough of value of a portfolio. If the drawdown crosses a previously set threshold value, say 10%, the system automatically exits all positions to prevent losing any more money. The Zerodha API executes sell orders in this scenario. The drawdown (DW) is given as 2:
  $$DW = \frac{PPV - TPV}{PPV}; \quad (2)$$
  where PPV is Peak portfolio value and TPV is Trough Portfolio value The maximum drawdown is calculated after every trading execution.

### D. Automatic Trading Execution

- **Real-time market data access** Fetch real-time stock market data is fetched through the Zerodha API, which includes the current price, volume, and order book depth for the selected stocks. The KNN-PSO model produces predictions regarding the stock price going up or down, based on real-time features such as Open, High, Low, and Close.

- **Buy Orders and Order Tracking** With the output of the KNN model comes the prediction, which the system automatically sets as a buysell order through the Zerodha API. For a potential upward move, a buy order is issued with a stop-loss order. In case there is a downtrend, sell is made. Trailing stops move dynamically with a price that is moving favorably and protect profits at those levels.

*1) Performance Monitoring and Metrics Calculation:*

- Risk-adjusted return is calculated using the Sharpe Ratio for the trading strategy.

$$SharpeRatio = \frac{MDR - RFR}{SDDR}; \tag{3}$$

where MDR is MEAN daily return , RFR is Risk free rate, and SDDR is standard deviation of daily return. The high Sharpe Ratio found by using the equation 3 means that, per unit of risk, the strategy offered higher returns.
- **Profitability Consistency** Profit Consistency is the percentage of profitable days or trades traded over the testing period. The system tracks how often KNN models make successful calls in relation to the total number of trades executed. A high percentage of profitable trades means stability and consistency.

*2) Backtest and Refine the Strategy:*

- **Backtesting Using Historical Data** The performance of this KNN model, regarding its trading strategy, is discussed using historical data for all different market conditions. The backtesting phase calculates metrics like accuracy, drawdown, and risk-adjusted returns in order to optimize the model.
- **Refinement of Risk Parameters** With backtesting results, all the risk parameters of the system can be optimized. Examples include the stop-loss threshold, position sizing, and drawdown limits. Another model which can be further tuned for optimizing prediction accuracy is the KNN algorithm, making use of hyperparameter optimization techniques like Particle Swarm Optimization(PSO) [5], [13].

*3) Deployment and Monitoring:*

- **In-host Deployment** Once the strategy has been backtested and optimized, it gets run on a live trading environment via the Zerodha API. In the system, KNN-PSO actually makes real-time decisions to trade, and thus trades are executed directly. Besides, risk is managed through stop-loss orders, trailing stops, and monitoring of draw-down.
- **Continued Observation** It continuously tracks the market, trade execution, and portfolio performance to have those predefined risk parameters adapted, while keeping the KNN model viable under the broad unfolding of these conditions.
- **The Algorithm 1 and 2** shows the details of the KNN and THE Particle swarm Intelligence algorithm used in this paper.

### *E. Proposed Method*

- Data Acquisition:
  Sub-activities: Connect to Binance API. Retrieve real-time market data. Download historical data (if needed). Store data in the database.
- Data Analysis:
  Sub-activities: Preprocess data (e.g., cleaning, normalization). Perform statistical analysis (e.g., moving averages, Bollinger Bands). Apply machine learning algorithms to identify patterns and trends. Analyze sentiment from social media and news.
- Strategy Selection:
  Sub-activities: Backtests various trading strategies using historical data. Analyze backtesting results and performance metrics (e.g., Sharpe ratio, drawdown). Select the optimal strategy based on market conditions and risk parameters.
- Trade Signal Generation:
  Sub-activities: Generate trading signals based on the chosen strategy and real-time data analysis. Calculate entry and exit points for trades. Filter signals based on risk management rules.
- Risk Management:
  Sub-activities: Monitor risk exposure for individual trades and the overall portfolio. Calculate risk metrics (e.g., Value at Risk, drawdown). Trigger stop-loss orders if risk thresholds are exceeded.
- Order Placement:
  Sub-activities: Send buy and sell orders to Binance exchange based on generated signals. Specify order parameters (e.g., price, quantity, type). Monitor order execution status.
- Trade Execution:
  Sub-activities: Receive order execution confirmations from the exchange. Record trade details (e.g., time, price, quantity). Update portfolio positions and holdings.
- Performance Monitoring:
  Sub-activities: Calculate key performance metrics (e.g., profit/loss, Sharpe ratio, drawdown). Display performance data in dashboards and reports. Allow users to analyze performance and optimize strategies.

The figure 1 represnts the usage of the kite api to fetch the live data and on that KNN and KNN-PSO and the ensembled based trading algorithm is executed to generate recommendation of BUY,SELL and WAIT.

### *F. Dataset*

The dataset used for backtesting and evaluating the performance of trading strategies was obtained from *TradingView*, a widely used platform for market analysis and backtesting.

**Algorithm 1** Standard K-Means Algorithm

**Input:** $N = \{x_1, \ldots, x_n\}$; $M = \{\mu_1, \ldots, \mu_k\}$
**Output:** Cluster assignments and centroids
**while** not converged **do**
  **Classification:**
  **for** each $x_i \in N$ **do**
    **for** each $\mu_j \in M$ **do**
      Calculate Euclidean distance from $x_i$ to $\mu_j$
    **end for**
    Assign $x_i$ to the closest centroid $\mu_j$
  **end for**
  **Centroid Calculation:**
  **for** each $\mu_j \in M$ **do**
    Calculate new centroid $\mu_j$
  **end for**
  **Convergence:**
  **if** centroids $M$ remain unchanged **then**
    **end**
  **end if**
**end while**

**Algorithm 2** Particle Swarm Intelligence Algorithm

**Input:** Parameters $P$, Objective function $F$, Termination criteria
**Output:** Best solution found
**Initialization:**
Initialize swarm population $S$ with random solutions
**for** each solution $s_i \in S$ **do**
  Evaluate the fitness of $s_i$
  Update personal best $pbest_i$
  **if** $pbest_i$ is better than the global best $gbest$ **then**
    Update $gbest$ to $pbest_i$
  **end if**
**end for**
**while** termination criteria not met **do**
  **Movement:**
  **for** each solution $s_i \in S$ **do**
    Update velocity of $s_i$ based on $pbest_i$ and $gbest$
    Move $s_i$ to a new position
  **end for**
  **Evaluation:**
  **for** each solution $s_i \in S$ **do**
    Evaluate the fitness of $s_i$
    Update $pbest_i$
    **if** $pbest_i$ is better than $gbest$ **then**
      Update $gbest$ to $pbest_i$
    **end if**
  **end for**
**end while**

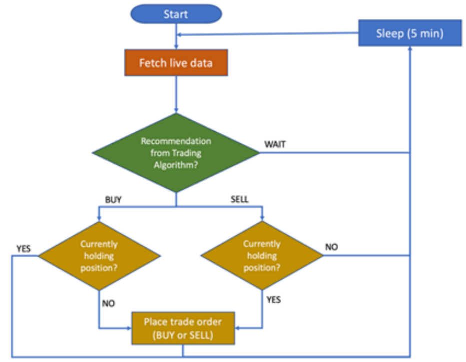


Fig. 1. Flowchart diagram

This dataset includes historical price data for various financial instruments, covering a range of asset classes such as stocks, indices, commodities, and currencies. The historical data spans over a substantial time period, enabling comprehensive analysis of different market conditions, including bullish, bearish, and sideways trends.

### G. Data sets

The dataset taken from NIFTY consists of the following key components:

- **Date and Time**: The timestamp for each data entry, represents the date and time at which the market data was recorded.
- **Open Price**: The price of the financial instrument at the beginning of the time interval.
- **High Price**: The highest price recorded during the time interval.
- **Low Price**: The lowest price recorded during the time interval.
- **Close Price**: The price of the financial instrument at the end of the time interval.
- **Volume**: The total number of units traded for the financial instrument during the time interval.
- **Shares Traded** Represents the sell and buy status.
- **Turnover** The price in crore.

### H. Backtesting Environment

The backtesting environment simulated real-world trading conditions using the processed dataset. TradingView's backtesting platform provided a realistic simulation of historical market behavior, allowing for the accurate evaluation of each strategy's performance. The simulation included transaction costs such as slippage and commission fees to ensure that the backtested results closely mimicked real trading outcomes.

## V. RESULTS

Evaluation based on Metrics –
The below figure shows the performance of our model against the evaluation parameters discussed earlier.

Random Forest Regressor Model:
Random Forest Regressor Model for Trading [6] Analysis–

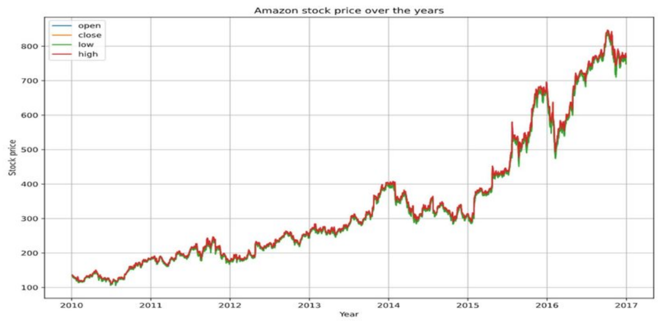


Fig. 2. Share price vs date graph

(Red: Actual Stock Price Movement, Blue: Bot predicted Stock Price Movement)
X-axis: 'TSLA' Stock share prices of test dataset and predicted prices. Y axis: Dates of trade.The figure 2 presents a graph where the Share Price is plotted against the Date, showcasing the performance of the Random Forest Model. The figure 3

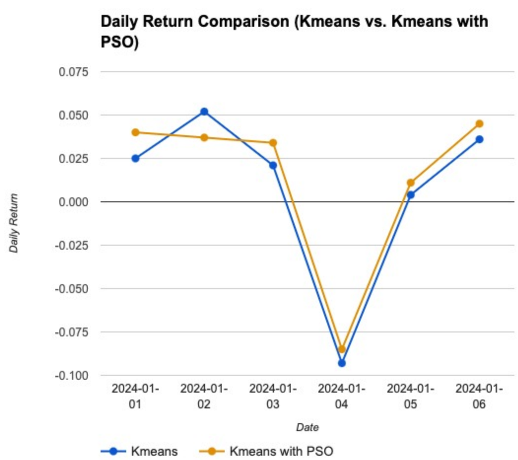


Fig. 3. Performance of KNN and KNN-PSO

shows the preference of addition of particle swarm optimization in KNN. Later we have also used an ensembled model i.e KNNPSD with LSTM to show the performance of KNNPSO in short term Trading and the figure in 4

## VI. CONCLUSION

The Trading Autobots streamline trading, allowing for lucrative results with less time and effort. Processes are eased for novice traders, and sophisticated tactics are quickly and precisely executed by expert traders. Because automation lightens the cognitive burden, traders can concentrate on making strategic decisions. Combining machine learning with financial expertise improves trading strategy optimization and decision-making. The result section shows the effectiveness of KNN-PSO in short-term trading because of its volatile performance as compared to LSTM, and by incorporating Kite connect API traders can pragmatically discover opportunities, reduce risks, and adjust to market situations by utilizing data-driven insights.

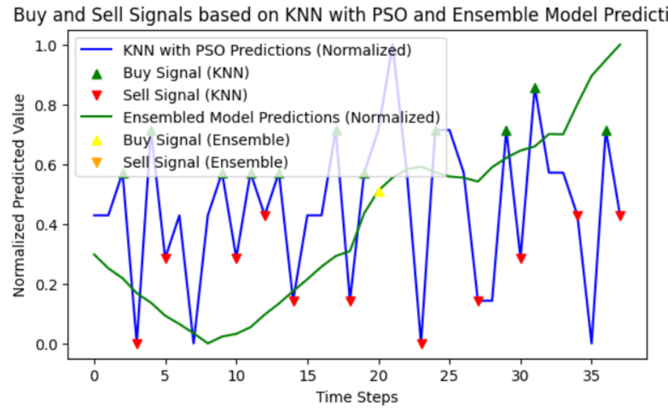


Fig. 4. Buy and Sell Signals based on KNNPSO and Ensemble model

## REFERENCES


[1] Vaibhav Aggarwal, Shallu Batra, Mahender Yadav, and Pankaj Kumar. Literature review on algorithmic trading in financial markets. In *2023 International Conference on Sustainable Islamic Business and Finance (SIBF)*, pages 76–80, 2023.

[2] Monira Essa Aloud and Nora Alkhamees. Intelligent algorithmic trading strategy using reinforcement learning and directional change. *IEEE Access*, 9:114659–114671, 2021.

[3] Yasmeen Ansari, Sadaf Yasmin, Sheneela Naz, Hira Zaffar, Zeeshan Ali, Jihoon Moon, and Seungmin Rho. A deep reinforcement learning-based decision support system for automated stock market trading. *IEEE Access*, 10:127469–127501, 2022.

[4] Salvatore M. Carta, Andrea Corriga, Anselmo Ferreira, Alessandro Sebastian Podda, and Diego Reforgiato Recupero. A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning. *Applied Intelligence*, 51:889–905, 2020.

[5] Adrian Ettlin. Evolutionary hedging model for power trading to maximize daily profits. In *2020 17th International Conference on the European Energy Market (EEM)*, pages 1–6, 2020.

[6] El Mehdi Ferrouhi and Ibrahim Bouabdallaoui. A comparative study of ensemble learning algorithms for high-frequency trading. *Scientific African*, 24:e02161, 2024.

[7] Boming Huang, Yuxiang Huan, Li Da Xu, Lirong Zheng, and Zhuo Zou. Automated trading systems statistical and machine learning methods and hardware implementation: a survey. *Enterprise Information Systems*, 13(1):132–144, 2019.

[8] Xiaoman Huang, Chang Wu, Xiaoqi Du, Hong Wang, and Ming Ye. A novel stock trading utilizing long short term memory prediction and evolutionary operating-weights strategy. *Expert Systems with Applications*, 246:123146, 2024.

[9] Sahil Mhatre Medha Mathur, Satyam Mhadalekar and Vanita Mane. Algorithmic trading bot. *ITM Web of Conferences*, 40, 2021.

[10] Jakub Michańków, Paweł Sakowski, and Robert Ślepaczuk. Lstm in algorithmic investment strategies on btc and s&p500 index. *Sensors*, 22(3), 2022.

[11] Md Sumon Gazi Nisha Gurung, Md Rokibul hasan and Md zahidul Islam. Algorithmic trading strategies: Leveraging machine learning models for enhanced performance in the us stock market. *Journal of Business and Management Studies*.

[12] Sangdon Park, Joohyung Lee, Ganguk Hwang, and Jun Kyun Choi. Event-driven energy trading system in microgrids: Aperiodic market model analysis with a game theoretic approach. *IEEE Access*, 5:26291–26302, 2017.

[13] Joaquín Pérez-Ortega and Nelva Nely Almanza-Ortega. The k-means algorithm evolution. *IntechOpen*, 2019.

[14] Nabeel M Shah. Trading strategies for autonomous agents in financial markets. *University of Western Sydney*, 2013.

[15] Binaebi Gloria Bello Sunday Tubokirifuruar Tula Olubusola Odeyemi Wilhelmina A Addy, Adeola Olusola AjayiNifise and Titilola Falaiye. Algorithmic trading and ai: A review of strategies and market impact. *World Journal of Advanced Engineering Technology and Sciences*, page 258–267, 2024.