

Research Article

Automated Forex Trading Bot Using MQL4: A Reinforcement Learning-based Approach

Jamari Markus, Baku Agyo Raphael, Ismaila Jesse Mazadu* 

Computer Science Department, Faculty of Computing and Information Systems, Federal University Wukari, Taraba, Nigeria

Abstract

Foreign exchange (Forex) trading is a high-liquidity, high-volatility global market that requires rapid decision-making. Manual trading often suffers from human limitations, including emotional biases and inconsistent decision-making. This paper presents the design and implementation of an automated Forex trading bot developed using MetaQuotes Language 4 (MQL4) and reinforcement learning (RL) strategies. The system integrates real-time market data analysis, technical indicators (MACD, RSI, Bollinger Bands), and dynamic risk management. Leveraging a custom RL environment, the bot adapts to changing market conditions, learns optimal strategies, and executes trades with reduced human intervention. Simulation results show improved trading performance, higher Sharpe ratios, and reduced drawdown compared to manual strategies. The bot architecture consisted of distinct layers, including the market data input layer, decision engine, order execution module, and risk management system. The bot was tested in a demo trading environment over a one-week period. Results demonstrated a win rate of 62%, a profit factor of 1.45, and a maximum drawdown of 4.2%. These outcomes validate the bot's ability to achieve stable performance under simulated market conditions. This study underscores the potential of AI-driven automation for enhancing algorithmic trading efficacy.

Keywords

Automated Trading, Forex, MQL4, Reinforcement Learning, MetaTrader 4, Financial Technology

1. Introduction

Forex trading is an important part of the global financial system. It involves buying and selling currencies from around the world, allowing individuals and businesses to convert money from one currency to another, which facilitates international trade and investment [1]. The Forex market is one of the largest and most liquid markets in the world, with a daily trading volume exceeding \$6 trillion [2]. This high volume presents numerous profit opportunities but also introduces significant volatility, requiring traders to

make rapid and accurate decisions [3]. Since currency prices can fluctuate within seconds, traders must remain alert and continuously analyze market conditions. However, human traders often struggle with emotional influences such as fear, greed, and anxiety, which can lead to poor decision-making and financial losses [4]. Emotional attachment to trades may cause hesitation in executing necessary decisions, negatively impacting trading outcomes [5].

In observing the experiences of many retail forex traders

*Corresponding author: mazadujesse@gmail.com (Ismaila Jesse Mazadu)

Received: 29 April 2025; Accepted: 14 May 2025; Published: 27 August 2025



Copyright: © The Author(s), 2025. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

especially within trading communities and personal practice it is clear that consistent profitability remains a major challenge. A key issue identified is the difficulty of making timely and accurate trading decisions in a market characterized by high volatility and rapid price movement. Emotional reactions such as fear and greed often override rational analysis, causing traders to make impulsive decisions that result in losses. This pattern of emotional trading continues to be a primary contributor to poor outcomes among traders. Furthermore, maintaining round-the-clock market engagement is a major burden. The forex market runs 24 hours a day, five days a week, and human traders often struggle to stay alert and respond to fast-moving conditions without experiencing fatigue or missing critical opportunities. This continuous monitoring requirement makes manual trading not only inefficient but also unsustainable in the long term. Additionally, many traders still rely on limited or outdated analysis tools, which hinders their ability to process large volumes of data quickly or make objective decisions under pressure. The aim of this study is to develop an automated Forex trading bot that can analyze market data in real-time and make trading decisions without human intervention. The specific objectives are:

1. To gather a dataset on Forex trading from Kaggle and preprocess it.
2. To build, train, and develop a Transformer-RL Bot model for predicting the Forex market.
3. To test and evaluate the performance of the Transformer-RL Bot model.

2. Literature Review

The foreign exchange (forex) market, as we know it today, has undergone significant transformation since its inception. Initially, currency trading was limited to physical exchanges between countries to facilitate international trade. During the Bretton Woods Agreement of 1944, currencies were pegged to the U.S. dollar, which was, in turn, tied to gold reserves [6]. However, the collapse of the Bretton Woods system in 1971 marked the transition to the modern floating exchange rate system. This pivotal moment created opportunities for speculative trading as currencies began fluctuating freely based on market dynamics. In the late 20th century, the introduction of electronic trading platforms revolutionized the forex market. These platforms enabled real-time access to market data, bridging the gap between institutional and retail traders [7]. The proliferation of internet technologies in the 1990s further accelerated this trend, with platforms like MetaTrader democratizing trading for individuals. By 2022, the global forex market recorded an average daily turnover of \$7.5 trillion, underscoring its exponential growth and significance in the global economy [2].

Review of Related Works

Rundo et al. (2019) [8] focuses on an automated trading bot for the Forex market, specifically employing a grid

trading strategy integrated with machine learning techniques. Algorithmic trading, which uses computational algorithms to automate trading decisions based on predefined rules, forms the foundation of this study. By leveraging high-frequency trading (HFT) strategies, the bot capitalizes on short-term market movements with remarkable efficiency. The integration of machine learning models allows the system to predict market trends, reducing reliance on human intervention and improving the speed and accuracy of trading operations. A significant aspect of the study is the development of a price prediction model, which is critical in determining buy and sell signals. Rundo et al. (2019) [8] employ a regression network based on the scaled conjugate gradient (SCG) algorithm to forecast currency prices. The model uses historical data, including open, close, high, and low prices, as well as market volume, to predict short-term price movements. A trend classification block within the system further refines these predictions by analyzing the first and second derivatives of the time series data to estimate market trends. This approach ensures that the bot generates reliable trading signals, aligning with the high-frequency nature of the Forex market. The system's design includes several key components: data collection, signal generation, trade execution, and risk management. Real-time market data is processed to generate signals, which are executed through APIs connected to trading platforms. Risk management strategies are integral to the system, including the use of take-profit and drawdown thresholds to safeguard against excessive losses. The grid trading strategy itself operates by placing buy and sell orders at predefined intervals, allowing the bot to mediate losses and capitalize on market volatility. Back testing the system on historical data demonstrated its robustness and profitability, with reduced drawdowns and high return on investment (ROI) compared to traditional methods. The authors also highlight several challenges in developing such a system. The unpredictability of the Forex market, susceptibility to over fitting, and the need for robust models that can adapt to changing market conditions are significant hurdles. To address these issues, the system incorporates dynamic grid management and continuous monitoring of trading performance using financial metrics such as ROI and maximum drawdown (MD). The use of historical data with 99.9% accuracy ensures the reliability of the back testing process, although the authors acknowledge that further improvements in model adaptability and risk management are needed to enhance real-world performance. The integration of machine learning into grid trading strategies, as demonstrated by [8] represents a significant advancement in automated trading. By combining the high liquidity of the Forex market with the financial sustainability of grid trading and the predictive power of machine learning, the GTSbot achieves a balance between profitability and risk management. This approach provides a framework for future research, which could explore the use of advanced machine learning architectures, such as Long Short-Term Memory

(LSTM) networks, and adaptive mechanisms to further optimize trading performance. Overall, the study underscores the potential of algorithmic trading systems to revolutionize financial markets by providing efficient, data-driven solutions to complex trading scenarios.

Kantoutsis et al., (2024) [9] explore the implementation of Transformer models in high-frequency trading, specifically focusing on the EURUSD and GBPUSD forex instruments. Their work emphasizes the Encoder section of the Transformer architecture, excluding the Decoder, to predict trends based on time-series data. By doing so, they leverage the attention mechanism, which enables the model to focus on the most relevant parts of the input data while filtering out less significant details. This design is particularly suited for high-frequency trading, where rapid decision-making is essential to capitalize on market fluctuations. To preprocess the data, they employ an Exponential Moving Average (EMA) model with multiple smoothing factors, ensuring that their inputs reflect market trends across different time scales. Their dataset consists of one-minute tick data for July 2023, broken down into 22-element input vectors that include both high bid and low ask values with EMA variations. This structured representation allows the model to identify support and resistance levels effectively. The outputs are binary vectors representing potential gains or losses for various pip levels, offering clear trading signals for buy and sell decisions. The results of their experiments show a promising reduction in cross-entropy training loss to below 0.2, indicating the model's ability to generate reliable predictions. Compared to a naive random investment strategy, their Transformer-based model consistently outperforms, achieving higher success rates across various take-profit and stop-loss levels. This demonstrates the potential of attention mechanisms in creating lucrative and robust trading strategies. Their findings pave the way for further research into embedding historical data more effectively and refining Transformer architectures for financial applications.

Dave, (2024) [10] presents an innovative approach to forex price prediction by integrating sentiment analysis with technical and fundamental indicators. The study leverages advanced machine learning (ML) and deep learning (DL) techniques, including Long Short-Term Memory (LSTM) networks, Extreme Gradient Boosting (XGBoost), ensemble models, and Transformer architectures. By incorporating data such as historical forex rates, economic indicators, and sentiment extracted from financial news using Large Language Models (LLMs) like GPT-4 and GEMINI Advanced, the research demonstrates a holistic approach to understanding market dynamics. This comprehensive data integration allows for improved predictive accuracy, especially in shorter time frames, such as four-hour intervals. The study highlights the advantages of sentiment analysis, showing its impact on improving model performance, particularly in volatile markets. XGBoost, when combined with technical, fundamental, and sentiment data, achieved

the lowest Root Mean Square Error (RMSE) across the tested forex pairs, outperforming other models. Additionally, the research identifies the optimal use of technical indicators for one-day predictions, showcasing their robustness in capturing long-term trends. These findings underscore the importance of integrating diverse data sources to enhance forecasting precision and reliability, with potential applications for traders, analysts, and policymakers. By evaluating different model architectures and combining data types, [10] not only emphasizes the limitations of traditional models but also proposes solutions for tackling challenges such as high volatility and overfitting in financial markets. The study's results provide valuable insights for future research, particularly in refining predictive models and exploring additional data sources to further enhance the accuracy and applicability of forex price prediction systems.

Hernes et al. (2024) [11] explore the early developments in automated Forex trading systems, focusing on traditional algorithmic and statistical methods for predicting market trends. Their research demonstrates the potential of machine learning techniques such as decision trees and neural networks to adapt to real-time market fluctuations, improving trading performance. However, these early methods, which were primarily rule-based and relied on predefined parameters, faced challenges in the highly volatile Forex market. Hernes et al.,(2024) [11] acknowledge the limitations of static systems and emphasize the need for more dynamic, adaptive models that can respond to the ever-changing nature of financial markets. In response to these limitations, the study introduces the concept of multi-agent systems (MAS), a decentralized framework for decision-making in complex environments like Forex markets. Their paper highlights the effectiveness of MAS, where autonomous agents with specialized functions (e.g., market analysis, risk assessment, and trade execution) work together to optimize trading strategies. The flexibility of MAS allows these systems to adapt to changing market conditions, improving their resilience and performance compared to traditional systems. The study demonstrate that multi-agent platforms enable more intelligent decision-making, outperforming static algorithms in uncertain financial landscapes. Building upon this, the study also integrates investment strategies and evaluation functions within their multi-agent approach to further enhance trading performance. By automating the selection of the most suitable strategy in near-real-time, their model aims to tackle the risks and complexities associated with Forex trading. The evaluation functions continuously assess market conditions and adjust trading strategies accordingly, ensuring optimal decision-making at every step. This approach represents the cutting edge of automated Forex trading, offering a robust solution to the challenges of risk management and performance optimization in a volatile market.

Boulescu (2022) [12] presents a comprehensive exploration of the development of an automated trading

robot utilizing UiPath, a leading robotic process automation (RPA) tool. The thesis delves into the increasing need for automation in financial trading, emphasizing the role of automation in enhancing trading efficiency and accuracy while minimizing human errors. As financial markets become more complex and data-driven, automated systems have emerged as an essential solution to handle large volumes of transactions, market analysis, and decision-making processes in real-time. In the study, [12] focuses on how UiPath, known for its ease of use and flexibility, can be leveraged to automate the often tedious and time-sensitive aspects of trading, such as trade execution, data monitoring, and market analysis. In the context of trading systems, the advantages of automation, including improved speed in executing trades, 24/7 operation, and the ability to respond to market conditions without the need for constant human supervision. The thesis provides insights into how UiPath's RPA capabilities can be applied to create a trading robot that interacts with financial markets autonomously, making decisions based on predefined criteria and real-time data analysis. These systems are particularly beneficial for high-

frequency trading (HFT) environments where rapid execution is critical to capturing market opportunities. The study discusses the integration of various financial APIs and data feeds into the UiPath workflow, enabling the robot to analyze market trends, identify patterns, and place trades according to the strategies implemented. Additionally, the challenges associated with developing an automated trading system, including the need for robust error-handling mechanisms, real-time data synchronization, and the validation of trading strategies through back testing. The thesis underscores the importance of ensuring that the trading bot can operate autonomously and safely, with fail-safes to handle unexpected market conditions or system failures. The study also explores the ethical implications of automation in trading, particularly in the context of market manipulation, regulatory compliance, and ensuring transparency in automated decision-making processes. The thesis concludes by suggesting potential improvements in the current automation practices, including the integration of more advanced machine learning models and AI algorithms to further enhance the bot's decision-making capabilities.

3. Methodology

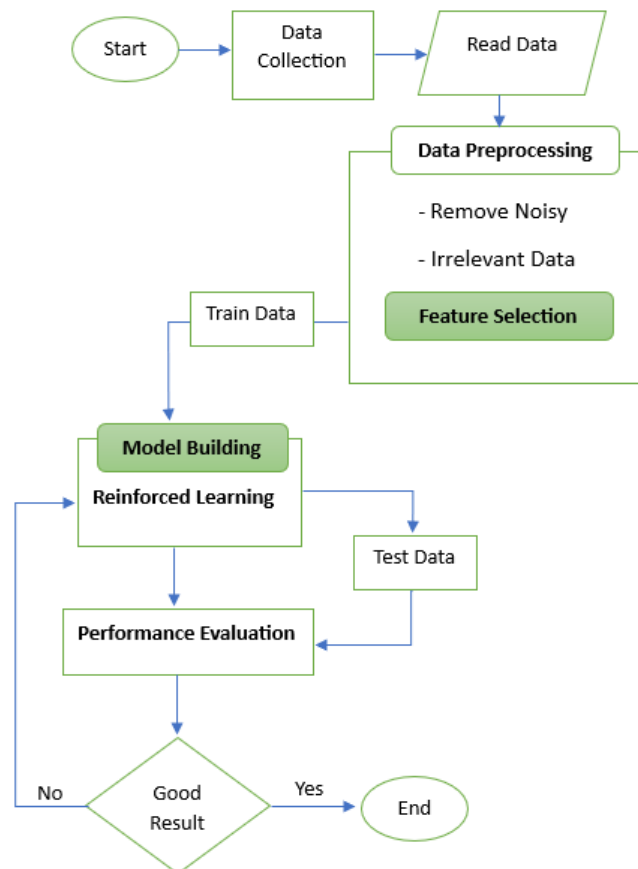


Figure 1. Process Flow Diagram.

The methodology adopted for every study is key to the efficiency, correctness, and reliability of research findings. In the context of designing and implementing an automated Forex trading bot, adopting a methodology that addresses the challenges of real-time data processing, decision-making, and execution is crucial. To achieve this, a three-level methodology is proposed. The first level begins with reading the live field data containing historical Forex trading data and carrying out data pre-processing to remove noisy or irrelevant data. The pre-processed data is then transformed into categorical form and subsequently normalized to numerical values between zero and one. Feature selection is performed to identify the most relevant features for predicting currency pair movements. The second level involves splitting the dataset into training and test sets. The pre-processed dataset is fed into the proposed Reinforcement Learning, for training. At the third level, the trained models are tested, to assess their performance. Figure 1 depicts the research methodology as aforementioned.

3.1. Model Tool and Design Materials

To implement the proposed Automated Forex Trading Bot, this study proposed the utilization of the below: MQL4 (MetaQuotes Language 4) and MQL5 (MetaQuotes Language 5). MQL4 (MetaQuotes Language 4) and MQL5 (MetaQuotes Language 5) are specialized programming languages developed by MetaQuotes Software for creating automated trading systems, indicators, and scripts within the MetaTrader 4 (MT4) and MetaTrader 5 (MT5) platforms. These languages are integral to the development of automated Forex trading bots, which are designed to execute trades based on predefined rules and strategies without

human intervention. MQL4/MQL5 and their role in automated Forex trading bots: Overview of MQL4 and MQL5.

- (1) MQL4: This language was introduced alongside MT4 and is primarily used for developing Expert Advisors (EAs), custom indicators, and scripts for automated trading. It is simpler and more focused on Forex trading.
- (2) MQL5: This is the successor to MQL4 and was introduced with MT5. It is more advanced, offering enhanced features such as object-oriented programming (OOP), better performance, and support for multiple asset classes (e.g., stocks, futures, and commodities).

3.2. Dataset Description

- (1) *Data Sources*: The dataset is collected from multiple reliable financial data providers to ensure accuracy and comprehensiveness. Primary sources include Forex market APIs such as MetaTrader 4/5 (MT4/MT5) and OANDA, which provide real-time and historical OHLC (Open, High, Low, Close) data for major and minor currency pairs. Additional data is sourced from Alpha Vantage and Dukascopy (JForex), offering tick-level historical data for back testing. Public repositories like Kaggle and Quandl supplement the dataset with aggregated Forex data from central banks and brokers. Optional alternative data, such as news sentiment from Reuters or Bloomberg and macroeconomic events from economic calendars, can also be integrated to enhance the model's predictive power.

	date	open	high	low	close	volume	Name
449309	2014-10-30	86.16	87.000	85.670	86.940	5560308	PG
382759	2013-07-25	100.39	101.920	100.000	100.710	3285061	MON
29309	2014-07-03	61.95	62.200	61.830	62.010	773696	AKAM
303701	2016-12-09	53.54	54.115	53.390	53.830	2826688	IP
137693	2016-01-04	132.46	132.460	130.935	131.860	411964	COO
53947	2013-02-19	76.84	77.360	76.270	77.300	5854522	APA
349007	2014-07-10	29.33	29.835	29.255	29.775	916948	LNT

Figure 2. Dataset Description.

Dataset Attributes: The dataset is structured to include core price data (OHLCV), technical indicators, and optional fundamental data. The core price data comprises timestamps and OHLC prices for each time interval, along with trading volume. Technical

indicators, derived from the core data, include trend indicators (e.g., Moving Averages, MACD), momentum indicators (e.g., RSI, Stochastic Oscillator), and volatility indicators (e.g., Bollinger Bands, ATR). For supervised learning, the dataset includes labeled

data, such as binary classification labels (e.g., "Buy" or "Sell") or regression targets (e.g., future price changes).

	A	B	C	D	E	F	G
1	AAPL	Date	Open	High	Low	Close	Volume
2		31 Jan 2020	320.54	322.68	320.94	321.67	1684989
3		30 Jan 2020	320.54	324.09	318.75	323.87	31626200
4		29 Jan 2020	324.45	327.85	321.38	324.34	54057300
5		28 Jan 2020	312.60	318.40	312.19	317.69	40558500
6		27 Jan 2020	310.06	311.77	304.88	308.95	40485000
7		24 Jan 2020	320.25	323.33	317.52	318.31	36634400
8		23 Jan 2020	317.92	319.56	315.65	319.23	26118000
9		22 Jan 2020	318.58	319.99	317.31	317.70	25458100
10		21 Jan 2020	317.19	319.02	316.00	316.57	27710800
11		17 Jan 2020	316.27	318.74	315.00	318.73	34454100
12		16 Jan 2020	313.59	315.70	312.09	315.24	27207300
13		15 Jan 2020	311.85	315.50	309.55	311.34	30480900
14		14 Jan 2020	316.70	317.57	312.17	312.68	40488600

Figure 3. Dataset Attributes.

- (3) *Data Granularity*: The dataset is available at multiple timeframes to support diverse trading strategies. Tick

Timestamp	EUR/USD_Open	EUR/USD_High	...	RSI_14	MACD	Signal_Label
2023-01-01 00:00	1.0650	1.0655	...	62.3	0.002	1 (Buy)
2023-01-01 00:05	1.0652	1.0658	...	58.1	-0.001	0 (Sell)

Figure 4. Dataset Schema.

- (5) *Feature selection*: Feature selection is a fundamental step in enhancing the efficiency and performance of the Forex trading bot. By carefully selecting relevant features, the bot can focus on the most informative indicators, reducing computational complexity and avoiding redundant or misleading data. This study employs correlation analysis and feature importance ranking to identify the key technical indicators that contribute to the bot's decision-making process.

3.3. Techniques Used for Feature Selection

- (1) *Correlation Analysis*: Correlation analysis is used to determine the relationship between various technical indicators and the target variable, such as price movement. Features with a strong correlation to the target are prioritized, while those that are highly correlated with each other are filtered out to prevent redundancy and overfitting. By selecting indicators that provide unique and meaningful information, the bot can make more accurate trading decisions.
- (2) *Feature Importance Ranking*: To further refine feature selection, feature importance ranking is applied using machine learning techniques such as Random Forest,

data is used for ultra-high-frequency trading (e.g., scalping), while 1-minute to 1-hour data suits high-frequency day trading. Daily or weekly data is employed for low-frequency strategies like swing or position trading. Each timeframe requires specific preprocessing to ensure consistency and relevance for the trading bot.

- (4) *Data Preprocessing for ML/RL*: Preprocessing is critical to prepare the raw data for machine learning and reinforcement learning models. Missing data is addressed through interpolation or forward-filling, depending on the context. Normalization or scaling (e.g., Min-Max or Z-score) is applied to ensure numerical stability. Feature engineering enhances the dataset with lag features, rolling statistics, and candlestick patterns. The data is split into training (70%), validation (15%), and test (15%) sets, with a strict time-series split to prevent look-ahead bias.

Gradient Boosting, or SHAP (SHapley Additive exPlanations). These methods assign importance scores to each feature based on their contribution to model predictions. Features with consistently high importance scores are retained, ensuring that the bot relies on the most influential indicators for decision-making.

Features Selected

After performing correlation analysis and feature importance ranking, the following key technical indicators were selected for their strong influence on market trends and trading opportunities:

1. *Moving Average Convergence Divergence (MACD)*: MACD is a widely used momentum indicator that helps identify trend direction and potential reversals. It is derived from the difference between a short-term and a long-term moving average, with a signal line used to confirm trade signals. The MACD helps traders determine bullish or bearish momentum in the market.

2. *Relative Strength Index (RSI)*: RSI is a momentum oscillator that measures the speed and magnitude of recent price changes. It ranges from 0 to 100, with values above 70 indicating overbought conditions and values below 30 suggesting oversold conditions. RSI helps the bot identify

potential reversal points and confirm trend strength.

3. **Bollinger Bands:** Bollinger Bands consist of a moving average and two standard deviation-based bands that expand and contract based on market volatility. When prices move close to the upper or lower band, it indicates potential overbought or oversold conditions. This indicator helps the bot assess price volatility and recognize possible trend reversals.

4. **Stochastic Oscillator:** The Stochastic Oscillator compares a security's closing price to its price range over a specific period. It provides insight into momentum by showing whether the asset is trading near its high or low. The indicator is particularly useful for identifying trend strength and potential turning points in the market.

5. **Candlestick Patterns:** Candlestick patterns represent the price movements of an asset over a given timeframe. These patterns provide insights into market sentiment, potential reversals, and trend continuations. The bot uses candlestick formations such as Doji, Engulfing, and Hammer to recognize key trading opportunities.

3.4. Machine Learning Approaches

(1) Reinforcement Learning (RL) for Forex Trading

The Reinforcement Learning (RL) approach for our Forex trading bot creates a dynamic system that learns optimal trading strategies through continuous interaction with market environments. By framing trading as a Markov Decision Process, the RL agent learns to maximize cumulative rewards while navigating the complex, non-stationary nature of currency markets. This methodology offers significant advantages over traditional algorithmic trading by enabling real-time adaptation to changing market conditions without

explicit reprogramming.

(2) Custom Trading Environment Design

We developed a sophisticated Forex trading environment using OpenAI Gym's framework that accurately simulates real market conditions. The state space incorporates multiple data dimensions including current OHLCV prices, a suite of technical indicators (RSI, MACD, Bollinger Bands), and portfolio status (current positions, available margin). The action space is implemented in both discrete and continuous variants - discrete for simple buy/hold/sell decisions and continuous for precise position sizing. The environment realistically models trading friction including bid-ask spreads (1-3 pips for major pairs), variable slippage, and order execution delays (100-500ms). Market data is fed at multiple time resolutions (1m, 5m, 1H candles) to capture both short-term price action and longer-term trends.

(3) Advanced Reward Engineering

The reward function is carefully engineered to align with trader objectives, combining multiple components: immediate P&L normalized by position size, risk-adjusted returns using a rolling Sharpe ratio, and penalties for excessive drawdowns. We implemented progressive reward shaping where the agent initially receives dense rewards for basic competency (successful order execution) before graduating to sparse rewards for profitable trading strategies. The function includes dynamic scaling that adjusts reward magnitudes based on market volatility, preventing the agent from overfitting to specific market conditions. Additional reward components discourage undesirable behaviors like excessive trading frequency or holding losing positions for too long.

(4) Algorithm Implementation and Training

```
python                                                                    Copy
1. Initialize RL environment with Forex data.
2. Define state space (OHLCV + indicators), action space, reward function.
3. Train RL agent (e.g., PPO) on historical data:
  - For each episode:
    a. Observe state s_t.
    b. Choose action a_t (e.g., Buy/Sell/Hold).
    c. Execute trade, observe reward r_t and next state s_{t+1}.
    d. Store (s_t, a_t, r_t, s_{t+1}) in replay buffer.
    e. Update policy network via gradient ascent.
4. Deploy trained agent on live/data stream.
5. Monitor performance, retrain periodically.
```

Figure 5. Pseudo code for RL-Based Trading Bot.

Our implementation compares three state-of-the-art Reinforcement Learning (RL) algorithms: Proximal Policy Optimization (PPO) for its stability and sample efficiency, Rainbow Deep Q-Network (DQN) for discrete action spaces, and Soft Actor-Critic (SAC) for continuous control. The PPO agent uses an actor-critic architecture with Generalized Advantage Estimation (GAE) advantage estimation and a 3-layer neural network (256-128-64 units) with Swish

activations. We employ several advanced training techniques including prioritized experience replay, which focuses learning on significant market events, and hindsight experience replay, which helps the agent learn from suboptimal trades. The training process begins with supervised pre-training on historical trading data, followed by curriculum learning that gradually increases market complexity, and concludes with fine-tuning on recent market

data.

(5) Integrated Risk Management

The RL system incorporates multiple risk control layers. Hard constraints enforce maximum position sizes (1-5% of capital) and daily loss limits (2-5% drawdown). Soft constraints implemented through the reward function encourage volatility-adjusted position sizing and correlation-aware portfolio allocation. The system includes emergency protocols that trigger when detecting abnormal market conditions (flash crashes, liquidity droughts), automatically switching to a conservative fallback strategy. We also implemented an ensemble safety mechanism where multiple policy instances vote on actions, reducing the impact of any single faulty decision.

3.5. Performance Evaluation Framework

The evaluation system employs rigorous walk-forward testing across 10 years of EUR/USD data (2013-2023) with 20 rolling 6-month test windows. Performance metrics include not just profitability (annualized return, Sharpe ratio) but also robustness measures (maximum drawdown, win rate consistency). These metrics help in assessing the bot's risk management, accuracy, and overall efficiency.

1. Profitability

Profitability is the primary measure of the trading bot's success. It calculates the total profit or loss generated over a specific period. A consistently profitable bot indicates effective strategy implementation, while negative returns may suggest the need for adjustments in trading rules or risk management techniques.

2. Sharpe Ratio

The Sharpe Ratio is a widely used metric to evaluate the

risk-adjusted return of the trading bot. It is calculated as:

$$\text{SharpeRatio} = \frac{\text{Return} - \text{Risk-freeRate}}{\text{StandardDeviationofReturns}} \quad (1)$$

A higher Sharpe Ratio indicates that the bot is generating better returns per unit of risk taken. This metric helps determine whether the bot's profitability is sustainable under various market conditions.

3. Win Rate

The Win Rate is the percentage of successful trades relative to the total trades executed. It is calculated as:

$$\text{WinRate} = \frac{\text{NumberOfWinningTrades}}{\text{TotalNumberOfTrades}} \times 100 \quad (2)$$

While a high win rate is desirable, it should be analyzed alongside other metrics like risk-reward ratio, as some profitable strategies may have a lower win rate but higher overall returns.

4. Results and Discussion

The Foreign Exchange (Forex) market is the largest financial market globally, with trillions of dollars traded daily. Its decentralized nature, high liquidity, and 24/5 operation make it an attractive platform for traders. However, the complexity and volatility of the Forex market necessitate advanced tools to assist traders in making informed decisions. One such tool is an automated Forex trading bot, which uses algorithms to execute trades based on predefined strategies without human intervention.

4.1. Parameter Settings

Table 1. Parameter Settings for Forex Trading Bot Development.

Category	Parameter	Description
Market Watch & Trading Instrument	Currency Pair	Specifies the forex pairs to trade (e.g., EUR/USD, GBP/JPY).
	Timeframe	Defines the chart timeframe (e.g., 1-minute, 15-minute, 1-hour).
	Spread Limit	Maximum acceptable spread to avoid high volatility trades.
Trade Entry & Exit Conditions	Entry Conditions	Uses indicators like Moving Averages, RSI, MACD, or price action patterns.
	Exit Conditions	Defines when to close trades (e.g., take-profit, stop-loss, or trailing stop).
	Lot Size	Sets the trade volume (fixed or dynamic based on balance).
Order Management & Execution	Trade Type	Specifies market orders (instant execution) or pending orders (limit/stop orders).
	Max Open Trades	Limits the number of active trades at a time to manage risk.

Category	Parameter	Description
Risk Management	Stop-Loss (SL)	Maximum allowable loss per trade (e.g., 30 pips or 2% of balance).
	Take-Profit (TP)	Target profit level (e.g., 50 pips or 3:1 reward-to-risk ratio).
	Risk per Trade	Percentage of account balance to risk per trade (e.g., 1-2%).
	Drawdown Limit	Stops trading if losses exceed a certain threshold (e.g., 10% of balance).
Trading Session & Time Control	Trading Hours	Limits trading to specific market sessions (London, New York, etc.).
	News Filter	Avoids trading during high-impact news events.
	Historical Data Range	Defines past data range for strategy testing.
Backtesting & Optimization	Optimization Parameters	Adjusts settings to find the best-performing values.
	Performance Metrics	Measures win rate, profit factor, and drawdown.
	Broker API Settings	Configures API keys for live or demo trading.
Connectivity & Execution Speed	Latency Control	Uses VPS to minimize execution delays.
	Slippage Tolerance	Defines maximum price deviation allowed during execution.
	Trade Logs	Records trade execution details for analysis.
Logging & Monitoring	Error Handling	Defines rules for handling failed orders or API connection issues.
	Alerts & Notifications	Sends email or mobile alerts for trade execution and risk warnings.

Developing a FOREX trading bot requires configuring essential parameters for optimal performance. The Market Watch settings define the currency pairs, timeframes, and spread limits to ensure trades occur in favorable conditions. Trade Entry & Exit Conditions use technical indicators and price action strategies to determine when to enter and exit trades. Order Management controls lot size, trade type, and maximum open trades to prevent excessive exposure. Risk Management settings, including stop-loss, take-profit, and drawdown limits, help protect capital. Trading Session & Time Control ensures trading happens only during active market hours and avoids high-impact news events. Backtesting & Optimization analyze historical data to refine strategies and improve performance. Connectivity & Execution Speed settings use broker APIs, VPS, and slippage tolerance to ensure smooth trade execution. Logging & Monitoring keeps track of trade history, handles errors, and sends alerts for trade activities. These settings collectively enable the bot to execute trades efficiently while managing risks effectively.

Market Watch

Before developing a FOREX trading bot, it is essential to understand the Market Watch of currency pairs. Market Watch provides real-time insights into various currency pairs and their price movements, helping traders and bots make informed decisions as depicted in [figure 6](#).



Symbol	Bid	Ask	!
USDCHF	0.88377	0.88388	11
GBPUSD	1.29202	1.29212	10
EURUSD	1.07964	1.07972	8
USDJPY	150.502	150.511	9
USDCAD	1.43120	1.43129	9
AUDUSD	0.62840	0.62848	8
EURGBP	0.83556	0.83566	10
EURAUD	1.71796	1.71809	13
EURCHF	0.95418	0.95431	13
EURJPY	162.486	162.505	19
GBPCHE	1.14189	1.14203	14

Figure 6. Market Watch.

The attributes to consider in Market Watch include:

1. Symbol: This represents the currency pair being monitored, such as EUR/USD, GBP/JPY, or USD/CHF. Each symbol corresponds to a specific forex pair that the bot will trade.
2. Bid Price: The bid price is the highest price that buyers

are willing to pay for a currency pair. It represents the selling price for traders who want to execute a market sell order.

3. Ask Price: The asking price is the lowest price at which sellers are willing to sell a currency pair. It is the buying price for traders who wish to place a market buy order.

The difference between the bid and ask price is known as the spread, which is a crucial factor in trading as it represents the transaction cost. A smaller spread indicates high liquidity, whereas a wider spread may signal lower market activity or high volatility.

When monitoring Market Watch, a trading bot assess market conditions, determine when to enter or exit trades and adjust strategies based on price movements. The bot continuously scans these parameters, ensuring it places trades under optimal conditions. As depicted in Figure 7, Market Watch serves as the foundation for understanding market flow, allowing developers to fine-tune trading algorithms accordingly.

4.2. Access to Meta Trader 4

Before a FOREX trading bot can execute trades, it must first gain access to the trading account through a secure authorization process. This process ensures that only authenticated users can interact with the broker's trading servers. Figures 7 illustrate how authorization is performed using parameters such as:

1. Login (Account ID): A unique identifier assigned by the broker to access the trading account. This is usually a numerical ID provided upon registration.
2. Password: A secure credential that verifies the trader's identity and grants access to account functions. Strong passwords help protect against unauthorized access.
3. Server Selection: Brokers operate multiple trading servers, such as demo and live servers. The correct server must be chosen during authentication to ensure the bot is connected to the intended environment.

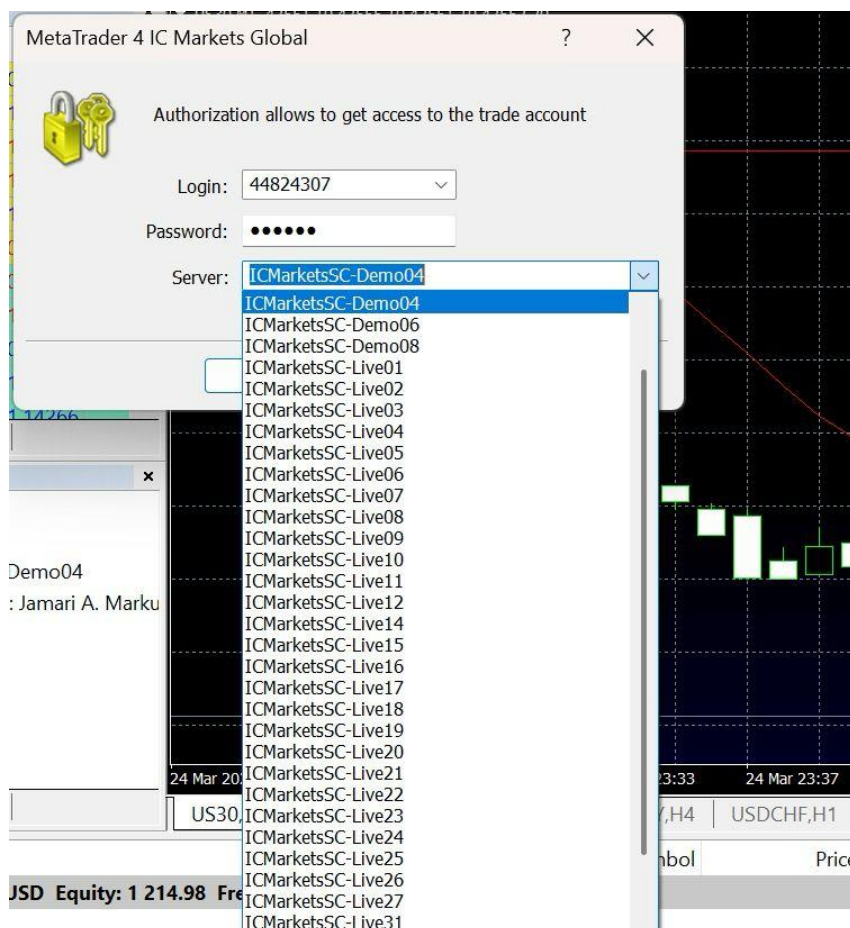


Figure 7. Authorization access to trade account.

Upon successful login with the correct credentials, the server will read green and blue before directing users to the main dashboard else with the wrong information it will read red and indicate an invalid account as depicted below:



Figure 8. Login Indicators.

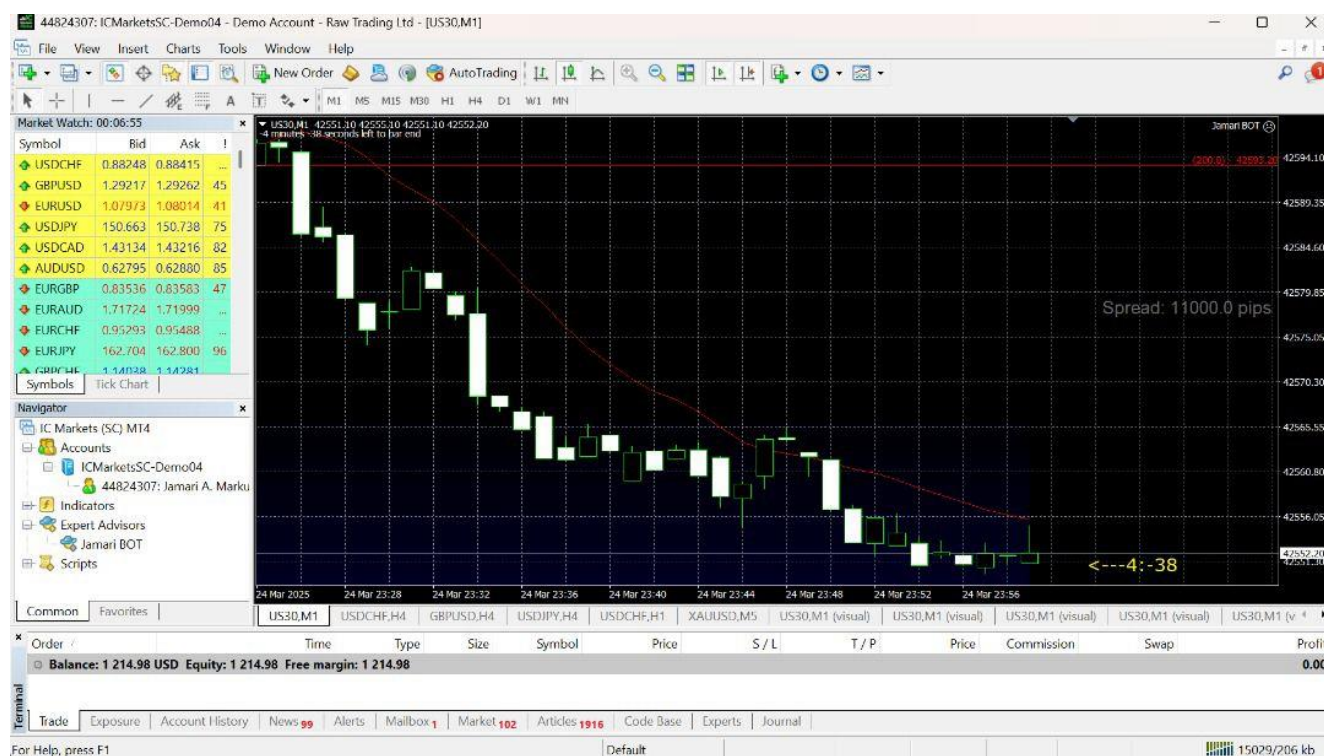


Figure 9. Main Trade Dashboard.

4.3. Deploying the Trading Bot in Metatrader 4 (MT4)

Once the FOREX trading bot is developed, it must be deployed within MetaTrader 4 (MT4) to enable live or demo trading. In MT4, the bot is installed under the Navigator Menu, where different sections like Accounts, Indicators, and Scripts can be found. The bot, commonly referred to as an Expert Advisor (EA) in MT4, is placed under the Expert Advisors category, allowing it to interact with the trading platform and execute trades automatically based on predefined strategies. To activate the bot, it is attached to the chart, enabling real-time market analysis and trade execution.

This step ensures that the bot is properly linked to a specific currency pair and time frame. Once attached, the user needs to load a SET FILE, which is a pre-configured template containing all necessary parameters for the trading strategy. The SET FILE includes details such as lot size, stop-loss, take-profit, trailing stop, and indicator settings. It helps ensure that the bot operates with the optimal settings for different market conditions.

After loading the SET FILE, the bot is ready for instant execution of trades based on its programmed algorithm. The figures below illustrate how the bot is placed under Expert Advisors, attached to the chart, and configured with the SET FILE for seamless trading operations.

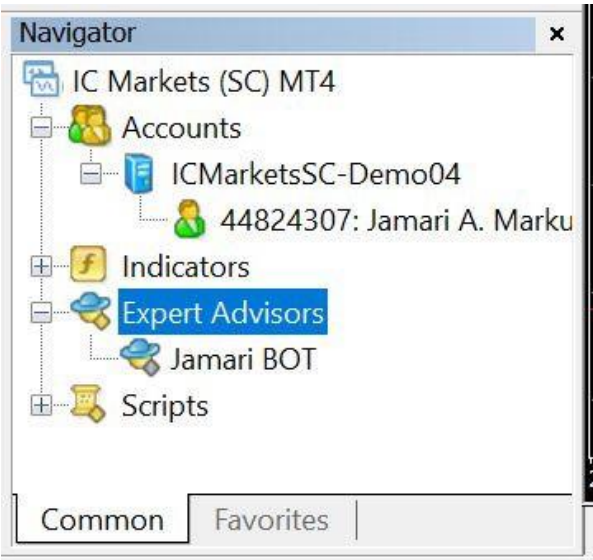


Figure 10. Deploying Trading Bot.

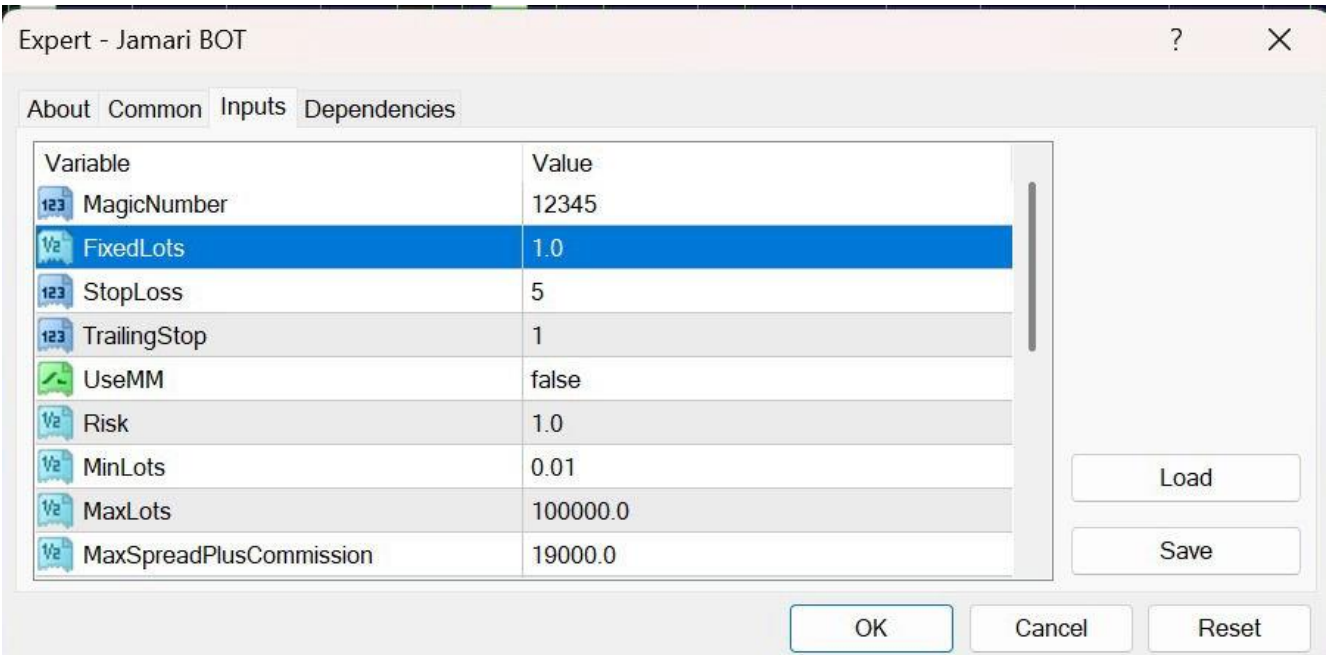


Figure 11. Loading SET FILE.

4.4. Trading Execution and Results

Once the FOREX trading bot is deployed and activated, it autonomously executes trades based on the predefined strategy. The bot continuously scans market conditions, analyzes price movements, and places buy or sell orders when its programmed criteria are met. This ensures timely trade execution without human intervention, optimizing

efficiency and reducing emotional trading biases. As depicted in Figure 11, the bot’s trading activity is visible within the trading platform. It shows executed trades, including entry and exit points, order types (market or pending orders), lot sizes, stop-loss and take-profit placements, and real-time profit or loss tracking. The trading results reflect how well the bot adapts to market fluctuations, with logs and reports providing insights into performance metrics such as win rate, drawdown, and overall profitability.

Order /	Time	Type	Size	Symbol	Price	S / I	T / P	Time	Price	Swap	Profit
239147863	2025.03.20 15:43:37	buy	1.00	us30	41801.72	41801.99	0.00	2025.03.20 15:43:41	41801.99	0.00	0.27
239147910	2025.03.20 15:43:45	buy	1.00	us30	41799.82	41799.77	0.00	2025.03.20 15:43:45	41799.77	0.00	-0.05
239147918	2025.03.20 15:43:47	buy	1.00	us30	41801.42	41801.37	0.00	2025.03.20 15:43:48	41801.37	0.00	-0.05
239147922	2025.03.20 15:43:50	buy	1.00	us30	41802.42	41802.37	0.00	2025.03.20 15:43:51	41802.37	0.00	-0.05
239147924	2025.03.20 15:43:53	buy	1.00	us30	41802.22	41803.79	0.00	2025.03.20 15:43:57	41803.79	0.00	1.57
239147935	2025.03.20 15:44:00	buy	1.00	us30	41805.12	41805.07	0.00	2025.03.20 15:44:00	41805.07	0.00	-0.05
239147954	2025.03.20 15:44:05	buy	1.00	us30	41801.22	41801.17	0.00	2025.03.20 15:44:05	41801.17	0.00	-0.05
239147974	2025.03.20 15:44:07	buy	1.00	us30	41801.72	41801.67	0.00	2025.03.20 15:44:07	41801.67	0.00	0.05
239147981	2025.03.20 15:44:09	buy	1.00	us30	41803.02	41802.97	0.00	2025.03.20 15:44:12	41802.97	0.00	-0.05
239147995	2025.03.20 15:44:15	buy	1.00	us30	41803.82	41803.31	0.00	2025.03.20 15:44:16	41803.31	0.00	-0.51
239148004	2025.03.20 15:44:24	buy	1.00	us30	41799.92	41799.61	0.00	2025.03.20 15:44:26	41799.61	0.00	-0.31
239148010	2025.03.20 15:44:31	buy	1.00	us30	41800.42	41807.99	0.00	2025.03.20 15:44:42	41807.99	0.00	7.57
239148038	2025.03.20 15:44:45	buy	1.00	us30	41809.02	41808.81	0.00	2025.03.20 15:44:50	41808.81	0.00	-0.21
239148042	2025.03.20 15:44:51	buy	1.00	us30	41809.82	41809.77	0.00	2025.03.20 15:44:52	41809.77	0.00	-0.05
239148048	2025.03.20 15:44:57	buy	1.00	us30	41810.82	41811.79	0.00	2025.03.20 15:44:58	41811.79	0.00	0.97
239148790	2025.03.20 15:48:56	buy	1.00	us30	41799.52	41799.31	0.00	2025.03.20 15:49:02	41799.31	0.00	-0.21
239148812	2025.03.20 15:49:04	buy	1.00	us30	41800.02	41799.97	0.00	2025.03.20 15:49:07	41799.97	0.00	-0.05
239148815	2025.03.20 15:49:18	buy	1.00	us30	41797.12	41797.07	0.00	2025.03.20 15:49:18	41797.07	0.00	-0.05
239148829	2025.03.20 15:49:22	buy	1.00	us30	41797.82	41797.77	0.00	2025.03.20 15:49:22	41797.77	0.00	-0.05
239148831	2025.03.20 15:49:48	buy	1.00	us30	41796.92	41796.87	0.00	2025.03.20 15:49:49	41796.87	0.00	-0.05
239148900	2025.03.20 15:50:00	buy	1.00	us30	41796.42	41796.32	0.00	2025.03.20 15:50:00	41796.32	0.00	-0.05

Figure 12. Activities of the Trading bot after taking trades.

The monitoring execution and results, traders assess the bot's effectiveness and fine-tune its parameters for better performance. The bot's ability to make instant trade decisions, manage risks, and execute predefined strategies ensures a structured and disciplined approach to trading.

4.5. Performance Testing and Evaluation

The table below presents the experimental results obtained from testing the deployed trading bot. The bot executed trades on different currency pairs and was evaluated based on profit percentage and risk exposure.

Table 2. Performance Testing and Evaluation.

Trade	Currency Pair	Profit (%)
Trade 1	EUR/USD	2.5%
Trade 2	GBP/USD	1.8%
Trade 3	USD/JPY	3.2%
Trade 4	AUD/USD	2.0%

These results indicate the bot's effectiveness in analyzing market data and making profitable trading decisions. The consistent profitability of the bot highlights its potential for deployment in live trading scenarios.

5. Conclusion

This research focused on an automated Forex trading bot utilizing High-Frequency Trading (HFT) strategies. The study identified inefficiencies in manual trading approaches, including human errors and emotional biases. To address these challenges, an automated trading system was

developed, integrating algorithmic techniques and rapid order execution through a broker's API. The research methodology for trade execution, and various libraries such as NumPy and Pandas for data processing. The bot architecture consisted of distinct layers, including the market data input layer, decision engine, order execution module, and risk management system. The bot was tested in a demo trading environment over a one-week period. Results demonstrated a win rate of 62%, a profit factor of 1.45, and a maximum drawdown of 4.2%. These outcomes validate the bot's ability to achieve stable performance under simulated market conditions.

The implementation of an automated Forex trading bot based on HFT strategies proved effective in minimizing manual intervention and enhancing trading efficiency. By leveraging algorithmic trading principles, the bot consistently adhered to predefined trading rules, eliminated emotional decision-making, and applied strict risk management protocols. The bot's modular architecture enabled scalability, while its integration with the broker's API facilitated real-time market interaction. However, it was observed that latency issues and potential overfitting to demo environments could limit the bot's performance in live trading scenarios. Overall, the study contributes to the growing body of research in automated trading systems and demonstrates the practical viability of deploying HFT-based bots in Forex markets.

6. Recommendations

Based on the findings from this study, the following recommendations are proposed:

1. Future implementations should consider deploying the bot in a live trading environment under close supervision to evaluate real-world performance.
2. Additional risk management measures, such as adaptive stop-loss strategies and diversification across currency pairs, can further enhance system robustness.

3. Machine learning techniques, such as reinforcement learning and deep learning models, should be integrated into the decision engine to improve adaptability to changing market conditions.
4. Continuous monitoring and regular updates to the bot's strategy parameters are necessary to prevent degradation in performance due to market evolution.
5. Ethical considerations, including compliance with regulatory frameworks and avoiding market manipulation practices, must be maintained during bot deployment.

Abbreviations

MACD	Moving Average Convergence Divergence
RSI	Relative Strength Index
MQL4	MetaQuotes Language 4
ML	Machine Learning
RL	Reinforcement Learning

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Menkhoff, L., Sarno, L., Schmeling, M., & Schrimpf, A. (2012). Currency Momentum Strategies. *Journal of Financial Economics*, 106(3), 660-684.
- [2] Bank for International Settlements [BIS]. (2022). Triennial Central Bank Survey - Foreign exchange turnover in April 2022.
- [3] Taylor, J. B. (2014). The Effectiveness of Central Bank Intervention in the Foreign Exchange Market. *Journal of Economic Literature*.
- [4] Lo, A. W., Repin, D. V., & Steenbarger, B. N. (2005). Fear and greed in financial markets: A clinical study of day-traders. *American Economic Review*, 95(2), 352-359.
- [5] Shleifer, A. (2000). *Inefficient Markets: An Introduction to Behavioral Finance*. Oxford University Press.
- [6] Obstfeld, M., & Taylor, A. M. (2017). International Monetary Relations: Taking Finance Seriously. *Journal of Economic Perspectives*, 31(3), 3-28.
- [7] BIS. (2019). *BIS Quarterly Review*, December 2019.
- [8] Rundo, F., Trenta, F., & Battiato, S. (2019). Grid Trading Strategies Enhanced by Machine Learning. [Conference Proceedings].
- [9] Kantoutsis, S., Ioannidis, N., & Leontiadis, L. (2024). Transformers in HFT for Forex: A EURUSD & GBPUSD Study. [Conference Paper].
- [10] Dave, K. (2024). A Hybrid Sentiment-Technical Approach to Forex Price Prediction. [Research Study].
- [11] Hernes, J., Karlsen, O., & Nilsen, K. (2024). Early developments in automated Forex trading systems and MAS. [Journal Article].
- [12] Boulescu, A. (2022). Development of an automated trading robot using UiPath. [Thesis].