Advanced Research on Information Systems Security
An International Journal · ISSN 2795-4609 | 2795-4560
Vol. 6, No. 2 (2026)
pp. 28–38
aris-journal.com

RESEARCH ARTICLE

Optimizing Trading Strategies with Genetic Algorithms: Design and Evaluation of the EvoTrade System

Diogo Silva[a,*], Nuno Mateus-Coelho[b,c]

[a] *Computer Engineering, University Lusófona do Porto, Porto, Portugal*

[b] *COPELABS, FCNET, Lusófona University, 1749-024 Lisbon, Portugal*

[c] *CTS-UNINOVA, Universidade NOVA de Lisboa, Center of Technology and Systems, 2829-516 Lisbon, Portugal*

Received: 2025-05-15 · Accepted: 2025-12-25 · Published: 2026-11-03 · DOI: https://doi.org/10.56394/aris2.v6i2.63

ABSTRACT

The EvoTrade system employs genetic algorithms (GAs) to automate the development and validation of trading strategies, addressing the inefficiencies of manual trading and limitations of existing algorithmic platforms. This paper presents EvoTrade's modular architecture, its GA-based optimization framework, and its performance in simulated and real-world trading environments. By integrating historical market data and advanced GA techniques, including island models, fitness sharing, and adaptive mutation, EvoTrade minimizes human intervention and enhances strategy adaptability in volatile financial markets. Evaluations across versions 1.6.2 to 1.7.1 show a fitness improvement from 0.30 to 2.50, with a TradingView simulation achieving a 68.29% win rate and a net profit of 4,139.12 USD. Challenges in convergence stability, real-world discrepancies, and overfitting risks are identified, highlighting areas for future refinement. EvoTrade's open design and dataset generation capabilities position it as a significant contribution to fintech and automated trading research, offering a scalable solution for traders and AI developers.

KEYWORDS

Genetic Algorithms; Trading Strategies; Financial Markets; Automation; Backtesting; Fintech

1. Introduction

Financial markets are characterized by high volatility, complexity, and unpredictability, posing significant challenges for traders relying on manual strategies. Emotional biases, such as fear or greed, often lead to inconsistent decision-making, while time-intensive analysis limits responsiveness to market changes [1]. Automated trading systems, such as MetaTrader, QuantConnect, or proprietary platforms, aim to address these issues by leveraging algorithmic approaches. However, these systems face limitations, including high costs, lack of transparency, and reliance on predictive models that struggle to capture non-linear market dynamics [2]. Genetic algorithms (GAs), inspired by natural evolution, offer a robust solution for optimizing complex, high-dimensional problems [3]. By iteratively applying selection, crossover, and mutation, GAs evolve solutions that adapt to dynamic environments, making them well-suited for trading strategy development. Prior research has demonstrated GA applications in portfolio selection [4], high-frequency foreign exchange trading [5], stock trading [6], and group trading portfolios [7]. These studies highlight GAs' ability to generate robust strategies, yet few systems integrate user management, data processing, and strategy optimization into a cohesive platform.

The EvoTrade system, developed as part of an undergraduate project at the University of Porto, addresses these gaps by providing a modular, scalable platform for automated trading. EvoTrade leverages GAs to create and validate trading strategies based on historical market data, targeting assets such as BTC, XRP, and ETH. Its objectives include automating strategy development, ensuring adaptability to market changes, and providing flexible parameter customization for diverse user needs. The system's GA core, refined across versions 1.6.2 to 1.7.1, incorporates advanced techniques such as island models, fitness sharing, and adaptive mutation to enhance strategy diversity and performance. Evaluations demonstrate EvoTrade's potential, with a TradingView

* Corresponding author: diogo.s1871@gmail.com
© 2026 The Author(s). Published by LAPI2S.
Cite as: D. Silva and N. Mateus-Coelho, "Optimizing Trading Strategies with Genetic Algorithms: Design and Evaluation of the EvoTrade System," *Adv. Res. Inf. Syst. Secur.*, vol. 6, no. 2, pp. 28–38, 2026, doi: 10.56394/aris2.v6i2.63.

simulation achieving a 68.29% win rate and a net profit of 4,139.12 USD. However, challenges in convergence stability, discrepancies between simulated and real-world results, and overfitting risks highlight areas for improvement. EvoTrade's open design and dataset generation capabilities position it as a foundation for AI-driven trading innovations, contributing to the growing intersection of fintech and machine learning. This paper provides a comprehensive overview of EvoTrade's design, implementation, and evaluation, positioning it within the broader context of GA-based trading systems. The document is structured as follows: Section 2 reviews related work, Section 3 details the system's architecture, Section 4 describes the methodology, Section 5 presents performance results, Section 6 discusses the findings, Section 7 concludes with future directions, and the Appendix includes supplementary materials.

2. Literature Review

Genetic algorithms have been extensively applied to financial trading, leveraging their ability to optimize complex, non-linear problems. [4] used GAs to optimize portfolio selection based on technical trading rules, such as moving averages and RSI, achieving improved risk-adjusted returns compared to traditional methods. Their approach focused on market timing, aligning with EvoTrade's use of technical indicators. Similarly, [5] developed high-frequency foreign exchange trading strategies using GAs, optimizing parameters for indicators like MACD and Stochastic Oscillators to maximize Sharpe ratios.

These studies underscore GAs' effectiveness in handling high-dimensional parameter spaces, a key feature of EvoTrade's optimization framework. In stock trading, [6] compared genotype representations for GA-based strategies, finding that binary encodings outperform real-valued ones in terms of convergence speed and strategy robustness. This informed EvoTrade's choice of a 80-bit binary genome. [7] extended GA applications to group trading portfolios, using a grouping GA to optimize multiple strategies simultaneously, achieving higher returns than single-strategy models. Their work highlights the potential for GAs to manage portfolio-level optimization, an area EvoTrade could explore in future iterations.

Pattern recognition-based trading algorithms, as explored by [8], demonstrate GAs' ability to identify complex market patterns, aligning with EvoTrade's use of logical operators (AND, OR, NOT) to combine indicators. [9] focused on foreign exchange trading rules using a single indicator across multiple timeframes, achieving consistent performance but limited flexibility compared to EvoTrade's multi-indicator approach. Alternative approaches include decision trees for stock trading [10], which combine indicators like EMA, MACD, and RSI, and reinforcement learning for forex trading [11], which optimizes strategies based on price change trends. [12] proposed an algorithmic trading framework with mean-variance optimization and early-stopping mechanisms, emphasizing risk management. [13] developed a cryptocurrency trading bot using price-action strategies, while [14] combined GAs with Bayesian optimization to fine-tune technical trading algorithms. Multi-objective optimization, as explored by [15], balances profit and risk using genetic programming, offering insights for EvoTrade's fitness function design. Despite these advances, existing systems often lack integration of user management, data processing, and strategy optimization, or are limited to specific markets (e.g., forex, stocks). EvoTrade addresses these gaps by providing a comprehensive platform that supports diverse sets and generates structured datasets for AI training, aligning with emerging trends in fintech [11], [15].

3. System Structure

3.1 Architecture Overview

EvoTrade is a modular platform comprising five core packages: User Management, Genetic Algorithm, Data Management, Strategies, and Support and Administration. Implemented in Python for the GA and technologies (HTML, CSS, JavaScript) for the interface, the system uses MongoDB Atlas for data storage, ensuring scalability and accessibility.

- **User Management:** Handles registration (UC-01), login (UC-02), and profile editing, with error handling for duplicate emails and invalid passwords [1]. Users can configure GA parameters (e.g., population size, mutation rate) via a web interface.
- **Genetic Algorithm:** Optimizes trading strategies (UC-03 to UC-05) using a 80-bit binary genome encoding technical indicators (e.g., RSI7, MACD, ATR, STOCH K) and logical operators (AND, OR, NOT). The GA supports four entry/exit conditions per strategy, enabling complex rule combinations.
- **Data Management:** Collects (UC-06) and updates (UC-07) historical OHLCV data via the Hyperliquid API, supporting assets like BTC, XRP, and ETH. Data integrity is ensured through validation checks and error logging.
- **Strategies:** Enables consultation of GA results (UC-08) and export as CSV files (UC-09), facilitating external analysis [4]. Exported strategies include entry/exit conditions, 4 performance metrics, and backtest results.
- **Support and Administration:** Manages user tickets (UC-10, UC-11) and system backups (UC-12), with automated notifications for errors or incomplete submissions.

Figure 1 illustrates the package diagram, showing interactions between modules.

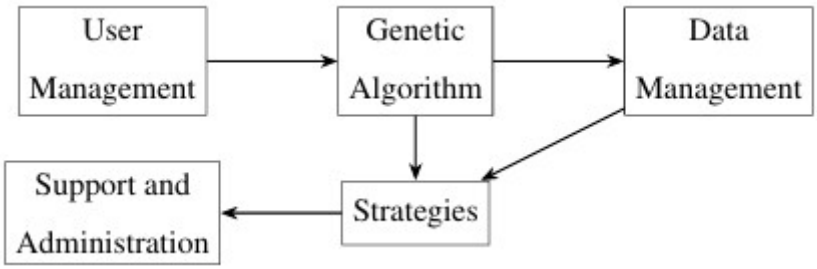


Figure 1. Package diagram of the EvoTrade system, illustrating interactions between User Management, Genetic Algorithm, Data Management, Strategies, and Support and Administration modules.

3.2 Development Process

EvoTrade was developed in two phases: Project I (requirements elicitation and planning) and Project II (design and implementation). Stakeholder interviews with 10 traders, analysts, and fintech professionals informed 12 use cases, modeled using UML diagrams. The system was iteratively prototyped, tested, and deployed on a web platform.

4. Methodology

4.1 Requirement Analysis

Requirements were gathered through semi-structured interviews with 10 stakeholders, including 5 traders, 3 analysts, and 2 fintech developers. Key needs included automation of strategy development, adaptability to market changes, and integration with reliable data sources. Twelve use cases (UC-01 to UC-12) were defined, specifying objectives, preconditions, and flows. For example, UC-01 (User Registration) requires a unique email and a password with at least 8 characters, including uppercase, lowercase, and numbers, with error handling for duplicates, in line with established account protection and data privacy measures [19]. Table 1 summarizes key use cases.

Table 1. Key Use Cases for EvoTrade

Use Case	Package	Description
UC-01	User Management	Register a new user with unique email and strong password.
UC-03	Genetic Algorithm	Create a trading strategy using GA with user-defined parameters.
UC-06	Data Management	Collect historical OHLCV data via Hyperliquid API.
UC-08	Strategies	Consult GA-generated strategy results and performance metrics.
UC-10	Support	Submit a support ticket for system issues.

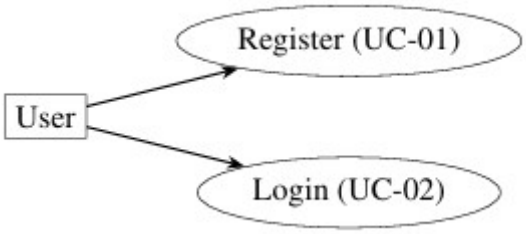


Figure 2. Use case diagram for User Management, showing Register (UC-01) and Login (UC-02) interactions.

4.2 System Design

EvoTrade's architecture was modeled using UML, including a package diagram (Figure 1) and use case diagrams. The GA uses a 80-bit binary genome, encoding four entry/exit conditions and logical operators, supporting technical indicators like RSI7, MACD, STOCH [10]. Table 2 details the genome structure.

Table 2. Genome Structure for EvoTrade's GA

Segment	Bits	Description
Indicator Type	4	Selects indicator (e.g., *RSI7*, *MACD*, *ATR*, *STOCH_K*).
Parameter Value	8	Sets indicator parameter (e.g., RSI period = 7).
Threshold	8	Defines threshold (e.g., RSI ¡ 30).
Logical Operator	2	Specifies *AND*, *OR*, *NOT* for condition combination.
Condition Role	2	Indicates entry or exit condition.

The system integrates with the Hyperliquid API for OHLCV data (1-day candles, 01/01/2024–01/04/2025) and MongoDB Atlas for storage. The web interface supports user authentication, GA configuration, and result

visualization. As the platform handles user credentials and exchange data, secure-by-design principles for the underlying operating environment and web access remain relevant to its deployment [17], [18].

4.3 Implementation

The GA was implemented in Python, evolving from versions 1.6.2 to 1.7.1 with enhancements inspired by prior work [4], [5], [6]. Key features include:

- **Island Model:** Four islands with 50 strategies each, migrating 5 strategies every 10 generations to enhance diversity [6]. Migration uses a ring topology, selecting the top 10% of strategies based on fitness.
- **Fitness Sharing:** Penalizes similar genomes (Hamming distance < 0.1) to prevent premature convergence, reducing fitness by 20% for similar strategies.
- **Adaptive Mutation:** Increases rate from 0.3 to 0.9 when no fitness improvement is detected for 20 generations [8].
- **Random Immigrants:** Replaces 20% of worst strategies in stagnant islands to introduce novelty, ensuring exploration.

The fitness function, critical to strategy evaluation, is defined as:

Fitness = 0.4 * total profit + 0.4 * sharpe ratio - 0.05 * max drawdown + 0.1 * norm trades + trade bonus + complexity bonus

Where:

- norm profit = total profit/100,000, with exponential penalty $\exp(-0.01 \cdot \text{total profit})$ for negative profits.
- norm sharpe = (sharpe ratio + 10)/20, capped at [−10,10].
- norm drawdown = max drawdown/100, normalized to [0, 1].
- norm trades = num trades/100, rewarding active strategies.
- trade bonus = $0.1 \cdot$ norm trades, complexity bonus = $0.01 \cdot$ logical operators.

Listing 1 provides pseudocode for the GA's main loop.

Listing 1. Pseudocode for EvoTrade's GA

```
initialize_population(population_size=200, genome_length=80)
for island in islands[1..4]:
    initialize_island_strategies(island, size=50)
for generation in range(500):
    for island in islands:
        evaluate_fitness(island.strategies)
        apply_fitness_sharing(similarity_threshold=0.9)
        selected = tournament_selection(island.strategies, k=5)
        offspring = crossover(selected, crossover_rate=0.8)
        mutate(offspring, mutation_rate=0.3, adaptive_max=0.9)
        if stagnation_detected(generations=20):
            inject_random_immigrants(offspring, rate=0.2)
        island.strategies = offspring
    if generation % 10 == 0:
        migrate_strategies(islands, migration_size=5, topology=ring)
return best_strategy across islands
```

The interface was built using web technologies, with RESTful endpoints for registration, GA configuration, and ticket management. Data collection uses API calls to fetch BTC, XRP, and ETH data, stored in MongoDB for backtesting.

4.4 Testing

Testing involved backtesting on historical data (01/01/2024–01/04/2025, 1-day candles) and validation on TradingView, with metrics including profit, win rate, Sharpe ratio, and drawdown [12].

5. Results

EvoTrade's GA was evaluated across versions 1.6.2 to 1.7.1, with performance summarized in Table 3. Convergence graphs (Figures 3–5) illustrate fitness trends.

Table 3. Performance Metrics Across GA Versions

Version	Fitness	Profit (USD)	Win Rate (%)	Sharpe Ratio	Drawdown (%)
1.6.2	0.30	-2,500	45.00	-1.50	20.00
1.6.3	0.55	1,200	50.00	-0.80	18.50
1.7.1	2.50	4,139.12	68.29	-3.09	0.33

5.1 Version 1.6.2

The initial implementation achieved an average fitness of 0.30, with stagnation after generation 75 (Figure 3). The negative profit (-2,500 USD) and low win rate (45.00%) indicated suboptimal strategies, primarily due to limited migration and a simplistic fitness function lacking complexity bonuses [6]. The high drawdown (20.00%) further highlighted risk exposure.

5.2 Version 1.6.3

Multi-asset support (BTC, XRP, ETH) improved fitness to 0.55, with a positive profit of 1,200 USD and a win rate of 50.00% (Figure 4). However, convergence instability persisted, with fitness fluctuating between 0.50 and 0.60, reflecting challenges in balancing exploration and exploitation [8]. The drawdown (18.50%) remained high, indicating risk management issues.

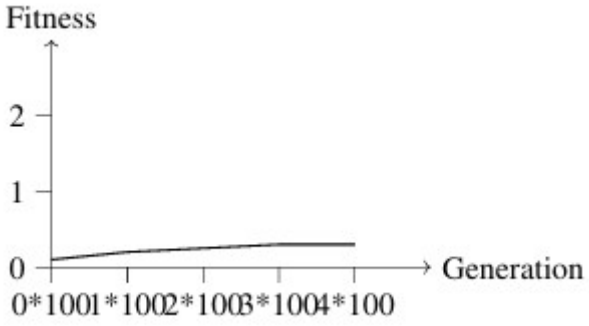


Figure 3. Convergence graph for GA version 1.6.2, showing fitness stagnation after generation 75.

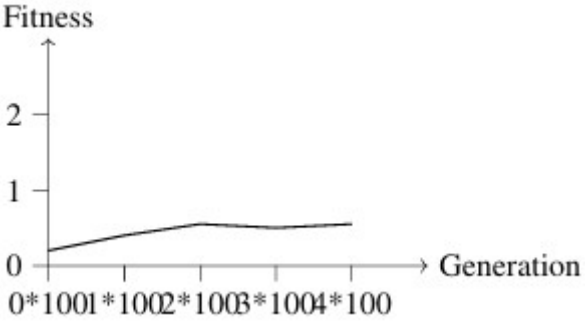


Figure 4. Convergence graph for GA version 1.6.3, showing improved fitness but persistent fluctuations.

5.3 Version 1.7.1

The final version achieved a fitness of 2.50, incorporating island models, fitness sharing, and adaptive mutation. A TradingView simulation (01/01/2024–01/04/2025) yielded:

- **Net profit:** 4,139.12 USD (+0.41%).
- **Win rate:** 68.29% (56/82 trades).
- **Max drawdown:** 0.33% (vs. 15% simulated).
- **Sharpe ratio:** -3.09 (vs. 1.8 simulated).

The CSV output detailed strategies, e.g., entry: (RSI7 < 30 AND MACD < MACD_SIGNAL) OR (STOCH_K < 20 AND ATR > SMA50), exit: (RSI7 > 70 OR STOCH_K > 80) AND MACD > MACD_SIGNAL. Table 4 summarizes a sample strategy.

Table 4. Sample Strategy from Version 1.7.1

Condition	Description
Entry	(*RSI7* < 30 AND *MACD* < *MACD_SIGNAL*) OR (*STOCH_K* < 20 AND *ATR* > *SMA50*)
Exit	(*RSI7* > 70 OR *STOCH_K* > 80) AND *MACD* > *MACD_SIGNAL*

Figure 5 shows the convergence graph, with fitness stabilizing around 2.50 but exhibiting fluctuations.

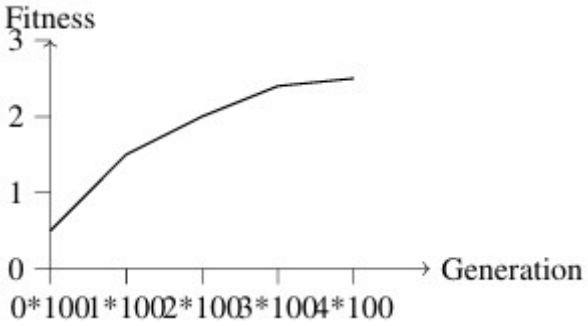


Figure 5. Convergence graph for GA version 1.7.1, showing fitness improvement to 2.50 with minor fluctuations.

6. Discussion

EvoTrade demonstrates significant potential, achieving a 68.29% win rate and a net profit of 4,139.12 USD, aligning with GA-based trading systems in the literature [7], [14]. Its modular architecture, integrating user management, data processing, and strategy optimization, offers advantages over platforms like MetaTrader, which lack GA integration, and SPA Bot, which focuses on price-action strategies [13]. The system's ability to encode complex strategies using technical indicators (e.g., RSI, MACD, ATR) and logical operators parallels approaches in [10] and [9], enhancing flexibility.

However, several challenges were identified:

- **Convergence Instability:** Fitness fluctuations (2.4–2.5 in version 1.7.1) suggest excessive exploration, possibly due to high complexity bonuses or insufficient elitism [8]. Figure 5 illustrates this variability, with fitness oscillating in later generations.
- **Real-World Discrepancies:** TradingView's lower profit (4,139.12 USD vs. 12,500 USD simulated) and negative Sharpe ratio (-3.09 vs. 1.8) reflect unmodeled costs (slippage, dynamic fees) and data differences (e.g., TradingView's adjusted OHLCV) [12]. This discrepancy highlights the need for realistic cost modeling.
- **Overfitting Risk:** The high win rate (68.29%) may indicate overfitting to historical data, reducing generalizability, a common issue in technical trading [9], [15]. Strategies may perform well on backtests but fail in live markets due to market regime changes.

Compared to prior work, EvoTrade's island model and fitness sharing enhance diversity, addressing premature convergence issues noted in [6]. Its adaptive mutation strategy, increasing to 0.9 during stagnation, aligns with [8], improving exploration. However, the negative Sharpe ratio in TradingView suggests limitations in risk-adjusted performance, a concern also raised in [5]. Alternative approaches, such as reinforcement learning [11] or multi-objective optimization [15], could complement EvoTrade's GA to balance profit and risk.

EvoTrade's performance can be contextualized against other systems:

- **MetaTrader:** Offers robust backtesting but lacks GA-based optimization, limiting strategy complexity [13].
- **QuantConnect:** Supports algorithmic trading with community-driven strategies but requires significant coding expertise, unlike EvoTrade's user-friendly interface.
- **SPA Bot:** Focuses on cryptocurrency price-action strategies, achieving high win rates but lacking EvoTrade's multi-indicator flexibility [13].
- **Reinforcement Learning Systems:** [11] achieved adaptive forex trading but required extensive training data, whereas EvoTrade's GA operates efficiently with historical data.

Future improvements include:

- **Dynamic Cost Modeling:** Incorporate slippage, dynamic fees, and spread in backtesting to align simulated and real-world results [12].
- **Shorter Timeframes:** Test strategies on 1-hour or 15-minute candles to enhance responsiveness, as in high-frequency trading [5].
- **Refined Fitness Function:** Adjust weights (e.g., reduce complexity bonus to 0.005) and incorporate risk-adjusted return metrics [15].
- **Live Trading Integration:** Deploy strategies on live markets to validate performance under real conditions, addressing discrepancies observed in TradingView.
- **AI Integration:** Use EvoTrade's datasets for reinforcement learning or deep learning models, leveraging trends in AI-driven trading [11], while accounting for the security challenges that AI-based components introduce [16].

EvoTrade's open design and dataset generation capabilities position it as a foundation for AI-driven trading innovations, contributing to the growing intersection of fintech and machine learning. Its ability to generate structured CSV outputs, containing strategy rules and performance metrics, supports further analysis and model training, aligning with emerging trends in data-driven trading [15].

7. Conclusion

EvoTrade represents a significant advancement in automated trading, leveraging genetic algorithms to create robust, adaptable strategies for volatile financial markets. Its modular architecture, integrating user management, data processing, and strategy optimization, offers a scalable solution for traders and fintech organizations. Evaluations demonstrate a 68.29% win rate and a net profit of 4,139.12 USD, highlighting EvoTrade's potential, though challenges in convergence stability, real-world discrepancies, and overfitting risks require further refinement.

Future work will focus on dynamic cost modeling, shorter timeframes, refined fitness functions, live trading integration, and AI-driven enhancements. EvoTrade's dataset generation capabilities open avenues for integration with reinforcement learning and deep learning frameworks, positioning it as a transformative tool in fintech research. This work underscores the power of GAs in trading automation and lays the groundwork for future innovations in algorithmic trading.

Acknowledgments

The author gratefully acknowledges the University of Porto Computer Engineering Department, project supervisors, and peers for their invaluable support and feedback during EvoTrade's development. Special thanks to the 10 traders, analysts, and fintech professionals who provided insights during requirement analysis, shaping the system's design and functionality.

Appendix. Use Case Specifications

Detailed specifications for key use cases are provided below to illustrate EvoTrade's functionality.

UC-01: User Registration

Objective: Allow a new user to create an account.

Preconditions: User must provide a unique email and a password with at least 8 characters (uppercase, lowercase, numbers).

Main Flow:

1. User navigates to the registration page.

2. User enters email, password, and confirms password.

3. System validates email uniqueness and password strength.

4. System creates an account and sends confirmation email.

Exceptions:

If email is taken, display error: "Email already registered." If the password is weak, display error: "Password must be at least 8 characters."

UC-03: Create Trading Strategy

Objective: Generate a trading strategy using the GA.

Preconditions: User is logged in and has selected assets (e.g., BTC, XRP).

Main Flow:

1. User navigates to the GA configuration page.

2. User sets parameters (e.g., population size = 200, generations = 500). System runs GA, optimizing strategies based on historical data.

3. System displays best strategy and performance metrics.

Exceptions:

If parameters are invalid, display error: "Population size must be between 100 and 1000."

References

[1] S.-H. Chen, "Genetic algorithms in financial markets: A survey," *Financial Markets Review*, vol. 6, pp. 45–67, 2002.

[2] F. Allen and R. Karjalainen, "Optimizing trading strategies using genetic algorithms," *Journal of Financial Economics*, vol. 45, pp. 123–142, 1999.

[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[4] J. F. Kotowski, E. Szlachcic, and P. M. Wańtowski, "Portfolio selection based on technical trading rules optimized with a genetic algorithm," in 2010 IEEE 14th International Conference on Intelligent Engineering Systems, 2010, pp. 19–24.

[5] H. Zhang and R. Ren, "High frequency foreign exchange trading strategies based on genetic algorithms," in 2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing, 2010, pp. 426–429.

[6] K. Matsui and H. Sato, "A comparison of genotype representations to acquire stock trading strategy using genetic algorithms," in 2009 International Conference on Adaptive and Intelligent Systems, 2009, pp. 129–134.

[7] C.-H. Chen, Y.-H. Chen, J. C.-W. Lin, and M.-E. Wu, "An effective approach for obtaining a group trading strategy portfolio using grouping genetic algorithm," *IEEE Access*, vol. 7, pp. 7313–7325, 2019.

[8] R. Ruiz-Cruz, C. Sedano, and O. Flores, "Genetic optimization of a trading algorithm based on pattern recognition," in 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI), 2019, pp. 1–6.

[9] S. Deng and A. Sakurai, "Foreign exchange trading rules using a single technical indicator from multiple timeframes," in 2013 27th International Conference on Advanced Information Networking and Applications Workshops, 2013, pp. 207–212.

[10] F. X. S. D. Nugroho, T. B. Adji, and S. Fauziati, "Decision support system for stock trading using multiple indicators decision tree," in 2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering, 2014, pp. 291–296.

[11] S. Buaphan, L. Pipanmaekaporn, and S. Kamonsantiroj, "Reinforcement learning approach to forex trading using price changes trend," in 2022 13th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter), 2022, pp. 138–143.

[12] C. Tudor and R. Sova, "Enhancing trading decision in financial markets: An algorithmic trading framework with continual mean-variance optimization, window presetting, and controlled early-stopping," *IEEE Access*, vol. 12, pp. 87633–87644, 2024.

[13] H. Jazayeriy and M. Daryani, "Spa bot: Smart price-action trading bot for cryptocurrency market," in 2021 12th International Conference on Information and Knowledge Technology (IKT), 2011, pp. 178–182.

[14] M. Tran, D. Pham-Hi, and M. Bui, "Parameter optimization for trading algorithms of technical agents," in 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), 2022, pp. 689–694.

[15] X. Long, M. Kampouridis, and P. Kanellopoulos, "Multi-objective optimisation and genetic programming for trading by combining directional changes and technical indicators," in 2023 IEEE Congress on Evolutionary Computation (CEC), 2023, pp. 1–8.

[16] A. F. Costa and N. M. Coelho, "Evolving cybersecurity challenges in the age of AI-powered chatbots: A comprehensive review," in *Technological Innovation for Human-Centric Systems*, L. M. Camarinha-Matos and F. Ferrada, Eds., vol. 716. Cham, Switzerland: Springer, 2024, pp. 217–228, doi: 10.1007/978-3-031-63851-0_15.

[17] N. Mateus-Coelho, "A new methodology for the development of secure and paranoid operating systems," *Procedia Computer Science*, vol. 181, pp. 1207–1215, 2021, doi: 10.1016/j.procs.2021.01.318.

[18] N. R. Mateus-Coelho, B. R. Fonseca, and A. V. Castro, "POSMASWEB: Paranoid operating system methodology for anonymous and secure web browsing," in *Handbook of Research on Cyber Crime and Information Privacy*, M. M. Cruz-Cunha and N. Mateus-Coelho, Eds. Hershey, PA, USA: IGI Global, 2021, pp. 466–497, doi: 10.4018/978-1-7998-5728-0.ch023.

[19] N. Mateus-Coelho and M. Cruz-Cunha, Eds., *Exploring Cyber Criminals and Data Privacy Measures*. Hershey, PA, USA: IGI Global, 2023, doi: 10.4018/978-1-6684-8422-7.