International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(Research Article)

Algorithmic options trading bot using machine learning

Abhinav Srivastava [1], Garima Srivastava [1,*] and Lalita Kumari [2]

[1] *Department of Computer Science and Engineering, Amity University Uttar Pradesh, India.*
[2] *Department of Computer Science and Engineering, Amity University Patna, India.*

International Journal of Science and Research Archive, 2026, 19(02), 808-820

Publication history: Received on 02 April 2026; revised on 10 May 2026; accepted on 12 May 2026

Article DOI: https://doi.org/10.30574/ijsra.2026.19.2.1055

Abstract

Financial markets have witnessed a rapid transformation with the integration of automation and data-driven decision-making systems. Traditional trading approaches, which rely heavily on human intuition and manual execution, are often limited by emotional biases, delayed reaction times, and inconsistent strategies. In contrast, algorithmic trading systems enable systematic, rule-based, and high-speed execution of trades, significantly improving efficiency and consistency. However, most existing algorithmic trading systems either rely on static rule-based strategies or require complex infrastructure for real-time deployment, making them less accessible for academic exploration and small-scale implementation. This project presents the design and development of a multi-instrument algorithmic trading bot framework powered by machine learning, capable of generating intelligent trading signals across different financial instruments such as equity indices (NIFTY, BANKNIFTY) and commodities (Crude Oil, Natural Gas). The system focuses on building a scalable and modular architecture where a common machine learning decision engine is shared across multiple trading bots, each configured for a specific instrument. The proposed framework operates through a structured pipeline consisting of market data acquisition, feature engineering using technical indicators, predictive modeling, strategy mapping, and back testing. Historical market data is collected using financial data APIs and processed to extract meaningful features such as moving averages and momentum indicators like the Relative Strength Index (RSI). These features are used to train supervised machine learning models, specifically Logistic Regression, to predict short-term market direction. The predicted direction is then mapped to options trading strategies, generating actionable signals such as “Buy At-The-Money Call” or “Buy At-The-Money Put”. To evaluate the effectiveness of the proposed system, a paper trading and back testing module is implemented. This module simulates trading decisions over historical data without involving real capital, allowing performance assessment in a risk-free environment. The system calculates key performance metrics such as cumulative profit/loss, trade frequency, and directional accuracy. Experimental results indicate that even modest prediction accuracy can lead to profitable outcomes when combined with structured risk management and disciplined execution logic. A key contribution of this project is the development of a multi-bot architecture, where the same predictive model can be extended to multiple instruments with minimal modification. This demonstrates the scalability of the system and its potential application in real-world trading environments. Additionally, a lightweight dashboard interface is proposed using modern web technologies to visualize trading signals, bot performance, and system status, making the framework user-friendly and accessible. The project emphasizes modularity, simplicity, and reproducibility. It avoids dependency on complex trading infrastructure while still demonstrating the core principles of algorithmic trading and machine learning integration. The system is designed as a prototype for research and educational purposes, with future scope including real-time data integration, advanced machine learning models such as LSTM networks, broker API connectivity, and deployment as a fully automated trading platform.

Keywords: Algorithmic Trading; Machine Learning; Multi-Instrument Trading Bot; Logistic Regression; Technical Indicators; Back testing; Financial Markets; Paper Trading; Options Trading; Predictive Modelling; Trading Strategy Automation

* Corresponding author: Garima Srivastava

1. Introduction

Financial markets have experienced a substantial transformation with the transition from manual trading mechanisms to fully electronic and automated systems. Earlier trading practices relied heavily on human expertise, intuition, and manual interpretation of market data. Although such approaches offered flexibility, they were inherently constrained by cognitive biases, limited processing capacity, and delays in execution. The increasing complexity and speed of modern markets have reduced the effectiveness of purely manual decision-making.

Algorithmic trading has emerged as a key development in this context, enabling the automation of trading decisions through predefined computational rules. These systems enhance efficiency, execution speed, and consistency by leveraging real-time data processing capabilities. However, traditional rule-based algorithms are often limited by their static nature, as they operate on fixed conditions that may not generalize well across varying market regimes. Financial markets are dynamic systems influenced by multiple interacting variables, and rigid strategies frequently fail to capture such evolving patterns.

Machine learning techniques provide a data-driven approach to address these limitations by allowing models to infer relationships directly from historical data. In contrast to rule-based systems, these methods can model complex and non-linear interactions among market variables. Supervised learning approaches are particularly relevant for predicting short-term price movements, where models are trained on labeled datasets to estimate future outcomes. Even incremental predictive improvements can contribute to enhanced trading performance when combined with appropriate execution frameworks. Furthermore, machine learning supports feature extraction and representation learning, enabling the transformation of raw financial data into more informative inputs for decision-making.

Despite its potential, the application of machine learning in financial markets is associated with several challenges, including high levels of noise, non-stationarity of data, and susceptibility to overfitting. Consequently, predictive models must be designed and evaluated with caution to ensure robustness and generalizability. Additionally, machine learning should not be viewed as a standalone solution but rather as a component within a broader trading system that incorporates domain knowledge, strategy formulation, and risk management.

Another critical aspect of modern trading systems is the ability to operate across multiple financial instruments. Many existing studies focus on single-asset frameworks, which limits their applicability in real-world scenarios. Different asset classes exhibit distinct statistical properties and behavioral patterns, necessitating a flexible and scalable system design. A multi-instrument framework enables the use of a unified computational architecture while accommodating instrument-specific characteristics through configurable components.

This work aims to develop a machine learning-based trading framework capable of generating signals across multiple instruments using historical market data. The proposed system integrates data preprocessing, feature engineering based on technical indicators, predictive modeling, and strategy execution within a modular architecture. A backtesting mechanism is incorporated to evaluate performance under simulated conditions. The scope of the study is restricted to research and prototype development, with an emphasis on interpretability, structural clarity, and extensibility rather than real-time deployment or high-frequency trading.

The contribution of this study lies in demonstrating a structured approach to integrating machine learning into algorithmic trading systems while maintaining scalability and transparency. By combining principles from finance, machine learning, and software engineering, the proposed framework provides a foundation for future research and development in intelligent trading system design.

2. Literature Review

The field of algorithmic trading and financial prediction has attracted significant attention from researchers over the past two decades. With the increasing availability of market data and computational power, various approaches have been proposed to model market behavior and develop automated trading systems. This section reviews existing research in the areas of algorithmic trading, machine learning in finance, technical indicator-based strategies, and multi-instrument trading systems.

2.1. Algorithmic Trading Systems

Algorithmic trading refers to the use of automated systems to execute trading decisions based on predefined rules and strategies. Early work in this domain primarily focused on rule-based systems, where trading decisions were derived from mathematical models and statistical techniques. Robert Kissell [1] discussed the design and implementation of algorithmic trading strategies, emphasizing execution efficiency and transaction cost minimization. These systems improved trade execution speed but lacked adaptability to dynamic market conditions. Further advancements introduced quantitative trading frameworks that incorporated statistical arbitrage and time-series analysis. Ernest Chan [2] highlighted the importance of systematic trading approaches and demonstrated how algorithmic models can exploit market inefficiencies. However, these approaches still relied heavily on handcrafted rules and assumptions about market behavior.

2.2. Machine Learning in Financial Markets

The application of machine learning techniques in financial prediction has been extensively studied. Andrew Lo [3] explored the adaptive nature of financial markets and proposed that machine learning models could better capture non-linear relationships compared to traditional statistical models. Supervised learning models such as Logistic Regression, Support Vector Machines (SVM), and Decision Trees have been widely used for predicting stock price movements. Alexandre Tsantekidis [4] applied machine learning techniques to high-frequency trading data and demonstrated improved predictive performance over traditional methods. In another study, Sezer Obay [5] conducted a comprehensive survey of machine learning models in financial forecasting, concluding that ensemble and hybrid models often outperform single-model approaches. However, the study also highlighted challenges such as overfitting and data noise.

2.3. Deep Learning Approaches

With the rise of deep learning, researchers have explored neural network-based models for financial prediction. Zhang Guo [6] demonstrated the effectiveness of Long Short-Term Memory (LSTM) networks in capturing temporal dependencies in stock price data. These models showed improved performance in sequential data prediction tasks. Similarly, Kim Kyoung-jae [7] applied artificial neural networks to stock market prediction and reported higher accuracy compared to traditional statistical models. Despite their advantages, deep learning models require large datasets and significant computational resources, making them less suitable for lightweight implementations.

2.4. Technical Indicators and Feature Engineering

Technical indicators play a crucial role in financial prediction models by transforming raw price data into meaningful features. John Murphy [8] emphasized the importance of indicators such as moving averages, RSI, and MACD in identifying market trends and momentum. Research by Brock William [9] analyzed the effectiveness of moving average and trading range break rules, showing that these indicators can provide statistically significant signals under certain market conditions. Another study by Neely Christopher [10] evaluated technical analysis in foreign exchange markets and found that technical indicators can be profitable when combined with proper risk management strategies.

2.5. Hybrid Trading Systems

Hybrid systems that combine machine learning with technical indicators have gained popularity in recent years. Patel Jigar [11] proposed a hybrid approach integrating technical indicators with machine learning models, achieving improved prediction accuracy. Similarly, Shah Dev [12] developed a model combining multiple indicators and classification algorithms, demonstrating enhanced performance compared to standalone models. These studies highlight the importance of combining domain knowledge (technical indicators) with data-driven methods (machine learning).

2.6. Back testing and Performance Evaluation

Back testing is a critical component of algorithmic trading systems, allowing strategies to be evaluated on historical data. Pardo Robert [13] discussed methodologies for designing and testing trading systems, emphasizing the importance of avoiding overfitting. Bailey David [14] introduced statistical techniques to evaluate backtesting performance and detect false discoveries. The study highlighted the risks of data snooping and over-optimization in trading strategies.

2.7. Multi-Instrument Trading Systems

Most early research focused on single-instrument trading systems. However, recent studies have explored multi-instrument frameworks. Atfalatis George [15] proposed models capable of predicting multiple financial assets simultaneously, demonstrating improved diversification and robustness. Another study by Valavanis Kimon [16] explored intelligent systems for multi-market trading, highlighting the need for scalable architectures.

2.8. Challenges in Financial Prediction

Despite significant advancements, several challenges remain in applying machine learning to financial markets. Fama Eugene [17] introduced the Efficient Market Hypothesis (EMH), which suggests that market prices already reflect all available information, making prediction difficult. Malkiel Burton [18] further supported this view, arguing that stock price movements follow a random walk pattern. However, more recent studies, such as Lo Andrew [19], suggest that markets are not perfectly efficient and that exploitable patterns may exist under certain conditions.

2.9. Recent Trends and Developments

Recent research has focused on integrating alternative data sources and advanced machine learning techniques. Fischer Thomas [20] explored the use of deep learning for financial time series forecasting, demonstrating improved predictive capabilities. Dixon Matthew [21] investigated the application of reinforcement learning in trading systems, where agents learn optimal strategies through interaction with the market environment. Other studies, such as Krauss Christopher [22], provide comprehensive reviews of machine learning applications in finance, highlighting the growing importance of data-driven approaches.

2.10. Research Gaps Identified

2.10.1. Based on the reviewed literature, several gaps can be identified

- Many systems rely on single-instrument analysis, limiting scalability [15][16].
- Deep learning models, while powerful, require high computational resources [6][20].
- Rule-based systems lack adaptability to changing market conditions [1][2].
- There is limited focus on lightweight, modular architectures suitable for academic implementation.
- Few studies integrate machine learning, technical indicators, and multi-instrument design into a unified framework.

3. Methodology

3.1. Overview of the Proposed System

The proposed system is a multi-instrument algorithmic trading framework designed to automate the process of market analysis, prediction, and trade signal generation using machine learning techniques. The system is built with a modular architecture to ensure scalability, flexibility, and ease of integration with additional financial instruments. Unlike traditional trading systems that rely solely on static rule-based strategies, the proposed system integrates data-driven machine learning models with technical analysis, allowing it to adapt to varying market conditions. The system is capable of processing historical market data, extracting relevant features, generating predictive insights, and mapping those predictions to actionable trading strategies. The overall workflow of the system can be divided into the following major stages

- Data Acquisition
- Data Preprocessing
- Feature Engineering
- Model Training and Prediction
- Strategy Formulation
- Back testing and Performance Evaluation
- Multi-Instrument Deployment

Each stage is interconnected and contributes to the overall decision-making capability of the system.

3.2. System Architecture

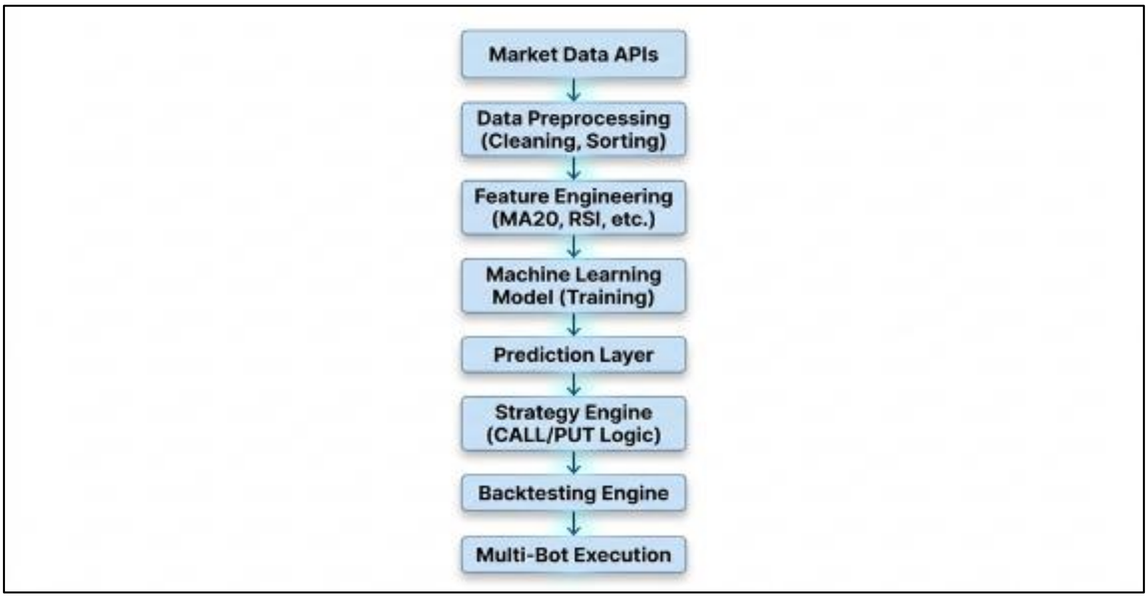

Figure 1 A flowchart explaining the system's architecture

The system follows a pipeline-based architecture, where each module performs a specific transformation on the input data. This design ensures that the system remains modular, allowing individual components to be modified or upgraded independently without affecting the entire system. The architecture is also designed to support parallel execution across multiple instruments, enabling the deployment of multiple trading bots that share a common machine learning engine but operate on different datasets.

3.3. Data Acquisition

Data acquisition is the foundational step of the system, as the performance of the machine learning model heavily depends on the quality and reliability of the input data. The system utilizes financial data APIs to retrieve historical market data for multiple instruments. The dataset includes time-series data consisting of daily price movements and trading volume. This data is structured in tabular format, where each row represents a single trading day. In addition to the data acquisition process, it is important to emphasize that the reliability of the entire system is heavily dependent on the quality of the collected data.

Any inaccuracies or inconsistencies at this stage can propagate through subsequent stages such as feature engineering and model training, ultimately affecting the performance of the system. Therefore, careful validation of the data source and consistency checks are essential to ensure that the dataset accurately represents real market conditions. Furthermore, the use of historical data across multiple time periods allows the system to capture diverse market behaviors. This diversity ensures that the model is exposed to different trading scenarios, including trending and non-trending markets, thereby improving its generalization capability.

Table 1 A sample of the dataset components being used

Field	Description
Date	Trading date
Open	Opening price of the instrument
High	Highest price during the day
Low	Lowest price during the day
Close	Closing price
Volume	Total traded volume

3.4. Market Price Trend Visualization

Figure 2 A line chart depicting market pricing trends of NIFTY 50

The closing price trend helps in visualizing the overall movement of the market over time. It forms the basis for further technical analysis and feature extraction. This helps us in getting a better understanding of which model is to be be used.

3.5. Data Preprocessing

Preprocessing ensures data quality and consistency. The steps used in data preprocessing are as follows

- Removal of Null values
- Sorting to maintain a proper chronological order
- Index alignment for having a consistent set of time series.

An additional consideration in data preprocessing is the need for maintaining temporal integrity. Since financial data is time-dependent, it is crucial to ensure that the chronological order of the data is preserved throughout the preprocessing stage. Any disruption in sequence may lead to incorrect model training and unrealistic results. Moreover, preprocessing not only prepares the data for analysis but also enhances computational efficiency. Clean and well-structured data reduces processing time and minimizes the chances of runtime errors, thereby contributing to smoother implementation of the system.

3.6. Feature Engineering

Feature engineering transforms raw data into meaningful indicators. There are majorly 2 indicators that we will be using for feature engineering. (Taking the sample bot into consideration)

Moving Average (MA20)- Used to identify trends. (Fig 3a): The Moving Average is a trend-following indicator used to smooth price fluctuations and identify the general direction of the market. In this project, a 20-day moving average (MA20) is used, which calculates the average closing price over the past 20 trading sessions.

Relative Strength Index (RSI)- Used to measure momentum. (Fig 3b): The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and magnitude of price movements. It is used to identify overbought and oversold conditions in the market. The RSI value ranges between 0 and 100. A value above 70 indicates that the asset is overbought and may experience a price correction, while a value below 30 indicates that the asset is oversold and may experience a price reversal. In this project, RSI is calculated using a 14-day window, which is the standard period used in financial analysis.

Figure 3A Price vs Moving Average (MA20) chart

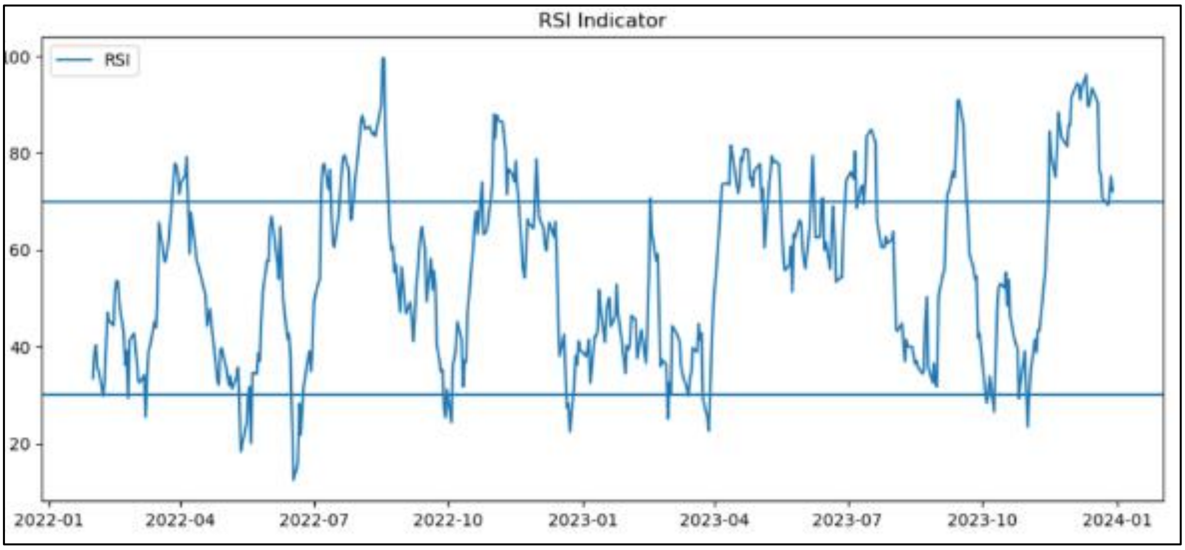

Figure 3B RSI indicator diagram to show how RSI is supposed to be used

Feature engineering plays a pivotal role in bridging the gap between raw data and meaningful insights. By transforming basic price information into informative indicators, the system is able to capture underlying patterns that are not immediately visible in raw data.

3.7. Machine Learning Model

The proposed system adopts a supervised machine learning framework to predict short-term market direction. The primary objective is to classify whether the market is likely to move upward or downward in the subsequent time interval using historical price information and derived features. In contrast to conventional rule-based approaches, the model learns directly from data, allowing it to identify underlying patterns and relationships among variables such as price movements, trend behavior, and momentum indicators.

The prediction task is formulated as a binary classification problem, where the output variable represents the expected direction of the market. A value of one corresponds to an anticipated upward movement, while zero indicates a predicted downward movement. This formulation simplifies the decision-making process and aligns the model output with actionable trading signals.

A crucial component of the modeling process is the construction of the target variable. In this work, the target is defined based on the comparison of closing prices between consecutive time steps, specifically focusing on the next period's closing value. The definition of this variable plays a significant role in shaping the learning objective, as it determines the patterns the model attempts to capture and directly influences prediction quality.

The input space consists of both raw market data and engineered features derived from technical analysis. These features are designed to capture meaningful aspects of market behavior, enhancing the model's ability to distinguish between different market conditions. By combining original and transformed inputs, the system improves its representation of underlying dynamics without relying solely on raw price information.

For the classification task, Logistic Regression is employed as the core learning algorithm. This method is selected due to its computational efficiency, ease of implementation, and strong interpretability. It provides probabilistic outputs that are well-suited for binary decision-making while maintaining robustness against overfitting, particularly in comparison to more complex models. Although advanced techniques such as deep learning could potentially offer improved performance, the chosen approach is sufficient for demonstrating the integration of machine learning within a structured trading framework. To ensure reliable evaluation, the dataset is partitioned into training and testing subsets, typically following an 80–20 split. The training portion is used to fit the model, while the remaining data serves to assess its predictive capability on unseen samples. This separation helps in obtaining an unbiased estimate of performance and reduces the likelihood of overly optimistic results.

Model performance is evaluated using classification accuracy, which measures the proportion of correct predictions relative to the total number of observations. While accuracy alone may not fully capture trading effectiveness, it provides a straightforward baseline metric. In financial applications, even modest predictive performance can be valuable when integrated with disciplined execution strategies and risk management techniques.

Once trained, the model produces directional predictions for each observation, which are subsequently utilized by the strategy layer to generate trading signals. These signals form the basis for simulated trade decisions within the overall system. Overall, the machine learning component serves as the decision-making core of the framework. By extracting patterns from historical data and translating them into predictive insights, it enables the system to move beyond static rules and supports more adaptive and data-driven trading behavior.

Table 2 Target Variable as per the closing price

Price	**Close**	**Target**
Date		
2022-01-31	17339.849609	1
2022-02-01	17576.849609	1
2022-02-02	17780.000000	0
2022-02-03	17560.199219	0
2022-02-04	17516.300781	0

3.8. Strategy Generation

The strategy generation component is responsible for translating the directional predictions produced by the machine learning model into executable trading actions. While the predictive model provides an estimate of future market movement, it does not inherently define how trades should be initiated or managed. To address this gap, a rule-based decision layer is incorporated, which systematically maps model outputs to specific trading strategies.

The strategy is designed using a hybrid approach that integrates machine learning predictions with a technical momentum indicator, specifically the Relative Strength Index (RSI). This combination ensures that trading decisions are not based solely on statistical predictions but are also validated through market momentum conditions. By requiring alignment between predictive signals and technical indicators, the system enhances the reliability of trade selection. The decision logic operates by evaluating both the predicted market direction and the prevailing momentum state. When the model indicates a potential upward movement and the RSI suggests that the market is not in an overbought condition, the system initiates a call option position. Conversely, when a downward movement is predicted and the RSI confirms that the market is not oversold, a put option position is selected. In situations where these conditions are not simultaneously satisfied, the system refrains from executing any trade. This selective approach helps in avoiding entries during extreme or unfavorable market conditions, thereby contributing to risk reduction. By integrating predictive modeling with technical validation, the strategy aims to minimize false signals and improve decision quality. The structured rule-based framework enforces consistency in execution, ensuring that trading actions are governed by predefined criteria rather than discretionary judgment. This not only promotes disciplined trading behavior but also reduces the likelihood of excessive or unnecessary trades. Overall, the strategy layer serves as a critical bridge between model outputs and real-world trading implementation. It converts abstract predictions into concrete actions while maintaining a controlled and systematic decision-making process. Without this layer, the predictive insights generated by the machine learning model would lack practical applicability. Through its design, the module ensures robustness, consistency, and alignment with market conditions, making it an essential component of the overall trading system.

Table 3 Simple table showing strategy logic

Condition	Action
Prediction = 1 and RSI < 70	Buy Call Option
Prediction = 0 and RSI > 30	Buy Put Option
Otherwise	No Trade

3.9. Back testing Methodology

Back testing is employed to assess the effectiveness of the proposed trading strategy by applying it to historical market data. It simulates how the system would have behaved under past market conditions, thereby providing an approximate indication of its practical performance. As a validation mechanism, back testing plays a crucial role in identifying both the strengths and potential weaknesses of the strategy before any real-world deployment. Although it offers valuable insights, its outcomes must be interpreted carefully, as they are derived from historical data and may not necessarily generalize to future market behavior. Nevertheless, it remains an indispensable step in the development and evaluation of trading systems.

The back testing procedure involves systematically applying the strategy to historical datasets and generating trade signals based on predefined logic. Each signal is treated as a simulated trade, and the corresponding profit or loss is calculated. Over time, these individual trade outcomes are aggregated to track the evolution of total capital, providing a comprehensive view of the strategy's financial performance.

To maintain simplicity and focus on core strategy evaluation, several assumptions are adopted. Trades are executed at the closing price of each period, and factors such as transaction costs, slippage, and market impact are not considered. Additionally, the simulation begins with a fixed initial capital, allowing consistent comparison of performance across different time intervals. While these assumptions simplify the analysis, they also introduce a degree of idealization that should be acknowledged when interpreting results. The performance of the strategy is measured using multiple evaluation metrics, including total profit or loss, final capital value, number of executed trades, and prediction accuracy. Together, these indicators provide a balanced perspective on both the profitability and operational characteristics of the system.

An important tool in performance evaluation is the equity curve, which illustrates the progression of capital over time. A consistently upward-trending curve generally reflects stable and profitable behavior, whereas irregular fluctuations or downward trends may indicate increased risk or instability in the strategy. This visual representation aids in understanding not only overall returns but also the consistency of performance.

The results obtained from back testing suggest that even moderate levels of predictive accuracy can lead to positive outcomes when combined with a well-defined execution framework. This observation underscores the importance of integrating predictive models with disciplined strategy design rather than relying solely on high prediction precision. Despite its usefulness, back testing has inherent limitations. It does not account for real-world trading frictions such as brokerage costs, bid-ask spreads, or execution delays. Furthermore, it assumes ideal conditions where trades are executed exactly as intended. Most importantly, reliance on historical data means that past performance cannot be considered a definitive indicator of future success. These limitations highlight the need for cautious interpretation and further validation before practical deployment.

3.10. Multi-Instrument Design

The proposed framework is structured to support trading across multiple financial instruments, enabling the deployment of separate yet coordinated trading units within a unified system. Rather than constructing isolated models for each asset, the design adopts a consolidated architecture in which a common machine learning framework is applied across different instruments while allowing each dataset to be processed independently. This approach ensures consistency in methodology while preserving the flexibility required to accommodate instrument-specific characteristics.

The system is capable of handling a diverse range of assets, including equity indices such as NIFTY 50 and BANKNIFTY, as well as commodities like Crude Oil and Natural Gas. These instruments differ significantly in terms of volatility, liquidity, and underlying market drivers, making them suitable for evaluating the adaptability of the proposed system across varied trading environments. From an architectural perspective, each instrument is managed by an independent trading unit that follows a standardized processing pipeline. This pipeline includes data acquisition, feature generation, predictive modeling, and strategy execution. Although the operational flow remains consistent, each unit functions autonomously, allowing parallel processing and reducing interdependencies between instruments. Such a modular arrangement simplifies system expansion, as additional instruments can be incorporated without altering the core framework. The multi-instrument design offers several practical advantages. It enhances scalability by enabling the system to operate simultaneously across multiple markets, thereby increasing potential opportunities for trade generation. It also supports diversification, as different instruments may respond differently to market conditions, reducing overall risk exposure. Furthermore, the reuse of a common modeling framework minimizes development complexity and improves efficiency in both implementation and maintenance. Overall, the multi-instrument architecture strengthens the practical relevance of the proposed system by demonstrating its ability to function across diverse trading contexts. The modular structure not only facilitates scalability and flexibility but also ensures that the system can be extended and refined with minimal effort. This design highlights how a unified machine learning framework can be effectively adapted to multiple financial instruments while maintaining consistency, efficiency, and ease of future development.

4. Results and Discussion

The proposed trading framework was evaluated using both predictive performance and back testing outcomes on historical data of the NIFTY 50 over a two-year period. The analysis combines quantitative metrics with qualitative observations to assess not only statistical performance but also practical trading behavior across varying market conditions, including trending and sideways phases. The Logistic Regression model, trained on technical features such as moving averages and RSI, achieved an accuracy of approximately 51–52% on unseen data. Although modest, such performance is meaningful in financial applications, where even a slight edge above random prediction can be leveraged for profitability. The model exhibited stable behavior across time and maintained a balanced distribution of upward and downward predictions, indicating the absence of directional bias and an ability to adapt to changing market dynamics. Trading signals generated from model outputs were further refined using RSI-based conditions, resulting in a selective execution mechanism. A significant number of instances resulted in no trade decisions, reflecting the system's conservative approach. This filtering improved trade quality by avoiding entries during extreme market conditions, reducing unnecessary exposure and limiting overtrading. Back testing results demonstrate that the strategy is capable of generating positive returns despite moderate predictive accuracy. Starting with an initial capital of ₹100,000, the system produced a return of approximately 4.47%. The profitability is attributed to the structured integration of prediction and execution logic rather than reliance on model accuracy alone. The equity curve shows a gradual upward trajectory with controlled fluctuations, indicating stable performance, limited drawdowns, and consistent capital growth. The strategy's effectiveness lies in its ability to convert probabilistic predictions into disciplined trading actions. The combination of machine learning outputs with technical validation enhances decision reliability, reduces noise, and ensures systematic execution. Additionally, the approach demonstrates adaptability, as the use of momentum indicators allows the system to adjust to evolving market conditions.

To evaluate scalability, the framework was extended to multiple instruments, including BANKNIFTY, Crude Oil, and Natural Gas. The results indicate variation in performance across assets, with equity indices showing stable behavior and commodities exhibiting higher volatility but improved short-term opportunities. Natural gas displayed relatively inconsistent outcomes due to its highly volatile nature. These findings highlight the importance of instrument-specific tuning while confirming the flexibility and scalability of the proposed multi-instrument architecture. Overall, the results demonstrate that the integration of machine learning with a structured strategy framework can yield stable and profitable outcomes. The system emphasizes consistency, risk control, and adaptability, making it a practical foundation for further enhancements in algorithmic trading.

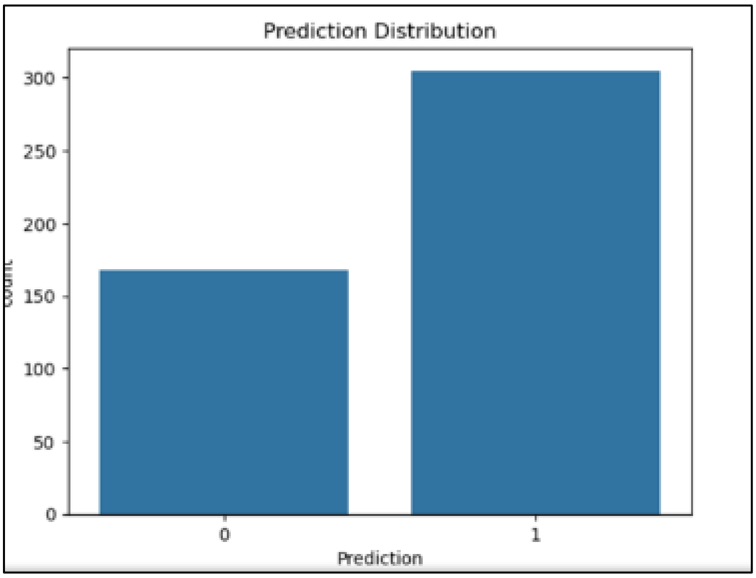

Figure 4A To understand the behavior of the model, the distribution of predicted classes was analyzed

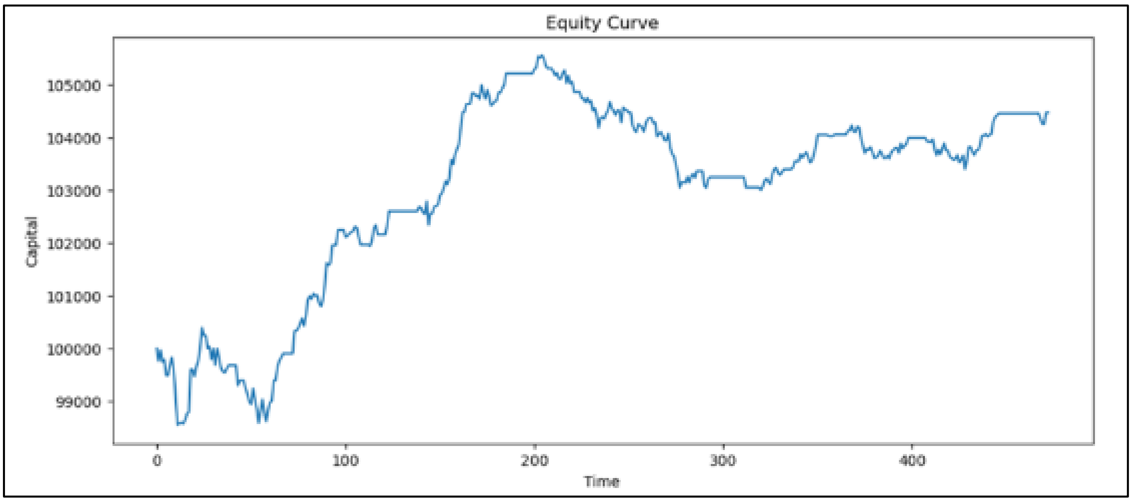

Figure 4B The equity curve provides a visual representation of the system performance over time by tracking the evolution of capital

5. Conclusion

This work set out to develop a machine learning-based algorithmic trading framework that combines predictive modeling with technical analysis to support systematic trading decisions. The proposed system follows a modular pipeline that includes data preprocessing, feature engineering, model training, strategy generation, and backtesting. By integrating Logistic Regression with technical indicators such as the moving average and Relative Strength Index, the framework demonstrates how data-driven prediction can be combined with market-based validation to produce a more disciplined trading approach. The findings show that even with moderate predictive accuracy, the system is capable of generating positive results when the predictions are translated through a structured execution layer. This is an important outcome, as it reflects a key principle in trading systems: useful performance depends not only on prediction quality, but also on how signals are filtered, interpreted, and applied in practice. The use of RSI as a constraint on trade execution helped reduce unnecessary entries and improved the overall selectivity of the strategy.

The study also confirmed that financial markets remain difficult to model with complete accuracy. Market noise, changing regimes, and hidden external influences continue to limit the predictive power of simple models. The back testing results therefore should be understood as evidence of feasibility rather than a guarantee of future performance. At the same time, they show that a well-structured framework can still produce meaningful outcomes even under these limitations. A further contribution of this project is its extension to multiple instruments through a unified architecture. This design improves flexibility and demonstrates that the same core framework can be applied across different markets, although performance naturally varies depending on the behavior of each instrument. Such variation reinforces the need for instrument-specific tuning in practical deployments.

In conclusion, the project successfully demonstrates the potential of combining machine learning with algorithmic trading in a scalable and structured manner. It provides a strong foundation for future enhancements such as richer feature sets, more advanced learning models, stronger risk controls, and real-time deployment. Overall, the study shows that machine learning can add genuine value to trading systems when it is supported by sound strategy design and careful system architecture.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] R. Kissell, The Science of Algorithmic Trading and Portfolio Management. Academic Press, 2013.

[2] E. P. Chan, Quantitative Trading: How to Build Your Own Algorithmic Trading Business. Wiley, 2009.

[3] A. W. Lo, Adaptive Markets: Financial Evolution at the Speed of Thought. Princeton University Press, 2017.

[4] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Using Deep Learning for Price Prediction by Exploiting Stationary Limit Order Book Features," Applied Soft Computing, vol. 65, pp. 134–147, 2018.

[5] O. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review," Applied Soft Computing, vol. 90, 2020.

[6] Z. Guo, H. Wang, Q. Chen, and Y. Liu, "A Stock Market Forecasting Model Combining Two-Dimensional CNN and LSTM," IEEE Access, vol. 6, pp. 73250–73261, 2018.

[7] K. J. Kim, "Financial Time Series Forecasting Using Support Vector Machines," Neurocomputing, vol. 55, pp. 307–319, 2003.

[8] J. J. Murphy, Technical Analysis of the Financial Markets. New York Institute of Finance, 1999.

[9] W. Brock, J. Lakonishok, and B. LeBaron, "Simple Technical Trading Rules and the Stochastic Properties of Stock Returns," Journal of Finance, vol. 47, no. 5, pp. 1731–1764, 1992.

[10] C. J. Neely, P. A. Weller, and R. F. Dittmar, "Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach," Journal of Financial and Quantitative Analysis, vol. 32, no. 4, pp. 405–426, 1997.

[11] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting Stock Market Index Using Fusion of Machine Learning Techniques," Expert Systems with Applications, vol. 42, no. 4, pp. 2162–2172, 2015.

[12] D. Shah, H. Isah, and F. Zulkernine, "Stock Market Analysis: A Review and Taxonomy of Prediction Techniques," International Journal of Financial Studies, vol. 7, no. 2, 2019.

[13] R. Pardo, The Evaluation and Optimization of Trading Strategies. Wiley, 2008.

[14] D. H. Bailey, J. Borwein, M. López de Prado, and Q. J. Zhu, "The Probability of Backtest Overfitting," Journal of Computational Finance, vol. 20, no. 4, pp. 39–69, 2017.

[15] G. S. Atsalakis and K. P. Valavanis, "Surveying Stock Market Forecasting Techniques – Part II: Soft Computing Methods," Expert Systems with Applications, vol. 36, no. 3, pp. 5932–5941, 2009.

[16] K. P. Valavanis, Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy. Springer, 2007.

[17] E. F. Fama, "Efficient Capital Markets: A Review of Theory and Empirical Work," Journal of Finance, vol. 25, no. 2, pp. 383–417, 1970.

[18] B. G. Malkiel, A Random Walk Down Wall Street. W. W. Norton & Company, 1973.

[19] A. W. Lo, "The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective," Journal of Portfolio Management, vol. 30, no. 5, pp. 15–29, 2004.

[20] T. Fischer and C. Krauss, "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[21] M. Dixon, D. Klabjan, and J. H. Bang, "Classification-Based Financial Markets Prediction Using Deep Neural Networks," Algorithmic Finance, vol. 6, no. 3–4, pp. 67–77, 2017.

[22] C. Krauss, X. A. Do, and N. Huck, "Deep Neural Networks, Gradient-Boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500," European Journal of Operational Research, vol. 259, no. 2, pp. 689–702, 2017.