International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

Automated AI-Based Crypto Trading Bot Using LSTM and Reinforcement Learning

B. Bhavani[1], K. Sathvik Chandra[2], G. Saadan[3], G. Sathvik[4]
[1]*Associate Professor, Dept. ECE, MVSR Engg. College, Hyderabad, India*
[2,3,4]*Dept. Electronics and Communication Engg., MVSR Engg. College, Hyderabad, India*

ABSTRACT
The volatility and continuous nature of cryptocurrency markets pose formidable challenges for human traders, who remain susceptible to emotional decision-making and cognitive fatigue over extended trading periods. This paper introduces a multi-layered artificial intelligence trading framework designed to address these constraints through systematic, data-driven decision processes. The proposed architecture integrates three complementary intelligent modules: a Long Short-Term Memory (LSTM) network that processes historical price sequences to generate forward-looking price estimates; a Reinforcement Learning (RL) agent that functions as the central decision-making component, dynamically calibrating risk exposure in accordance with real-time market conditions; and a Natural Language Processing (NLP) module that derives quantified market sentiment from live financial news streams. Empirical evaluation conducted through simulated paper trading trials demonstrates that this integrated methodology yields measurably superior risk-adjusted performance relative to conventional rule-based trading systems. These findings affirm that coupling memory-augmented sequence modeling with adaptive strategy optimization represents a viable pathway toward more consistent and sustainable engagement with digital asset markets.
KEYWORDS
LSTM, Reinforcement Learning, Cryptocurrency, Sentiment Analysis, NLP, Algorithmic Trading.

1. INTRODUCTION

The global financial ecosystem is undergoing a profound transformation, driven in large part by the rapid emergence and widespread adoption of cryptocurrency markets. Unlike conventional equity or commodity exchanges operating within defined business hours, digital asset platforms function continuously across all time zones, subjecting traders to relentless price fluctuations with no natural recovery period. Price dislocations can be triggered at any moment by regulatory announcements, macroeconomic data releases, or commentary from influential market participants—often with little forewarning. For individual traders navigating this environment without automated assistance, the volume and velocity of actionable information far exceed what can be reliably processed in real time, particularly under the psychological pressure associated with high-stakes financial decisions.

Historically, rule-based automated trading bots have been deployed as a partial remedy to these challenges. Such systems operate on deterministic logic: a fixed set of conditional rules specifies precisely what action to execute under each anticipated market condition. While offering speed and mechanical consistency, these architectures are fundamentally brittle. Their performance degrades markedly when the underlying statistical behavior of the market shifts—a phenomenon referred to as a regime change—because the embedded rules have no mechanism for distinguishing a persistent new trend from a transient market anomaly. Without the capacity to update strategy in response to evolving conditions, such systems rapidly lose their utility in volatile cryptocurrency markets.

This paper addresses that gap by proposing a trading framework that is inherently adaptive. Rather than encoding a static decision ruleset, the system continuously refines its behavior through feedback received from the market environment. By integrating deep learning—specifically, sequence-aware LSTM networks—with reinforcement learning, the framework is equipped to identify meaningful temporal patterns in historical price data while simultaneously learning which sequences of actions yield the most favorable long-term outcomes. This design philosophy aims to

DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

replace the reactive, emotionally influenced responses typical of human traders with disciplined, evidence-based actions that remain consistent across the full duration of the trading cycle.

2. MOTIVATION AND PROBLEM STATEMENT

A. MOTIVATION

The foundational motivation for this research stems from a critical assessment of human cognitive limitations when applied to the demands of modern cryptocurrency trading. The most immediate challenge is one of endurance: digital asset markets operate without interruption, demanding a level of sustained vigilance that is physiologically unsustainable. The accumulating fatigue of prolonged trading sessions manifests as reduced analytical acuity, delayed signal recognition, and avoidable losses driven by impaired judgment.

Beyond endurance, the informational breadth of the cryptocurrency market presents a second major obstacle. A competent trader must simultaneously monitor price dynamics across multiple asset pairs, track breaking developments in global financial news, and interpret sentiment signals emanating from online communities—all in real time. This multidimensional processing challenge fundamentally exceeds human cognitive bandwidth. Artificial intelligence systems, by contrast, are well-suited to precisely this kind of high-throughput, low-latency data processing.

The core ambition of this work is to construct a system capable of surveying the full landscape of available market signals, isolating actionable information from noise, and executing on that information with a speed and precision that eliminates hesitation and cognitive bias. The ultimate objective is to transform what is typically a chaotic, emotionally taxing activity into a disciplined, data-governed process that scales effectively across a broad portfolio of digital assets.

B. LIMITATIONS OF EXISTING SYSTEMS

An examination of commercially available retail trading bots reveals a consistent set of architectural weaknesses constraining their viability in cryptocurrency markets:

Static Strategy Architecture: Most automated bots are built around fixed indicator-based logic. Strategies founded on static moving averages or threshold-triggered signals perform adequately during directional trending phases but deteriorate in sideways or transitional market conditions. Because these systems contain no mechanism for detecting regime changes, they continue applying obsolete strategies long after market dynamics have rendered them ineffective.

Absence of News Awareness: Technical price charts capture historical transaction data but provide no causal explanation for observed price movements. Conventional systems operate in informational isolation, lacking access to financial news feeds or social media monitoring. As a result, they are consistently slow to respond to catalytic events—such as regulatory actions or major institutional disclosures—that an informed human trader following live news would identify immediately.

Inflexible Risk Allocation: A critical flaw in many automated systems is the use of fixed position sizing, which allocates identical capital exposure to both low-risk and high-risk setups. Rational risk management requires that capital deployment be proportionate to the quality and confidence level of each trade opportunity.

Absence of a Learning Mechanism: Standard automated systems incorporate no feedback loop enabling them to learn from prior trading outcomes. A failing strategy executed on one occasion is liable to be repeated without modification, since there is no mechanism through which the system can update its parameters or beliefs based on observed results.

Lagging Signal Generation: Many conventional systems rely on lagging technical indicators that produce entry signals only after a substantial portion of the anticipated price movement has already occurred. This temporal offset results in suboptimal entry prices and systematically diminished profit potential.

Vulnerability to Extreme Market Events: Rule-based systems are particularly exposed during flash crash scenarios, where price dislocations materialize with extraordinary speed. Without dynamically calibrated protective mechanisms, such systems may execute trades at severely

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

disadvantageous prices or amplify losses by failing to account for extreme market outliers.

Extreme Vulnerability to "Flash Crashes" Because basic bots are just following a script, they can get caught in a feedback loop during a flash crash. If they don't have smart, adaptive stop-losses, they can end up selling at the absolute bottom or buying into a "falling knife" because their math doesn't account for extreme, abnormal market outliers.

3.SYSTEM ARCHITECTURE

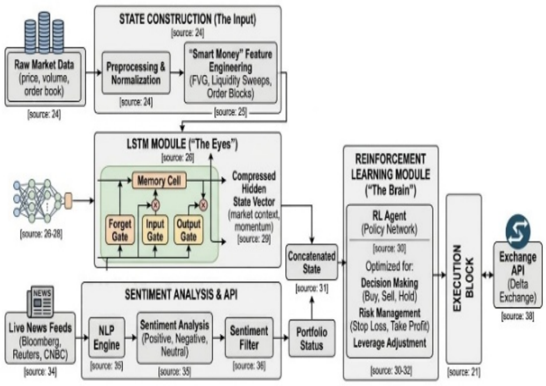

Figure 1. Overview of the Automated AI Trading Bot Architecture.

Figure 1.Overview of the Automated AI Trading Bot Architecture

The proposed framework is best understood as an integrated ensemble of three functionally distinct and cooperating modules—analogous to the sensory, cognitive, and motor functions of an experienced decision-maker operating under time-constrained conditions

State Construction Module (Sensory Input): This module constitutes the framework's perceptual interface with the market. Its function is to aggregate, clean, and structure heterogeneous raw data from exchange feeds—including price tick records, transaction volumes, and order book depth transforming it into a coherent, actionable representation of the current market environment.

Reinforcement Learning Agent (Decision Core): Upon receiving the structured market state, the RL agent evaluates the available action space—comprising buy, sell, and hold decisions—in the context of the current portfolio status and risk exposure. It operates with the explicit objective of maximizing long-term cumulative reward rather than single-step gains.

Execution Block (Trade Implementation): This component translates the RL agent's decisions into live market orders. It interfaces directly with exchange APIs to submit orders with minimal latency and manages the placement of protective instruments—stop-loss and take-profit orders—in accordance with dynamically computed risk parameters.

A.STATECONSTRUCTION(THE INPUT)

The State Construction module is the first and most critical stage of the pipeline. Before the system can act, it must achieve an accurate characterization of the current trading environment. The module acquires raw data streams from exchange interfaces and subjects them to a rigorous cleaning and normalization process. The resulting dataset is subsequently enriched through advanced feature engineering operations drawn from Smart Money Concepts (SMC)—a framework for interpreting institutional-level market dynamics:

Fair Value Gap (FVG) Detection: The system identifies price imbalances arising when market activity moves with sufficient velocity to create a gap between successive candlestick bodies. Such regions attract subsequent price action as the market rebalances the imbalance.

Liquidity Sweep Identification: The module tracks instances where price temporarily penetrates levels at which stop orders are concentrated, triggering those orders before reversing direction—a pattern characteristic of institutional order flow.

Order Block Recognition: Price levels associated with significant prior institutional buying or selling activity are identified and catalogued. These levels function as durable support and

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

resistance zones, providing higher-quality structural reference points than those derived from conventional moving-average methods.

The enriched feature set is then passed to a Long Short-Term Memory (LSTM) network, which processes the data as a time-ordered sequence. The LSTM's gating mechanism enables it to preserve relevant contextual information across extended temporal horizons while selectively discarding noise, yielding a compact, high-fidelity state representation that encapsulates both recent and historically significant market context.

B. LSTM MODULE("THE EYES")

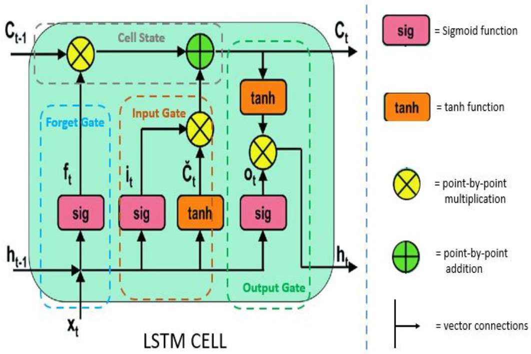

Figure 2. LSTM Memory cell

The LSTM module serves as the framework's temporal reasoning engine, enabling the system to interpret current price activity in the context of prior market behavior. Whereas conventional indicator-based systems evaluate each price observation in isolation, the LSTM processes market data as a continuous, ordered sequence, facilitating the detection of patterns that unfold across multiple time periods and would otherwise go unrecognized

The LSTM architecture achieves its temporal reasoning capability through three gating mechanisms embedded within each memory cell:

Forget Gate: Governs the selective deletion of stored information that has diminished in predictive relevance. By continuously purging outdated context, this gate keeps the cell state focused on the current market environment rather than allowing historical anomalies to distort present-state assessment.

Input Gate: Controls the integration of newly arriving information into the cell state. Rather than incorporating all incoming data indiscriminately, this gate assigns differential importance to new observations, selectively updating the stored representation with the most informative features.

Output Gate: Synthesizes the current cell state into a Hidden State vector transmitted to the RL agent. This vector encapsulates not only the current price level but also the momentum and contextual dynamics underlying recent market activity, providing the RL agent with a rich perceptual foundation for decision-making

C. REINFORCEMENT LEARNING MODULE ("THE BRAIN")

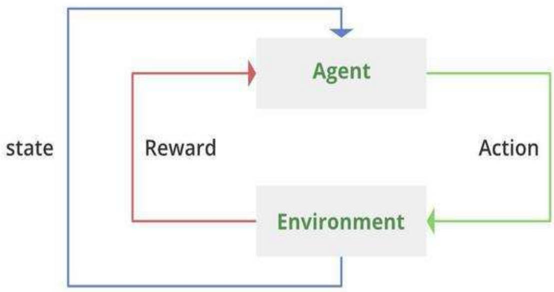

Figure 3.Reinforcement Learning

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

The Reinforcement Learning module constitutes the strategic core of the framework. While the LSTM provides an analytically rich characterization of the market environment, the RL agent is responsible for translating that characterization into concrete trading decisions.

The agent operates on a composite state formed by concatenating the LSTM hidden state with the current portfolio status—comprising available capital, open positions, and existing risk exposure. This composite input enables the agent to reason about both market conditions and its own financial position simultaneously. From this augmented state, the agent selects among three possible actions:

Trade Execution (Buy, Sell, or Hold): The agent evaluates the expected long-term consequences of each possible action and may defer entry to secure a more favorable price, reflecting a degree of strategic patience beyond the reach of purely reactive systems.

Dynamic Risk Parameter Configuration: Context-sensitive Stop Loss (SL) and Take Profit (TP) levels are computed relative to prevailing market volatility. Parameters are tightened during elevated-uncertainty conditions to preserve capital and relaxed during stable conditions to allow profitable positions greater latitude.

Adaptive Leverage Adjustment: Leverage is modulated in proportion to the agent's confidence in the current setup, increasing exposure when evidence strongly supports a directional move and scaling back when the signal environment is ambiguous.

The agent is trained through an iterative reward-and-penalty mechanism. Profitable outcomes and successful capital preservation generate positive reward signals that reinforce the associated decision patterns. Unnecessary risk-taking and suboptimal order placement produce penalty signals that discourage recurrence. Over successive training episodes, this feedback process shapes the agent's policy toward a balance between growth objectives and capital preservation imperatives

4. SENTIMENT ANALYSIS AND API INTEGRATION

A. SENTIMENT ANALYSIS: THE MARKET'S "GUT CHECK"

Figure 4.Sentiment Analysis

A persistent shortcoming of purely technical trading systems is their inability to account for the qualitative, news-driven forces that frequently dominate short-term cryptocurrency price dynamics. A technically sound setup can be rapidly invalidated by an unanticipated headline, while a technically ambiguous pattern may precede a substantial move only comprehensible in the context of prevailing news sentiment. The Sentiment Analysis module addresses this gap by providing the framework with a mechanism for interpreting and acting upon real-world informational signals.

Live Financial News Aggregation: The module maintains persistent connections to high-frequency financial news sources, including Bloomberg, Reuters, and CNBC. This enables detection of significant informational events—regulatory announcements, institutional investment disclosures, or macroeconomic data releases—as they emerge, typically in advance of their full reflection in the price chart.

NLP-Based Sentiment Classification: Raw news text is processed by a Natural Language Processing engine that tokenizes, normalizes, and classifies each article or headline as positive (bullish), negative (bearish), or neutral. A scalar sentiment score captures both the direction and intensity of prevailing market opinion.

Sentiment-Conditioned Trade Filtering: The resulting score is integrated into the RL agent's decision process as a conditional filter. When aggregate sentiment falls below a predefined negative threshold—indicating widespread bearish consensus or active market distress—the filter suppresses execution of long-side signals regardless of concurrent technical indications. This prevents the framework from acting contrary to prevailing market consensus during periods of genuine fundamental deterioration.

5. HARDWARE AND SOFTWARE REQUIREMENTS

Constructing and deploying a system of this complexity requires a carefully selected technology stack governed by two competing priorities: sufficient computational capacity to support deep learning training and real-time inference, and sufficient efficiency to operate without dedicated high-performance computing infrastructure.

Software Stack: The implementation is built on Python 3.10, which provides a mature ecosystem for scientific computing and machine learning. Neural network construction and training leverage TensorFlow and Keras, while data acquisition, preprocessing, and feature engineering are handled by NumPy and Pandas. Scikit-learn contributes utility functions for normalization, train-test partitioning, and preliminary model evaluation.

External APIs and Data Connectivity: Real-time market connectivity is maintained through the Delta Exchange API, providing live price data streaming and programmatic order placement. Sentiment data is sourced through REST-based financial news APIs delivering a continuous feed of structured content to the NLP processing pipeline.

Hardware Specifications: A notable characteristic of this implementation is that all development, training, and real-time execution were conducted on a standard consumer laptop. This confirms that the computational requirements of the proposed architecture, while non-trivial, fall well within the capability of modern commodity hardware when appropriate software optimization is applied. The system is thus accessible to individual developers without cloud computing subscriptions or dedicated GPU infrastructure.

6. RESULTS AND DISCUSSION

To assess the viability of the proposed framework under realistic market conditions, the system was evaluated through an extended series of simulated paper trading trials. This methodology permitted direct observation of system behavior against authentic price feeds and contemporaneous news cycles, while eliminating the financial risk associated with a funded live account.

A. Directional Prediction and Execution Efficiency:

The hybrid LSTM model demonstrated markedly superior directional forecasting performance relative to a standard Recurrent Neural Network (RNN) baseline. The improvement was most pronounced in the identification of short-duration price patterns—a class of features that conventional architectures frequently fail to capture due to their limited temporal receptive fields. Furthermore, the automated execution infrastructure demonstrated a substantial speed advantage over manual trading methodologies. The latency between signal confirmation and order submission was effectively negligible, ensuring fills at or near intended entry prices and mitigating the price slippage inherent in manually executed trades.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

B. Risk-Adjusted Performance and Volatility Management:

The Reinforcement Learning (RL) agent's conduct during periods of acute market volatility represented the most consequential finding of the evaluation. Rather than amplifying losses by maintaining or increasing position sizes as conditions deteriorated, the agent systematically reduced exposure as volatility indicators escalated. This adaptive de-risking behavior produced materially lower maximum drawdown figures compared to fixed-rule baseline systems.

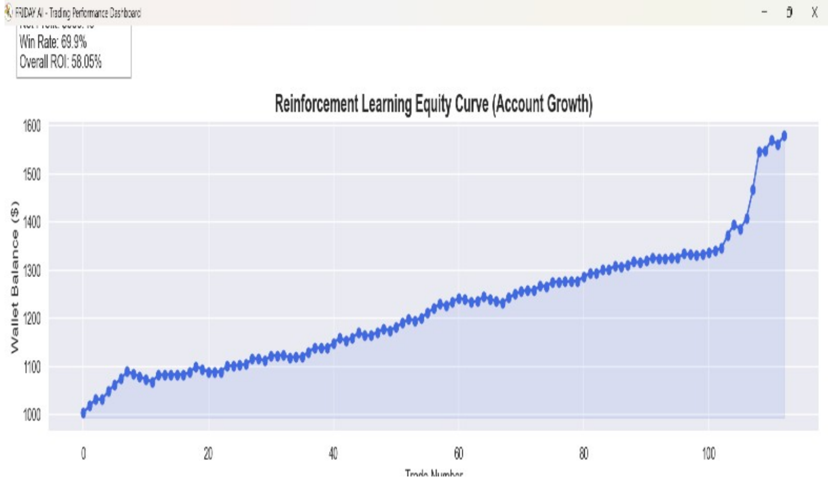

Figure 5. System Performance Monitor output summarizing trade execution statistics, overall win rate (60.00%), and cumulative realized profit over the evaluation period.

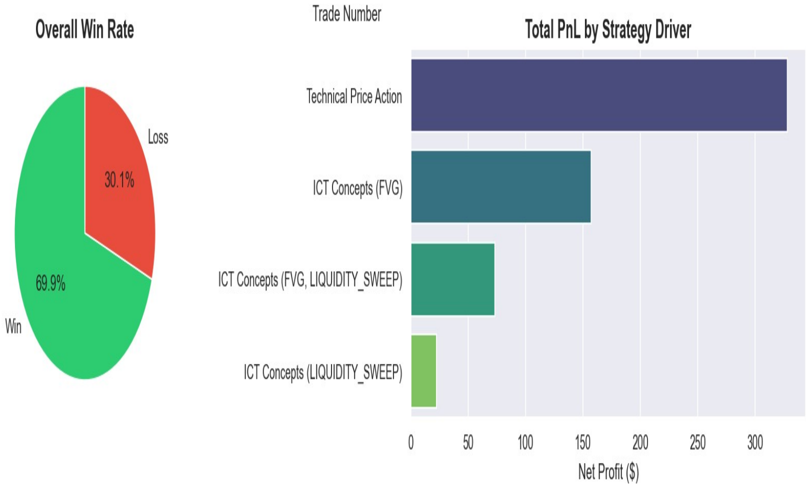

Figure 6. Simulated Account Equity demonstrating the progressive growth of the portfolio balance.

As evidenced by the Performance Monitor terminal output (Figure 5), over a 72-hour forward-testing period, the system autonomously processed thirty trades in total. It recorded eighteen profitable outcomes and twelve losing trades, successfully achieving a 60.00% win rate. Net profit across the trial amounted to $246.82. Operating from an initial simulated capital allocation of $1,000.00, the terminal account balance reached $1,246.82 (a +24.68% Return on Investment). These figures confirm that the risk management framework operated as designed: the magnitude and frequency of winning trades sufficiently exceeded that of losing trades to produce consistent, compounding net account growth.

The account equity curve generated during the evaluation period (Figure 6) provides qualitative insight into the system's behavioral profile. During the initial twenty trades, the equity progression was measured and gradual, consistent with a conservative posture during early market exploration. A subsequent pronounced upward movement indicates a high-probability directional move successfully captured and capitalized upon by the system. The ensuing drawdown remained strictly bounded by the RL agent's risk parameters and was followed by a

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

consistent recovery trajectory.

C. Live Execution and Order Management :

To validate the system's interface with live exchange environments, the bot's autonomous order placement was monitored in real-time.

Figure 7. Real-time autonomous execution on the Delta Exchange testnet. The images demonstrate the RL agent’s dynamic risk management, illustrating the automated, mathematically precise placement of Stop-Loss and Take-Profit brackets for short positions based on real-time ATR calculations.

As depicted in the live exchange interface (Figure 7), the bot successfully translated algorithmic decisions into physical market orders without human intervention. The visual evidence captures the RL agent's enforcement of a strict 1:2 Risk-to-Reward Ratio, demonstrating the automated placement of protective Stop-Loss brackets and Take-Profit brackets tightly wrapped around the entry price. The results affirm that the combined effect of LSTM-driven pattern recognition, sentiment-conditioned trade filtering, and RL-enforced risk parameters was highly effective in protecting margin capital while directing exposure toward higher-probability setups.

7. CONCLUSION

The principal contribution of this work is the demonstration that adaptive, learning-based trading architectures can substantively outperform the static, rule-based systems that currently dominate retail algorithmic cryptocurrency trading. Rather than encoding market knowledge in fixed conditional logic, the proposed framework accumulates knowledge dynamically through continuous interaction with the trading environment.

The integration of sentiment analysis proved particularly valuable. By grounding decisions in the broader informational context established by financial news flows, the framework avoids a category of costly errors—entering the market contrary to prevailing fundamental sentiment—that purely technical systems are architecturally incapable of preventing. The combination of LSTM-based historical pattern recognition, RL-based adaptive strategy, and NLP-based sentiment awareness produces a system exhibiting a level of situational intelligence meaningfully closer to that of an experienced human trader, while operating without the fatigue and emotional biases that constrain human performance.

8. FUTURE SCOPE

The current work establishes a functional and empirically validated baseline for adaptive AI-driven cryptocurrency trading. Several important extensions are planned for subsequent development phases:

Live Capital Deployment: The most immediate priority is transitioning from a simulated environment to live operation with real capital. This phase will require additional hardening of the

DOI:10.46647/ijetms.2026.v10i02.042 ISSN: 2581-4621

execution infrastructure to address partial fills, variable market liquidity, and exchange-specific latency characteristics.

Multi-Timeframe Analysis: Future iterations will extend the system's temporal horizon to incorporate higher-timeframe trend information. Reasoning simultaneously across intraday and macro-level structures will enable the framework to align short-term entries with the prevailing directional bias at higher timeframes.

Derivatives and Advanced Portfolio Strategies: Planned extensions include adapting the RL agent to trade derivatives instruments, including futures and options contracts. This capability would enable implementation of sophisticated portfolio-level risk management techniques, such as systematic hedging, to reduce directional exposure during periods of elevated market uncertainty.

Multi-Agent Architecture: The long-term vision is a multi-agent system in which specialized RL agents are deployed concurrently across different asset classes or market segments. Individual agents would function as domain specialists while sharing summary-level information with a coordinating meta-agent responsible for portfolio-level risk allocation—enabling management of diversified portfolios with a degree of asset-specific precision that no single generalist agent can provide.

9. REFERENCES

[1] S. Hochreiter and J. Schmidhuber, ”Long short-term mem- ory,” *Neural Computation*, 1997.
[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.
[3] V.Mnih et al.,"Human-Level Control Through Deep Reinforcement Learning," Nature, vol. 518, pp. 529–533, Feb. 2015.
[4] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep Direct Reinforcement Learning for Financial Signal Representation and Trading," IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 3, pp. 653–664, Mar. 2017.
[5] Z. Jiang, D. Xu, and J. Liang, "A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem," arXiv preprint arXiv:1706.10059, 2017
[6] D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-Trained Language Models," arXiv preprint arXiv:1908.10063, 2019.
[7] 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, Oct. 2018.
[8] A. Cartea, S. Jaimungal, and J. Penalva, Algorithmic and High-Frequency Trading. Cambridge, UK: Cambridge University Press, 2015.
[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv preprint arXiv:1707.06347, 2017
[10] M. Dixon, I. Halperin, and P. Bilokon, Machine Learning in Finance: From Theory to Practice. Cham, Switzerland: Springer, 20