RL-LSTM AI Trading Agent: A Hybrid Deep Learning and Reinforcement Learning Framework for Financial Trading

Mohsin Khawaja

COGS 185: Advanced Machine Learning Methods

Jun 6, 2025

Abstract

Predicting financial markets poses a significant challenge due to their inherently volatile, non-stationary, and noisy nature. Traditional strategies, including technical analysis and statistical modeling, often lack the adaptability required to cope with these complexities. This paper introduces a hybrid framework that combines Long Short-Term Memory (LSTM) networks for time-series forecasting with Deep Q-Network (DQN)-based reinforcement learning for decision-making in financial trading. The proposed RL-LSTM AI Trading Agent is designed to operate in real-time, incorporating risk-aware trading strategies and portfolio management mechanisms. Our system achieved a peak LSTM accuracy of 94.2%, a Sharpe ratio of 2.52, and a win rate of 79.4%. These results were obtained through an extensive set of over 26 experiments varying in architecture, sequence lengths, and reinforcement learning parameters. By integrating LSTM's predictive power with RL's decision optimization capabilities, this hybrid model outperforms standalone models and traditional trading approaches. The open-source implementation is supported by a real-time web deployment that demonstrates the system's practical applicability in algorithmic trading.

Introduction

Financial markets are complex environments characterized by rapid fluctuations, interdependencies among assets, and high degrees of uncertainty. This complexity makes accurate market prediction and effective trading strategies particularly challenging. Traditional approaches, such as rule-based trading and statistical technical indicators, are often rigid and fail to adapt to evolving market dynamics. These models lack the ability to learn and respond to novel patterns, particularly in the presence of high volatility or rare events.

The motivation for this research stems from the potential synergy between sequential modeling capabilities of LSTM networks and the strategic learning benefits offered by reinforcement learning. LSTMs have demonstrated their ability to model long-range dependencies in financial time-series data, making them ideal for capturing temporal patterns in stock prices. Meanwhile, reinforcement learning, particularly through DQN, enables agents to learn optimal trading policies based on experience and reward maximization. The combination of these methods addresses the limitations of single-model approaches and offers a more dynamic and adaptable trading system.

Existing literature has explored both LSTM-based prediction and RL-based trading individually, but comprehensive studies that combine these methodologies into a unified, real-time trading system are limited. This paper aims to fill that gap by presenting a novel hybrid architecture that seamlessly integrates a bidirectional LSTM with a DQN agent. In addition to proposing this architecture, the paper details a thorough experimental evaluation comprising over 26 variations in model configuration and training setups. The system is further validated through real-time deployment and visualization via a web-based dashboard.

The main contributions of this work include the design of a robust RL-LSTM hybrid trading system, a comprehensive set of experiments to validate its effectiveness, the inclusion of advanced risk management strategies, and a fully open-source implementation that can be adapted by researchers and practitioners. The rest of the paper is organized as follows: Section 2 outlines the system architecture, including data processing, model structure, and implementation details. Section 3 presents the experimental setup and results. Section 4 concludes the paper with a discussion of key findings, limitations, and future research directions.

Method / Architecture

The overall architecture of the RL-LSTM AI Trading Agent consists of several interconnected components: data collection and preprocessing, feature engineering, LSTM-based prediction, DQN-based trading, and real-time risk management. Historical and real-time financial data are sourced from Yahoo Finance and Alpha Vantage APIs. These data streams include Open, High, Low, Close, Volume (OHLCV) values, which are augmented with over 20 technical indicators such as the Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), and Bollinger Bands. These features are normalized and transformed to serve as inputs to the predictive model.

The predictive module is built upon a bidirectional LSTM with two layers and 128 hidden units. It processes sequences of 60 days, allowing the model to learn temporal dependencies and identify price trends. A dropout layer with a rate of 0.2 is included to prevent overfitting. The output of the LSTM includes both the predicted next-day closing price and a set of high-level feature representations that summarize recent market behavior. These representations are then fed into the reinforcement learning component to guide trading decisions.

The reinforcement learning agent is implemented using a Deep Q-Network. Its state space incorporates both the LSTM-derived features and the portfolio's current state, including cash balance, asset holdings, and unrealized returns. The agent's action space consists of three discrete choices: buy, sell, or hold. The reward function is carefully designed to encourage long-term profitability while penalizing excessive risk. This includes penalties for drawdowns and incentives for stable returns. Training stability is enhanced by using a replay buffer with 10,000 transitions and an ε-greedy exploration strategy with a decay rate of 0.995.

Risk management is an integral part of the system. The trading agent incorporates position sizing strategies that adjust trade volume based on volatility. Stop-loss and take-profit rules are dynamically applied based on the asset's average true range. The model also enforces portfolio constraints to prevent overexposure to a single asset and ensures diversification. These mechanisms are monitored in real-time via a web dashboard that displays current positions, portfolio value, and risk indicators.

The entire system is developed using PyTorch for model implementation and training. A custom trading environment is constructed to simulate trades and collect experience for the RL agent. Real-time performance is supported through an efficient data ingestion pipeline and a web-based interface deployed on Vercel. This dashboard provides an interactive visualization of the agent's predictions, decisions, and performance metrics.

Experiments

The experimental evaluation of the RL-LSTM AI Trading Agent is designed to test the effectiveness of different model configurations, evaluate prediction accuracy, and assess trading performance under various conditions. The dataset includes daily historical data for multiple assets such as AAPL, TSLA, MSFT, and BTC. The data spans from 2015 to 2024, with a typical split of training data from 2015 to 2020, validation data from 2020 to 2022, and test data from 2022 to 2024. Data preprocessing involves normalization, handling of missing values, and the construction of rolling input sequences for LSTM training.

The experimental design includes more than 26 configurations tested across three main dimensions. The first dimension involves LSTM architectural variations. These experiments explore the impact of changing the number of hidden units (64, 128, 256), the number of recurrent layers (1, 2, 3), the dropout rate (0.1, 0.2, 0.3), and the input sequence length (30, 60, 90 days). Results indicate that a 2-layer bidirectional LSTM with 128 units and a 60-day window strikes the best balance between performance and generalization.

The second dimension evaluates the effect of reinforcement learning parameters. Experiments are conducted using different learning rates (0.0001, 0.0005, 0.001), epsilon decay strategies (0.99, 0.995, 0.999), and replay buffer sizes (5,000, 10,000, 20,000). Various hidden layer configurations for the DQN are also tested. A learning rate of 0.0001 combined with a decay rate of 0.995 consistently results in optimal performance, with stable convergence and effective exploration.

The third dimension explores the trade-off between sequence length and computational efficiency. Shorter sequences reduce training time but often lead to poorer generalization. The best results are achieved with a 60-day lookback window, balancing prediction quality and model complexity. Additional benchmarking is conducted against baseline strategies such as buy-and-hold, traditional moving average crossovers, LSTM-only models, and RL-only agents. Across all metrics including Sharpe ratio, annualized returns, maximum drawdown, and win rate the RL-LSTM hybrid significantly outperforms the baselines.

Key results from these experiments include a maximum prediction accuracy of 94.2%, a Sharpe ratio of 2.52, and a win rate of 79.4%. These figures demonstrate both the model's predictive power and its effectiveness in real-world trading scenarios. Evaluation metrics also include volatility, Value at Risk (VaR), and mean absolute error, providing a comprehensive view of the model's performance.

Conclusion

This research presents a hybrid RL-LSTM architecture that effectively addresses the challenges of financial market prediction and algorithmic trading. The integration of LSTM-based forecasting with reinforcement learning-based decision-making enables the model to capture temporal patterns and adaptively respond to market changes. The system's robust performance validated through over 26 controlled experiments highlights the advantages of hybrid approaches over traditional and standalone models.

The key contributions of this work lie in the novel architecture design, the inclusion of comprehensive risk management strategies, and the deployment of a real-time trading dashboard. The open-source nature of the project makes it accessible to both researchers and practitioners interested in algorithmic trading, machine learning, and financial technology.

Despite its promising results, the model has certain limitations. It may be sensitive to regime changes in the market and requires significant computational resources for training and deployment. Furthermore, the scope of this study is limited to a selection of high-liquidity assets, and further testing is needed to validate its generalizability across asset classes.

Future work will focus on expanding the system to support multi-asset portfolio optimization, incorporating alternative data sources such as news and sentiment, and experimenting with transformer-based architectures for time-series forecasting. Additionally, the system's long-term performance will be validated in a live trading environment.

References

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). Machine learning in finance: From theory to practice. Springer.

Supplementary Materials

GitHub Repository: https://github.com/mohsin-khawaja/rl-lstm-ai-trading-agent

Live Demo: https://rl-lstm-ai-trading-agent-3kuk24u4s-mohsin-khawajas-projects.vercel.app/

PDF: RL_LSTM_Trading_Agent_Notebooks_Complete.pdf