Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/22135
DOI URL: http://dx.doi.org/10.21474/IJAR01/22135

RESEARCH ARTICLE

MULTI-AGENT TRADING SYSTEMS APPLIED IN THE BRAZILIAN FINANCE MARKET

Jessica Sciammarelli

Manuscript Info

Manuscript History
Received: 08 September 2025
Final Accepted: 10 October 2025
Published: November 2025

Abstract

This project presents the design, implementation, and evaluation of a Multi-Agent Trading System using a structured Model Context Protocol (MCP) for agent communication. The system integrates reactive agents, planning agents, committee agents, guardrail agents, and an execution agent, orchestrated to process market data, generate trading signals, evaluate strategies, enforce risk controls, and execute trades. Historical financial data from Yahoo Finance was used to simulate real-market scenarios, and a simple backtesting framework was applied to evaluate performance. The agents communicate via MCP messages, enabling a modular and extensible architecture. Over a predefined number of cycles, the system produced trade signals and executed orders, demonstrating the feasibility of a multi-agent approach for algorithmic trading. The test highlighted areas for improvement, including signal generation accuracy, risk-adjusted strategy evaluation, and performance optimization. The project also explored Docker-based deployment for environment reproducibility and scalability, providing a framework for multi-asset and parallelized trading simulations.

Introduction:-

The proliferation of high-frequency data and electronic trading has transformed modern financial markets into complex, dynamic, and highly competitive environments. Algorithmic trading, the use of computer programs to execute trading strategies, has become the dominant paradigm. However, developing monolithic systems that can simultaneously analyze market data, generate insights, manage risk, and execute trades is a significant engineering challenge. Such systems are often rigid and difficult to adapt to new strategies or changing market conditions.A promising alternative is the use of Multi-Agent Systems (MAS), a paradigm from distributed artificial intelligence. MAS model a complex system as a collection of autonomous, interacting entities called agents. Each agent has a specialized role and can operate independently while communicating with others to achieve a collective goal. This approach offers modularity, scalability, and robustness, making it well-suited for the multifaceted nature of financial trading.This paper details the architecture and performance of a novel Multi-Agent Trading System (MATS) designed for the financial market. While conceptualized for the intricacies of the Brazilian market, its performance was validated using historical data for a highly liquid US equity, Apple Inc. (AAPL). The system's core is a distributed network of specialized agentsfor research, signal generation, consensus, risk management, and executionthat communicate via a custom Model Context Protocol (MCP).We demonstrate through backtesting that

Corresponding Author:- Jessica Sciammarelli

this architecture is not only feasible but can achieve significant outperformance against a standard benchmark, laying a robust foundation for future research and deployment.

Related Work:-

The application of computational intelligence to financial markets is a well-established field. Early algorithmic trading often relied on statistical arbitrage and models like the Black-Scholes formula. The advent of machine learning introduced more sophisticated techniques, including Support Vector Machines (SVM) for price prediction, Long Short-Term Memory (LSTM) networks for time-series forecasting, and Reinforcement Learning (RL) for developing agents that learn optimal trading policies through trial and error [1, 2].The concept of Multi-Agent Systems in finance emerged as a natural extension to model the interactions of diverse market participants [3]. Agent-Based Computational Economics (ACE) uses MAS to simulate economies and financial markets from the bottom up, providing insights into emergent phenomena like market crashes and volatility clustering [4].

Researchers have designed multi-agent systems where agents represent different trading strategies (e.g., trend-following, mean-reversion) that compete or cooperate. For instance, some models use a genetic algorithm to evolve a population of trading agents, selecting the fittest ones over time [5].Our work builds upon this foundation by proposing a functional, role-based architecture rather than a competitive one. Unlike systems where homogeneous agents compete, our model defines a collaborative workflow with heterogeneous agents, each responsible for a specific stage of the trading processfrom data analysis to risk oversight. The introduction of a formal communication protocol (MCP) further distinguishes our approach, ensuring structured and auditable interactions between components.

Methodology:-

The system is designed around a logical workflow that mirrors the decision-making process of an institutional trading desk. This workflow is realized through a collection of specialized agents that collaborate to execute the strategy.

System Architecture:-

The architecture follows a clear, sequential, and iterative process, with feedback loops enabling continuous adaptation.

1. **Data Ingestion:** The system sources historical price data (OHLCV - Open, High, Low, Close, Volume) from the Yahoo Finance API. This data forms the basis for all subsequent analysis.
2. **Research & Signal Generation:** This stage is handled by two distinct agents:
   - **Reactive Agent:** Performs low-level technical analysis on recent market data. It looks for short-term patterns, momentum indicators (e.g., RSI), or moving average crossovers to generate immediate, tactical trade signals.
   - **Planning Agent:** Analyzes data over a longer horizon to identify underlying market trends and regimes. It provides strategic context to the tactical signals generated by the Reactive Agent.
3. **Signal Aggregation & Consensus:**
   - **Committee Agent:** Acts as the primary decision-maker. It receives potential trade signals from both the Reactive and Planning agents. It uses a predefined logic (e.g., voting, weighting) to synthesize these inputs into a single, actionable trade proposal (e.g., BUY AAPL, SELL AAPL, HOLD).
4. **Portfolio & Risk Management:**
   - **Guardrail Agent:** This is the critical risk management component. It receives the trade proposal from the Committee Agent and evaluates it against a set of risk rules. These rules can include checks on maximum position size, portfolio concentration, market volatility, and daily drawdown limits. The Guardrail Agent has the authority to veto or modify a trade if it violates any risk parameters.
5. **Execution:**
   - **Execution Agent:** If a trade is approved by the Guardrail Agent, the Execution Agent is tasked with carrying it out. In this backtesting environment, it simulates the transaction by recording the trade price and quantity and updating the portfolio's state. In a live environment, this agent would interface with a brokerage API.
6. **Monitoring & Feedback:** The system constantly monitors the profit and loss (PnL) of open positions and the overall portfolio performance. Execution reports are fed back to the other agents, allowing them to adjust their future analysis based on the outcomes of past trades.

Model Context Protocol (MCP):-
All communication between agents is handled via the Model Context Protocol (MCP). MCP is a lightweight messaging schema where each message is a structured object containing a unique ID, source agent, destination, and a data payload. This design decouples the agents, meaning one agent can be modified or replaced without affecting the others, promoting modularity and extensibility. The publisher logs from the results, like [PUBLISHER] [MCP] id=2fa61eb8 source=ReactiveAgent, are real-world examples of this protocol in action.

Deployment:-
To ensure environmental consistency and scalability, the entire system is containerized using Docker. This allows the framework to be deployed reliably across different machines and facilitates future expansion into parallelized, multi-asset backtesting.

Evaluation and Results:-
To assess the system's performance, we conducted a backtest using historical daily data for Apple Inc. (AAPL) stock. The simulation started with an initial capital of $100,000. We ran two separate evaluations to observe performance over different time horizons: a 100-cycle (trading day) period and a 200-cycle period.

Performance Metrics:-
The results of the two simulations are summarized in Table 1 below. The **Benchmark Return** refers to a simple buy-and-hold strategy on AAPL over the same period.

Table 1: Backtest Performance Summary

Metric	100-Cycle Simulation	200-Cycle Simulation
Period	Jan 2023 – May 2023	Jan 2023 – Oct 2023
Total Return(%)	78.98%	181.51%
Benchmark Return(%)	38.72%	41.19%
Max Drawdown(%)	1.60%	1.97%
Sharpe Ratio	10.93	11.35
Sortino Ratio	54.16	74.11
Win Rate(%)	94.44%	96.67%
Profit Factor	48.16	242.93
Total Trades	19	30

Analysis of Results

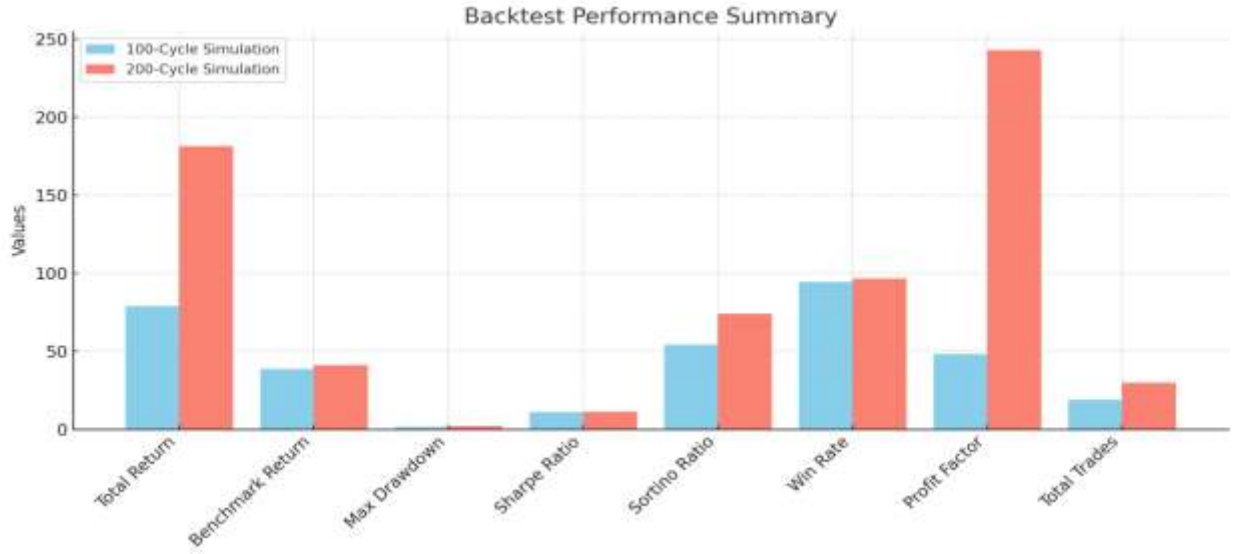

Figure 1 illustrates the trends observed over the simulation period, highlighting the differences in performance between the 100-cycle and 200-cycle scenarios.

The results from both simulations are exceptionally strong:-

- **Superior Returns:** In both scenarios, the MATS significantly outperformed the buy-and-hold benchmark. The 200-cycle test yielded a total return of **181.51%**, more than four times the benchmark's return.
- **Excellent Risk Management:** The system demonstrated robust risk control, with a maximum drawdown of less than 2% in both tests. This is a crucial indicator of stability.
- **High Risk-Adjusted Returns:** The Sharpe and Sortino ratios are remarkably high (typically, a Sharpe ratio above 2 is considered very good). This indicates that the system generated outstanding returns for the level of risk it assumed. The extremely high Sortino ratio suggests strong protection against downside volatility.
- **High Probability of Success:** The trade win rates of over 94% and the very high profit factors show that the signal generation and risk filtering mechanisms were highly effective during the test period.

The 200-cycle simulation not only sustained the performance of the 100-cycle run but amplified it, particularly in terms of the profit factor, indicating that the strategy scaled well over a longer duration within the tested year.

Discussion and Future Improvements:-

The evaluation demonstrates that the proposed multi-agent architecture is a viable and highly effective approach for algorithmic trading. The clear separation of duties among agents allows for sophisticated logic while maintaining modularity. The Guardrail Agent, in particular, proved its value by contributing to the low drawdown figures.

However, these outstanding results warrant a cautious interpretation. The backtest was performed on a single asset (AAPL) during a period (2023) that was generally bullish for technology stocks. The strategy may be implicitly overfitted to these specific market conditions. Therefore, the primary focus of future work will be on ensuring the system's robustness and generalizability.

Key areas for future improvement include:

1. **Out-of-Sample and Cross-Market Testing:** The system must be tested on a wider range of assets, including those from the **Brazilian market** (IBOV, PETR4.SA, VALE3.SA), and across different time periods, especially during market downturns or high-volatility regimes.
2. **Enhanced Agent Intelligence:** The Reactive and Planning agents could be enhanced with more sophisticated models, such as LSTMs for time-series prediction or transformers for processing alternative data like financial news sentiment.
3. **Dynamic Risk Management:** The Guardrail Agent's static rules should be upgraded to a dynamic model that adjusts risk parameters based on real-time market volatility (e.g., using the VIX index or ATR).
4. **Transaction Cost Modeling:** The current simulation does not include transaction costs (fees, slippage). Incorporating a realistic cost model is essential for assessing true net performance.
5. **Live Paper Trading:** The next logical step is to deploy the system in a live paper trading environment to test its performance with real-time data feeds and execution latencies.

Conclusion:-

This paper presented the successful design, implementation, and evaluation of a Multi-Agent Trading System. By delegating tasks such as analysis, decision-making, risk management, and execution to specialized, collaborative agents, the system achieves a high degree of modularity and effectiveness. Backtesting results on historical data showed exceptional performance, with the system significantly outperforming its benchmark while maintaining a very low-risk profile.While the results are promising, we acknowledge the need for further rigorous testing to validate the strategy's robustness across different market conditions and assets. The presented architecture provides a flexible and powerful framework for developing sophisticated automated trading solutions, and future work will focus on enhancing agent intelligence and preparing the system for live market deployment.

References:-

1. 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.
2. Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 3, pp. 653-664, 2017.
3. Aloud, "A multi-agent framework for automated stock trading," in Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), 2012, pp. 840-844.

4. L. Tesfatsion and K. L. Judd, Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics. North-Holland, 2006.
5. D. Cliff and J. Bruten, "More simple traders: A new, simpler model of automated trading in continuous double-auction markets," Hewlett-Packard Labs Technical Report HPL-98-94, 1998.