SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Optimizing Reinforcement Learning in High-Uncertainty Environments

Dr. Sanjay Agal[1], Rahul Bhatt[2], Vimal Bibhu[3], Harshini R[4], Gayathri M[5], Dr. K. Suneetha[6], Dr. Prashant Kalshetti[7], Kiran Ingale[8]

[1] Professor, Department of Artificial Intelligence and Data Science, Faculty of Engineering and Technology, Parul Institute of Technology, Parul University, Vadodara, Gujarat, India. Email: sanjay.agal32685@paruluniversity.ac.in ORCID: 0000-0003-0352-0418
[2] Assistant Professor, Department of Computer Science & Engineering, Dev Bhoomi Uttarakhand University, Dehradun, Uttarakhand, India. Email: socse.rahul@dbuu.ac.in ORCID: 0009-0004-0486-9414
[3] Professor, Department of Computer Science & Engineering, Noida International University, Greater Noida, Uttar Pradesh, India. Email: vimal.bhibu@niu.edu.in
[4] Assistant Professor, Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: harshinir@maher.ac.in
[5] Assistant Professor, Department of Management Studies, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, India. Email: gayathrimba@maher.ac.in
[6] Professor, Department of CS & IT, JAIN (Deemed-to-be University), Bengaluru, Karnataka, India. Email: keerthisuni.k@gmail.com ORCID: 0000-0001-6738-3921
[7] Professor & Head, Department of BBA, Dr. D. Y. Patil Vidyapeeth, Global Business School and Research Centre, Pune, Maharashtra, India. Email: prashantkalshetti@gmail.com
ORCID: 0000-0003-3553-2093
[8] Department of Electronics & Telecommunication (E&TC) Engineering, Vishwakarma Institute of Technology, Pune, Maharashtra 411037, India. Email: kiran.ingale@vit.edu

Abstract
Deep reinforcement learning (DRL) has recently been found to be quite beneficial for financial decision-making because it can effectively learn an optimal strategy in a highly volatile market environment. The problem with conventional RL model arises in their inherent problems related to the stability and efficiency of exploration in highly uncertain environments. However, high levels of volatility, stochasticity of the prices, and information incompleteness in financial data tend to generate highly unstable convergence and inefficient exploration in the process of learning. This research paper aims at presenting a novel Intelligent Harmony Search–Adaptive Twin Delayed Deep Deterministic Policy Gradient (IntHS-ATD3PG) framework that will enable financial decision-makers to make profitable decisions under conditions of high volatility and uncertainty. To do that, financial data will be collected from publicly available sources and preprocessed through min-max normalization. Features will be extracted with the help of the Discrete Wavelet Transform (DWT) method. These features will be utilized by the proposed ATD3PG model in order to learn an optimal strategy in the continuous action space. Experimental results demonstrate superior performance, achieving a cumulative return of 19.5%, a Sharpe ratio of 3.70, and reduced volatility of 10.4%, outperforming conventional RL models. The proposed method, implemented using Python, provides a scalable and robust solution for adaptive financial optimization.

Keywords: Reinforcement learning (RL), Financial decision-making, Risk-aware trading, Uncertainty environment.

1. Introduction

The application of adaptive Deep Reinforcement Learning (DRL) has proven to be highly effective in increasing the performance of financial decision-making in high uncertainty markets. It incorporates the elements of data-driven learning and optimization methods to address the challenges that arise, such as volatility, nonlinearity, and missing information. DRL increases the performance of risk-informed trading in dynamic environments and enhances convergence stability through feature selection, policy formulation, and parameter adjustment [1]. This approach ensures that models can adapt to different market dynamics through their ability to identify the optimal actions in response to changes in inputs and uncertainties. Adaptive DRL systems increase efficiency and decision-making policies by ensuring

continuous learning and interaction within the market context. The process of learning is optimized by incorporating optimization methods [2]. Additionally, learning from policy improvement and ensuring stable performance is highly dependent on adaptive feature extraction and adaptive parameter tuning. This maintains consistency during training and operation while ensuring that the system is able to respond rapidly to changes in financial market conditions. More effective learning, exploration, and decision-making are provided by modern RL approaches [3]. The adaptive DRL optimization forms an excellent foundation for intelligent decision-making in a financially uncertain environment [4]. Making financial decisions under conditions of high uncertainty in the financial markets is hampered by volatility, nonlinearity, and lack of information. Financial markets are very dynamic and stochastic, which makes it impossible to predict the behavior of these systems. The problems of learning under high-dimensional and constantly changing settings compound the issue [5]. Training RL models is fraught with instabilities, ineffective exploration, and vulnerability to changes in the environment. Uncertainty regarding data and predictions influences the quality of financial decision-making. In addition, changes in the distribution of the market present another challenge [6]. Traditional approaches, such as those used in rule-based systems, classical optimization models, and RL approaches, have various drawbacks while operating in an uncertain financial environment. Traditional approaches involve the use of fixed parameters and assumptions, making them less flexible with respect to market changes [7]. Both value-based and distributional RL approaches struggle with issues related to convergence stability and exploration problems. Model-based RL approaches, which help improve sample efficiency, suffer from overfitting problems, model bias, and unreliable predictions. Moreover, all of the above-mentioned approaches are highly sensitive to hyperparameters and cannot generalize [8].

Research Aim: To address the above concern, the aim of this research is to optimize the strategy for the application of DRL methods that allow for improved decision-making by the agent in uncertain financial markets. The IntHS-ATD3PG algorithm is used, where intelligence is incorporated in the harmony search technique with adaptive policy optimization.

Research Organization: Section 1 presents the research background, motivation, and objectives for adaptive DRL in uncertain financial markets. Section 2 reviews the previous RL and optimization techniques. Section 3 introduces the IntHS-ATD3PG approach. Section 4 provides results and discussion, and Section 5 concludes the research with key findings.

2. Related Works

Table 1 presents a table of existing RL methods, showcasing their procedures, successes, and limitations for financial decision-making problems.

Table 1: Adaptive RL Financial Decision Analysis

Ref	Objective	Method	Result	Limitations
[9]	RL for financial optimization through adaptive learning	Trading strategies based on RL that use data	Improving flexibility and effectiveness in decision-making	require unpredictable learning and many data inputs.
[10]	Optimizing stock choice with adaptive RL	Learning from data along with deep RL	Improved performance of trade prediction	markets Optimizing stock choice with adaptive RL
[11]	Optimize project portfolios dynamically under deep uncertainty	Adaptive RL using ensemble Q-learning with meta-learning and adaptive exploration–exploitation strategies	Achieved 68% improvement in resource allocation and 52% in strategic alignment	Performance varies across different organizational contexts
[12]	Integrate AI-driven analytics with expert insights for enhanced risk assessment.	A hybrid approach combining data-driven analytics with qualitative expert knowledge	Improves risk evaluation, reduces bias, and enhances predictive accuracy	Difficulty in balancing quantitative data with qualitative inputs
[13]	Enhance stock price prediction using Bayesian Gated Recurrent Unit (GRU) for trading strategies.	Bayesian-inferred GRU architecture for long-term stock forecasting	Improves volatility management and Return on Investment (ROI) compared to standard GRU	Performance degrades under extreme market conditions

[14]	Adaptive RL for optimizing financial decisions	DRL for trading policies	Better decision-making and improved profitability	Data-dependent and market-sensitive
[15]	Optimizing financial decisions through adaptive RL	Multiple agents in the DRL trading model	Increased effectiveness in return and risk considerations	Instability in training and high complexity
[16]	Integrate RL with game theory for trading strategies	Hybrid Deep Q-Network combined with Nash Equilibrium concepts	Outperforms static strategies and standard RL baselines	Validated only on synthetic and historical simulation data
[17]	Develop adaptive portfolio management using RL	Deep Deterministic Policy Gradient (DDPG) with flexible reward functions	Achieved a 12.4% increase in returns and a 15.7% improvement in Sharpe ratio	Evaluated only on the S&P 500 dataset

3. Methodology

The designed approach utilizes financial market data sourced from credible financial institutions, which comprise historical stock prices, transaction volumes, and macroeconomic indices. To maintain data consistency, preprocessing methods, such as min-max normalization and missing value imputation, are applied to the data set. DWT is used for detecting market trends by extracting features using DWT. Thereafter, optimization of financial decision making and adaptive learning will be achieved based on the extracted features. The overall framework design of the workflow for making financial decisions under uncertainty is presented in Figure 1.

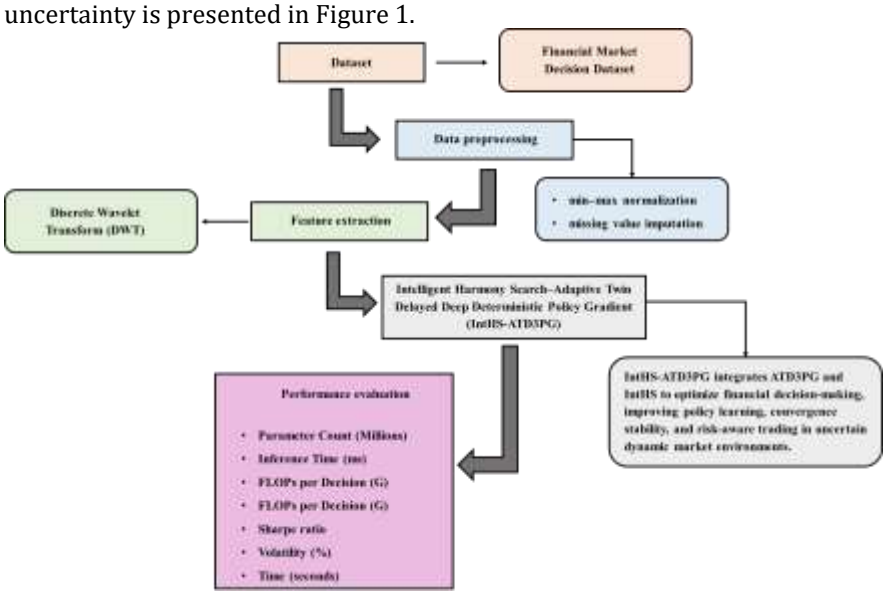

Figure 1: Financial Optimization in Uncertain Market Environments

Data collection: The data set consists of 12,000 observations from the daily financial market, created specifically for making decisions under uncertainty. This data set comprises not only price information but also technical indicators, macroeconomic factors, risk measures, and sentiment scores. Portfolio features, including the holdings, cash balances, and total portfolio value, are also included in the data set. The target variable represents trading decisions categorized as Buy, Hold, or Sell. Kaggle Source: https://www.kaggle.com/datasets/colabsss/financial-market-decision-dataset

Data feature exploration: Figure 2 (a) brings out the periods where there is an increase in market uncertainty since it indicates the distribution of the Volatility Index (VIX) over time. It enables one to understand the areas that can experience market stress because of the movement in the VIX index when there are important market events and investor emotions. Figure 2 (b) indicates the trend in stock prices over time, highlighting the impact of market conditions on asset values.

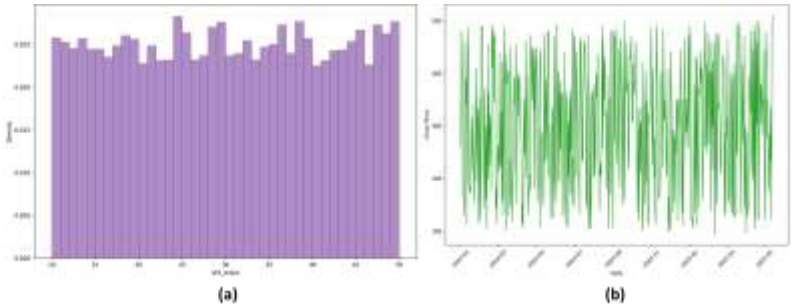

Figure 2: VIX Distribution and Stock Price Trends (a) Market Volatility (VIX) Distribution(b) Price Trends and Market Behavior

Figure 3 demonstrates the trends of relevant market indicators, such as momentum and market risk, showing their variability over time. It is essential to understand the effect of uncertainty conditions on the financial decisions made in situations.

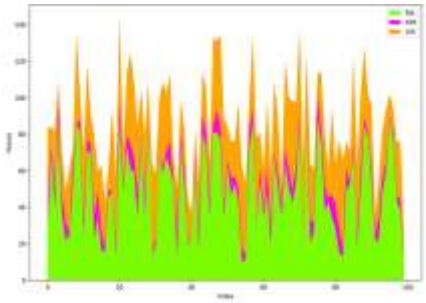

Figure 3: **Correlation of Market Indicators Over Time**

Data preprocessing using min–max normalization: Normalization of features is key in ensuring that learning is stable and convergence is improved in RL in environments, where there is significant uncertainty. Features such as the state variables, which include incentives, market variables, or the environment variables, are normalized into a certain range, normally [0,1], through a process referred to as min-max normalization. As part of training, the normalization minimizes any numerical instabilities that arise but keeps the relationship between the numbers constant. This means that the RL agent learns how to operate with similar inputs.

$$u' = \frac{u - min_B}{max_B - min_B}(new\_max_B - new\_min_B) + new\_min_B \quad (1)$$

Equation (1), u' shows the value of the original variable u after it is scaled into a new range. The starting number or datum that needs to be normalized is the one below: min_B is the minimum value of feature B is the initial data set, max_B is the maximum value of feature B is the initial data set, new_max_B is the expected highest value of the normalized range, new_min_B is the expected lowest value of the normalized range.

$$u' = \frac{u - \mu}{\sigma} \quad (2)$$

In Equation (2), μ is the mean for the data set. The arithmetic mean refers to the average of all data points in the data set, σ is the standard deviation for the data set u. The standard deviation measures the data points' scatter.

Feature extraction using Discrete Wavelet Transform (DWT): The DWT technique is used for the decomposition of financial signals corresponding to the states of decision-making or financial reward into many resolutions. By suppressing the high-frequency elements responsible for volatilities and the low-frequency market elements in the form of long-term trends, it enables the agent to generate stable features. This makes the learning process more efficient in terms of adapting the policies for optimal financial decision-making in uncertain market environments.

$$DWT(n, m) = \frac{1}{\sqrt{2^n}} \sum_l e(l)\, \psi \left[\frac{m - l2^n}{2^n}\right] \quad (3)$$

DWT at Point m and Level n. It is a wavelet coefficient, meaning that it is the result obtained through applying the wavelet transform on the discrete signal point m and level n, *where* n is the scale for the resolution. When it comes to the wavelet transform, the signal is decomposed at various scales, m is the translation, which refers to the location of the wavelet in relation to the signal. It allows the wavelet to move along the signal to obtain characteristics of the signal at different positions. $\sqrt{2^n}$ is an acts as a normalizing term. The normalization of wavelet coefficients for making sure that wavelet coefficients are properly

normalized as wavelets are scaled at different resolution levels is done using 2^n, and $\sum_l$ is an expression that denotes summation with respect to the index l, which varies within a particular range of l, $e(l)$ is the signal under analysis. It is the input signal labelled with index l, and ψ is the wavelet function. This is the mathematical equation used to transform the signal, $l2^n$ indicates the translation and dilation of the wavelet function. The notation $l2^n$ translates the wavelet along the signal, while the fraction 2^n dilates the wavelet function are represented in Equation (3).

Financial decision making in a high-uncertainty environment using IntHS-ATD3PG

The proposed IntHS-ATD3PG-based approach would allow for adaptive financial decision-making processes amid the highly uncertain market environment. Using policy learning through updating stable policies for the purpose of determining the best trading strategy with high accuracy, ATD3PG employs IntHS to fine-tune crucial parameters such as learning rates, weights of the policy network, and exploration techniques. This is done to optimize convergence speed and stability, hence improving performance.

ATD3PG for Financial Decision-Making: The proposed method makes use of a reward-aware parameter tuning process to adjust exploration noise in the ATD3PG model to tackle the large amount of uncertainty present in the financial environment. Instead of injecting pre-defined noise, the agent first evaluates the actions based on an estimated reward generated by state evaluation of the market. To ensure that favourable trading strategies are preserved, exploration strength is reduced when the estimated reward is high, and increased when the estimated reward is low to avoid suboptimal strategies.

$$z_j = q_j + \gamma_{i=1,2}^{min} R'\left(T_{j+}, b'_j \middle| \theta^{Ri'}\right) \quad (4)$$

In Equation (4), z_j can be interpreted as the new value or state of time j, q_j can denote the value (Q-value) of the state-action combination at the j-th time step, γ represents the discount factor (usually lies within the range of 0 to 1), which reflects the significance of future rewards over present rewards and $i = 1,2$ can be different operations or conditions indexed by i, min refers to the minimum operation, R' is the value of the reward function, taking into account the transition to the new state T_{j+1} represents the next state after the agent takes an action, b'_j denote the new or altered values for the parameters at time step j, $\theta^{Ri'}$ representing parameters of the model. The symbol ‖ might be denotes some kind of relationship or interaction between $b'_j$and the parameter $\theta^{(Ri')}$.

$$K = \frac{1}{m}\sum_j^m (z_j - R(T_{j,} b_j | \theta^R))^2 \quad (5)$$

Equation (5), K is the cost function or loss, m is the total number of observations in the dataset, and $\sum_j^m$ is the means that the summation is done with respect to m, where m can be an integer or some index set. z_j is the true or target value for the j-th data point, R is the predicted value based on the model's parameters, θ^R is the parameters of a reward model or policy network used for estimating expected rewards, and $R(T_{j,} b_j | \theta^R)^{\ 2}$ is the term is squared to calculate the squared error or loss**.**

$$b'_j = \mu'(T_{j+1} | \theta^{\mu'}) + clip(M(0, \sigma), -d, d) \quad (6)$$

In Equation (6), μ' this would usually be an example of a policy function, which maps a state space to an action space, $clip$ restricts 0 to lie within the range of min and max values, M is a random variable here take values between $-\sigma\ and\ \sigma$ on average with mean zero, $-d\ and\ d$ define the range over which 0 is the clipped and $\theta^{\mu'}$ represents the parameter set (weights) of the target policy network.

$$\sigma = \begin{cases} \frac{1}{0.8 q_{predict} + 0.5}, if\ q_{predict > 0} \\ 2, otherwise \end{cases} \quad (7)$$

In Equation (7), $q_{predict}$ denotes the estimated Q-value with respect to the RL algorithm.

IntHS for optimizing financial decision-making: To optimize the parameters of the proposed IntHS-ATD3PG approach for making financial decisions in highly uncertain environments, the IntHS algorithm is adapted. The algorithm automatically adjusts the pitch variation ratio and bandwidth depending on its current trading efficiency to cope with large-scale and stochastic search spaces. Financial conditions determine whether there should be increased or decreased exploration in case of varying rewards or market uncertainty.

$$PAR(j) = PAR_{min} + (PAR_{max} - PAR_{min}) \times degree \quad (8)$$

In Equation (8), PAR usually means Parameter in most cases, $PAR(j)$ denotes the scaled value of the parameter for the j-th object, PAR_{min} denotes the minimum value of the parameter PAR, PAR_{max} denotes the maximum value of the parameter PAR, $degree$ stands for the interpolation level, usually, it ranges from 0 to 1.

$$degree = [E_{max}(j) - mean(E)] \div [E_{max}(j) - E_{min}(j)] \quad (9)$$

In Equation (9), E can be used to represent a variable in the dataset. $E_{max}(j)$ is the highest value that is obtained from the variable E in the j-th data point or observation. $E_{min}(j)$ is the lowest value obtained from the variable E in the j-th data point or observation, $mean(E)$ represents the mean of E that is calculated from all data points.

Algorithm 1: IntHS-ATD3PG for Adaptive Financial Decision-Making

Load Financial Dataset

Preprocess DataNormalize the features using Min − Max normalization:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Feature ExtractionApply DWT to extract multi − scale features from the financial time − series:

$$\mathrm{DWT}(n,m) = \frac{1}{\sqrt{2^n}} \sum_l e(l)\psi\left(\frac{m - l2^n}{2^n}\right)$$

where $e(l)$*is the signal, and* ψ*is the wavelet function.*

Initialize IntHS to tune policy parameters, learning rates, and exploration strategies.

Initialize ATD3PG Model with actor, critic networks, and delay mechanisms for stable training.

Evaluate Initial Fitness E_{old}*using the ATD3PG model performance .*

For Iteration $\text{iter} = 1$*to* $\text{iter}_{\max}$*a. Update IntHS solution*:

$$A_{\text{new}} = \text{IntHS}(A_{\text{old}})$$

where A represents the updated parameters.

b. Train the ATD3PG model using the updated parameters A_{new}*and compute fitness* E_{new}.

c. If $E_{\text{new}} < E_{\text{old}}$*, update the solution*:

$A_{\text{old}} = A_{\text{new}}, E_{\text{old}} = E_{\text{new}}$

d. Else, retain previous parameters A_{old}.

End For

Train Final ModelTrain the ATD3PG model using the optimized parameters A_{final}*from IntHS.*

For Each Data Instance x

$\in$ *Wa. Predict the action* b_j*for the financial decision at time jusing the trained model*:

$$b_j' = \mu'(T_{j+1} \parallel \theta^{(\mu')}) + \text{clip}(M(0,\sigma), -d, d)$$

where μ'*is the policy function and* clip*restricts exploration noise.*

b. Execute the action in the environment, observe the reward, and update the model.

Return Optimized Trading Strategy.

Concerning the decision-making related to finance, the following algorithm 1 is used. While IntHS serves the purpose of tuning the learning rate and the exploration policy, DWT is responsible for feature extraction from time-series data. Stable policy learning can be guaranteed by ATD3PG.

4. Result and Discussion

The IntHS-ATD3PG algorithm has been tested and demonstrated increased profits, stability, and convergence speed. The algorithm was implemented in Python and executed on the Windows platform under the Intel® Core™ i7 processor.

Parameter Count (Millions): Reflects model complexity and required memory. Low value means efficiency and better deployment capabilities with maintained predictive power.

Inference Time (ms): Shows how fast the model makes a decision. Essential requirement in the case of modern high-frequency trading. Short inference time leads to quicker execution.

FLOPS per Decision (G): Computational costs per one prediction. The lower FLOPS lead to better scalability and suitability for high-frequency environments.

Cumulative Return (%): Model's profitability evaluated throughout the whole test period. Higher cumulative returns show better results.

Sharpe Ratio: Indicates risk-adjusted return that equals the difference between return and risk-free rate per unit of risk. High value corresponds to efficient trading strategy.

Volatility (%): Risk indicator showing variability of asset prices. Decreased volatility means that the model will perform better and its risks can be better controlled. Time (seconds): Total training period. Reduced training time means improved efficiency of the process and the ability to update models in changing market environments.

Performance evaluation using existing dataset: The suggested model IntHS-ATD3PG was trained on synthetic and historical data [16]. Inclusion of this type of data contributes to increased stability of the model. In comparison with traditional approaches to trading, such as ASM-RL, IntHS-ATD3PG outperformed in portfolio optimization tasks, which is demonstrated in table 2.

Table 2: Comparison of Model Architecture and Performance Metrics

Model Architecture	Parameter Count (Millions)	Inference Time (ms)	FLOPs per Decision (G)
ASM-RL [16]	3.5	2.40	0.12
IntHS-ATD3PG [proposed]	1.7	1.28	0.09

The suggested model IntHS-ATD3PG was trained using an already existing S&P 500 index data set in 100 assets [17]. This data set includes all kinds of information about financial markets like OHLCV Price values. And these data sets have been used to make a fair comparison between two models. Volatility and the variations in value of assets within this data set is a true reflection of the present world's state of affairs within financial markets. Through the improvements in performance of the IntHS-ATD3PG model in terms of mean return and Sharpe Ratio compared to traditional portfolio management techniques like Reinforcement Learning-based Adaptive Portfolio Management (RL-APM), the superiority of the former is evident from table 3 below.

Table 3: Comparative Analysis of Portfolio Management Strategies

Model Architecture	Cumulative return (%)	Sharpe ratio	Volatility (%)	Time (seconds)
RL-APM [17]	15.2	1.75	15.0	1350
IntHS-ATD3PG [proposed]	20.2	3.65	11.4	956

Performance evaluation using the proposed dataset: The both existing ASM-RL [16], RL-APM [17] was trained on proposed comprehensive data set encompassing past prices, trading volumes, and economic indicators. The IntHS-TD3PG yielded more favourable risk-adjusted performance than existing baseline models, regarding cumulative return (19.5%) and Sharpe ratio (3.70). The model achieved superior volatility control (10.4%) and stability, albeit with slightly extended inference time (1.10 ms) in Table 4.

Table 4: Model Performance Metrics in Financial Trading

Model Architecture	Parameter Count (Millions)	Inference Time (ms)	FLOPs per Decision (G)	Cumulative return (%)	Sharpe ratio	Volatility (%)	Time (seconds)
ASM-RL	3.3	2.20	0.11	12.1	1.30	14.0	1180
RL-APM	4.4	3.0	0.70	14.0	1.80	14.0	1370
IntHS-TD3PG [Proposed]	1.2	1.10	0.08	19.5	3.70	10.4	990

5. Conclusion

The IntHS-ATD3PG effectively deals with decision making difficulties in very uncertain financial environment as it incorporates the intelligent harmony search for policy optimization and uses the ATD3PG to learn consistently. This model performs exceptionally well with the cumulative return of 19.5%, Sharpe ratio of 3.70
and low volatility of 10.4%. Moreover, it also shows quick convergence indicating the scalabilityand effectiveness of the method to use in automated trading in fluctuating markets. In order to utilize this method, large size and quality data sets are needed. Also, the IntHS-ATD3PG can face some difficulties in highly fluctuating financial environment. Therefore, it needs to enhance the current capability of market adaptability, using diversified sources of financial data and hybrid RL models for improved efficiency in other financial environments.

Discussion: ASM-RL and RL-APM have certain limitations in complex financial situations due to their high parameter complexity, heavy computational cost and low risk-adjusted return. The proposed novel IntHS-TD3PG solves all these issues with its lightweight, quick inference process and convergence.

Reference

1 Bate, A.F., 2022. The nexus between uncertainty avoidance culture and risk-taking behaviour in entrepreneurial firms' decision making. Journal of Intercultural Management, 14(1), pp.104-132. http://doi.org/ 10.2478/joim-2022-0004

2 Keynia, F. and Memarzadeh, G., 2022. A new financial loss/gain wind power forecasting method based on a deep machine learning algorithm by using energy storage system. IET generation, transmission & distribution, 16(5), pp.851-868. https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/gtd2.12332

3 Hoel, C.J., Wolff, K. and Laine, L., 2023. Ensemble quantile networks: Uncertainty-aware reinforcement learning with applications in autonomous driving. IEEE Transactions on intelligent transportation systems, 24(6), pp.6030-6041 https://doi.org/10.1109/TITS.2023.3251376

4 Zhu, C., 2023. An adaptive agent decision model based on deep reinforcement learning and autonomous learning. J. Logist. Inform. Serv. Sci, 10(3), pp.107-118. https://doi.org/ 10.33168/JLISS.2023.0309

5 Fan, D., Ab Razak, N.H. and Soh, W.N., 2025. Financial trading decision model based on deep reinforcement learning for smart agricultural management. PeerJ Computer Science, 11, p.e3196. http://dx.doi.org/10.7717/peerj-cs.3196

6 Shavandi, A. and Khedmati, M., 2022. A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Systems with Applications, 208, p.118124. https://doi.org/10.1016/j.eswa.2022.118124

7 Sahu, S.K., Mokhade, A. and Bokde, N.D., 2023. An overview of machine learning, deep learning, and reinforcement learning-based techniques in quantitative finance: recent progress and challenges. Applied Sciences, 13(3), p.1956. https://doi.org/10.3390/app13031956

8 Charpentier, A., Elie, R. and Remlinger, C., 2023. Reinforcement learning in economics and finance. Computational Economics, 62(1), pp.425-462. https://doi.org/10.1007/s10614-021-10119-4

9 Hambly, B., Xu, R. and Yang, H., 2023. Recent advances in reinforcement learning in finance. MathematicalFinance, 33(3),pp.437-503. https://onlinelibrary.wiley.com/doi/10.1111/mafi.12382

10 Awad, A.L., Elkaffas, S.M. and Fakhr, M.W., 2023. Stock market prediction using deep reinforcement learning. Applied System Innovation, 6(6), p.106. https://doi.org/10.3390/asi6060106

11 Darvish, A. and Sepehri, M., 2025. Artificial Intelligence-Driven Project Portfolio Optimization Under Deep Uncertainty Using Adaptive Reinforcement Learning. Applied Sciences, 15(23), p.12713. https://doi.org/10.3390/app152312713

12 Beckley, J., 2025. Advanced risk assessment techniques: Merging data-driven analytics with expert insights to navigate uncertain decision-making processes. Int J Res Publ Rev, 6(3), pp.1454-1471. https://doi.org/10.55248/gengpi.6.0325.1148

13 Li, Y., Liwang, M. and Li, L., 2025. Learning to trade autonomously in stocks and shares: integrating uncertainty into trading strategies. Autonomous Intelligent Systems, 5(1), p.16. https://doi.org/10.1007/s43684-025-00101-4

14 Kabbani, T. and Duman, E., 2022. Deep reinforcement learning approach for trading automation in the stock market. IEEE Access, 10, pp.93564-93574. https://doi.org/10.1109/ACCESS.2022.3203697

15 Huang, Y., Zhou, C., Cui, K. and Lu, X., 2024. A multi-agent reinforcement learning framework for optimizing financial trading strategies based on TimesNet. Expert Systems with Applications, 237, p.121502. https://doi.org/10.1016/j.eswa.2023.121502

16 Black, J.A., 2026. Adaptive Strategic Decision Making Models for Volatile Markets using Reinforcement Learning and Game Theory. International Journal of Computational and Biological Sciences, 3(1), pp.34-42. https://doi.org/10.66238/ijcbs41

17 Huang, C. and Jing, Z., 2026. Reinforcement Learning for Single-Stock Portfolio Optimization in Equity Markets. International Journal of Computational Intelligence Systems, 19(1), p.37. https://doi.org/10.1007/s44196-025-01105-x