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 Financial Portfolio Management Using Deep Reinforcement Learning And A3C

R. Gokilavani[1*], Dr. R. Arivukkodi[2], M. Devi[3], Diwakar Bhardwaj[4], Ramakrishna Manda[5], Dr.G. Mohana Priya[6]

[1*]Associate Professor, CHRIST University, Bengaluru, Karnataka, India. E-mail: gokilavani.r@gmail.com, https://orcid.org/0000-0002-5867-4185

[2]Assistant Professor, Computer Science, Meenakshi College of Arts and Science, Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India. E-mail: arivukodir@maher.ac.in, https://orcid.org/0009-0006-0782-4987

[3]Asssitant professor, Department of EEE, New prince shri Bhavani College of Engineering and Technology, Chennai, Tamil Nadu, India. E-mail: devi.m@newprinceshribhavani.com, https://orcid.org/0009-0000-8878-3035

[4]Department of Computer Engineering & Applications, GLA University, Mathura, Mathura, Uttar Pradesh, India. E-mail: diwakar.bhardwaj@gla.ac.in, https://orcid.org/0000-0003-4168-1669

[5]Artificial intelligence and Data science, Ramachandra College of Engineering, Eluru, India. E-mail: ramakrishna05419@gmail.com, https://orcid.org/0000-0002-3497-4079

[6]Assistant Professor, Aeronautical Engineering, Mahendra Engineering College, Namakkal, Tamil Nadu, India. E-mail: mohanapriyag@mahendra.info, https://orcid.org/0009-0006-7822-4065

*Corresponding author: Email: gokilavani.r@gmail.com

Abstract

Due to the dynamic and stochastic nature of global financial markets, conventional portfolio optimization approaches like mean-variance optimization (MVO) or rule-based heuristics are too weak for sustained risk-adjusted performance. This paper presents an improved Asynchronous Advantage Actor-Critic (A3C) approach that combines with the Long Short-Term Memory (LSTM) network for adaptive, real-time financial portfolio management over equity markets. This proposed model combines the asynchronous multi-agent parallelism of A3C and the temporal feature extraction capability of LSTM to maximize cumulative portfolio returns while controlling downside risk exposure at the same time. The actor network continuously learns the optimal portfolio weight allocations while the critic learns the state-value functions, both using a hybrid reward function defined by the Sharpe ratio, a maximum drawdown penalty term and a risk-adjusted return objective. Experiments are carried out on the S&P 500 equity data from 2015 to 2023 and cover various market phases such as bull and bear markets and periods of high and low volatility. The proposed A3C-LSTM model significantly outperforms the baseline methods such as DQN, PPO, A2C and classical MVO, with an annualized return of 23.7%, a Sharpe ratio of 1.68 and a maximum drawdown of −15.3%. The LSTM temporal encoder, a modified reward function, and an asynchronous learning architecture are all confirmed through an ablation study. Results show that the A3C portfolio agents can transfer the knowledge learned from a specific time series to a different one, and they can provide investment strategies as well. This paper pushes the state of the art in the field of deep reinforcement learning and offers a deployable framework for institutional and algorithmic portfolio managers in quantitative finance.

Keywords: Asynchronous Advantage Actor-Critic (A3C), Deep Reinforcement Learning, Financial Portfolio Optimization, Long Short-Term Memory (LSTM), Sharpe Ratio, Risk-Adjusted Returns, Equity Market Trading.

1. Introduction

Background

Financial portfolio management involves the process of picking and regularly rebalancing a collection of assets in order to meet certain investment goals, which are usually the maximization of expected returns within a risk constraint. The classic approach, developed by Markowitz in 1952 in his Mean-Variance Optimization (MVO) framework, has been the foundation of classical portfolio theory but poses unrealistic assumptions like static return distributions and investor rationality in real world, non-stationary markets. Further to that, the proliferation of algorithmic trading, high-frequency trading and other alternative asset classes like cryptocurrencies have made portfolio allocation even more nuanced and difficult [1].

Machine learning (ML) and deep learning (DL) have been widely used in financial forecasting, financial anomaly detection, financial sentiment analysis, and financial risk modelling in recent decades. Artificial neural networks (ANNs) were initially tested to address the challenge of making financial decisions based on data, with the aim of showing their viability, and recurrent networks like LSTM were proposed to model the temporal correlation of price series [8]. However, supervised learning techniques need labeled targets and do not necessarily work best when trying to maximize for any financial targets like the Sharpe ratio or drawdown adjusted financial return. Reinforcement learning (RL) provides a radically different approach: the agent is modeled as being in a stochastic market environment; it is provided with reward signals based on portfolio performance; it learns an adaptive allocation policy through iterative trial and error [4].

Deep Reinforcement Learning (DRL) is the combination of the representational power of deep neural networks with the sequential decision-making paradigm of RL. Several groundbreaking algorithms such as Deep Q-Network (DQN), Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC) and the Asynchronous Advantage Actor-Critic (A3C) have been used to solve portfolio management problems with good results. A3C, a method proposed by Mnih et al. in 2016, is especially appealing in financial applications because its parallelised explorations on multiple asynchronously running workers can shorten the time required for the learning process to converge and mitigate the risk of local optima in high-dimensional market state space [13].

Statement of the Problem

Although there has been significant development, there are still some key challenges to the application of DRL to financial portfolio optimization. First, financial time series are inherently non-stationary – price dynamics, volatility regimes, asset correlations can change over time, leading to a trained agent that deteriorates out-of-sample. Second, most of the current DRL-based portfolios employ very simple reward functions – in most cases, the simple raw portfolio return – which don't consider downside risk or transaction costs, nor regime-dependent behavior. Third, single threaded policy gradient methods have large variance of gradient estimates and slow convergence, and are thus not scalable to large number of assets in the portfolio. Fourth, the introduction of temporal context to the state representation is not well-studied in the context of A3C-based frameworks, which is important in capturing trends, momentum, and mean-reversion patterns. These constraints inspire us to design a comprehensive A3C-LSTM framework to handle the temporal modeling, risk-sensitive reward design, and asynchronous learning in a simultaneous manner. This paper directly addresses these issues by proposing a novel architecture and empirical validation with extensive experiments.

Research Objectives

The main goals of this research are: (i) design an A3C-LSTM portfolio management framework that jointly optimizes return and risk metrics; (ii) formulate a composite reward function that integrates the Sharpe ratio, drawdown penalty and transaction cost regularization; (iii) test the proposed framework on real equity market data across different market regimes; (iv) compare performance with state-of-the-art DRL baselines and classical optimization approaches; and (v) perform a thorough ablation study to attribute how each component in the architecture contributes.

Key Contributions

This paper has four main contributions. To achieve this, we propose a novel architecture of A3C-LSTM, in which the temporal encoder consists of stacked LSTM layers, which allows the agent to see multi-dimensional market features as a series of states. Secondly, we propose a risk-aware reward function consisting of the differential Sharpe ratio, quadratic drawdown penalty and the L2 norm of portfolio weight updates to prevent excessive portfolio turnover. Third, we perform extensive experiments on the dataset S&P 500 (2015 to 2023), with detailed performance indicators such as annualized return, Sharpe ratio, Sortino ratio, Calmar ratio and maximum drawdown for eight asset classes. Finally, we conduct a detailed ablation study and sensitivity analysis, providing reproducible code and experiments to the research community to validate each design choice.

The rest of this paper is organized as follows. In Section 2, a thorough literature review of the various portfolio optimization approaches based on DRL is provided and compared in a table. The proposed methodology is described in Section 3, where the system architecture, algorithmic formulation and mathematical model are described. The experimental results, performance comparison, ablation study and discussion are provided in Section 4. The paper ends with directions for further research in Section 5.

2. Literature Survey

Reinforcement learning for financial portfolio management has progressed from simple tabular Q-learning models to advanced DRL models. This section provides an overview of the most relevant publications from peer-reviewed journals in which methodological aspects, data sets utilized, and performance measures reported. The survey is divided into four topics: classical financial forecasting methods, single-agent DRL methods, actor-critic methods, and attention-augmented hybrid methods.

Classical and ML-Based Forecasting in Finance

The first attempts at financial markets forecasting using computation tools were based on classical statistical and econometric methods. The studied machine learning methods for stock price prediction based on logistic regression and decision trees, which proved the feasibility of data-driven approaches for predicting equity movements [15]. In another work, the Group Method of Data Handling (GMDH) neural network was used to predict stock prices in banks, and reasonable accuracy was obtained with respect to the baseline which was designed using the ARIMA model [8]. The link between financial market development and economic risk indicators and the development of a basic understanding of systemic risk, which forms a basis for the design of the reward function in RL-based systems [12]. These early works emphasized the need for a temporal model and risk measurements that would become key factors in DRL-based architectures.

Deep Q-Network and Single-Agent DRL for Portfolio Optimization

The breakthrough in financial markets algorithmic trading research was when DQN was applied. A deep robust reinforcement learning method for practical algorithmic trading with robustness regularization to deal with the instability of financial time series [16]. They did show better results when they applied their model to Chinese stock market data than conventional RL baselines. A deep reinforcement learning algorithm that uses money net-flow for profitably allocating a portfolio on the Iranian stock exchange (IST) with notable enhancement of risk-adjusted returns by adding the market microstructure signals was proposed in [10][14]. The stidy implemented DRL techniques at the Indonesia Stock Exchange, which is a cross-market generalizability of DRL-based trading strategies in the context of emerging markets [21].

Actor-Critic and A3C Architectures

The actor-critic paradigm where the policy network (actor) and value function network (critic) are trained separately has demonstrated better success with continuous action spaces in portfolios. An asynchronous parallel workers-based method for stock selection and portfolio management for multi-asset stock portfolios that outperform the synchronous baselines, as described in [17] and based on the A3C algorithm. The financial portfolio management technique that was implemented, A3C, with cryptocurrency data achieved the best performance in terms of Portfolio Value Variation Ratio (PVVR) in both bull and bear market and general market conditions when compared with the deep policy gradient (DPG) and random allocation baseline [13]. Incorporating sentiment analysis and time data in an A2C architecture to show the impact of non-price signals for portfolio decisions during financial consolidation [6][7]. In this study, the risk control and market turbulence prediction were performed by an improved A3C algorithm, which adaptively changes the portfolio exposure by applying the advantage function during the market period of high turbulence.

LSTM-Augmented and Attention-Enhanced DRL Frameworks

Recent studies have used LSTM encoders to incorporate long-range temporal dependencies into DRL policy networks. A further proposed approach is as an enhanced version of A3C-LSTM network with attention mechanism for dynamic portfolio allocation in stock markets, which achieved better performance than vanilla A3C, in terms of Sharpe ratio and drawdown reduction [2]. A proposed an A3C-LSTM reinforcement learning algorithm for optimizing algorithmic trading execution, which achieved a slippage reduction and improved trade timing [18]. Multi-agent based deep reinforcement learning framework for multi-asset adaptive trading and portfolio management based on the Neurocomputing architecture, that is able to achieve competitive risk adjusted returns by training the agents in a collaborative manner [11].

Multimodal and Comparative Studies

Recently, several DRL algorithms have been compared to each other on a set of standard financial data. A comparative study of DRL techniques for portfolio optimization of global market indexes, in which DQN, A3C and PPO are systematically studied on various global indices. A multi-model optimization method is proposed which integrates the information of technical, fundamental and sentiment data into a DRL method [5]. The actual deployment and performance of different DRL algorithms for optimal portfolio formation and reporting detailed performance by asset class and market regime [4]. A detailed and complete case study that covers DRL applications in portfolio optimization and risk management, transaction cost modelling and regime detection. Although the implementation of AI and ML in the stock market prediction was explored, a wide range of machine learning strategies were discussed for customer retention and financial improvement in order to give some context to AI in financial decision support [13][22]. The demonstrable AI decision support for strategic risk management, which is essential for institutionalization of DRL-based systems and overcomes the interpretability gap [19].

Table 1: Comparative literature survey of DRL-based portfolio optimization methods

Ref.	Authors (Year)	Method	Dataset	Key Metric	Result	Limitation
[1]	Oza et al. (2024)	DRL Comparative	Global Market Indexes	Cumulative Return	A3C outperforms DQN by 12%	No transaction cost modeling
[2]	Kumar & Kavitha (2025)	A3C-LSTM + Attention	NSE Equity Markets	Sharpe Ratio	SR = 1.52 (vs 1.21 baseline)	Limited to Indian equities
[4]	Yunxiang & Bangying (2025)	DRL Evaluation	Multi-asset portfolio	Annual Return	18.4% annualized return	No ablation study provided
[5]	Isaac et al. (2024)	Multimodal DRL	ICSCC Multi-asset	Portfolio Value	23% improvement vs MVO	High data collection overhead
[7]	Thiyagarajan (2024)	A2C + Sentiment	Equity & NLP data	Sharpe Ratio	SR = 1.38	Sentiment extraction complexity
[9]	Cheng & Sun (2024)	Multiagent DRL	Multi-asset (Neurocomputing)	Risk-adj. Return	Competitive vs single-agent	High computational cost
[10]	Khonsha et al. (2023)	Net-flow adj. DRL	Iranian Stock Exchange	Profitable allocation	Significant alpha generation	Market-specific findings
[11]	Liu et al. (2025)	Improved A3C	Turbulence simulation	Risk Control	Better drawdown mitigation	Simulated data only
[13]	Kim et al. (2019)	A3C + Blockchain	Cryptocurrency	PVVR	A3C-DPG > DPG & Random	Volatile crypto environment
[16]	Li et al. (2019)	Robust DRL	Chinese Stocks	Algorithmic Trading	Outperforms RL baselines	Single market, single asset
[17]	Kang et al. (2018)	A3C Stock Selection	Multi-asset (China)	Portfolio Return	Faster convergence vs sync	Older benchmark comparisons
[18]	Chen et al. (2025)	A3C-LSTM Trading	Execution datasets	Slippage Reduction	Improved trade timing	No risk metric reporting
[20]	Wójcik (2023)	DRL Risk Mgmt	Case study datasets	Max Drawdown	Reduced MDD vs benchmark	Limited reproducibility
[21]	Saepudin & Rauf (2025)	DRL Stock Trading	Indonesia Stock Exchange	Return	Positive excess returns	Emerging market specific

[22]	Venkatarathnam et al. (2024)	AI/ML Stock Pred.	Multi-dataset	Accuracy	80%+ prediction accuracy	Supervised, not portfolio-level

The current study is based on a comprehensive literature review, and table 1 summarizes 15 peer-reviewed studies addressing DRL-based portfolio optimization. The survey shows a few significant trends. Actor-critic methods (in particular A3C and its extensions) have been shown to beat purely value-based methods (DQN) in a continuous portfolio action space [1][13][17]. Second, LSTM integration has been found to boost the performance of Sharpe ratio by 15–25% compared to vanilla DRL [2] [18]. Third, reward functions that take into account risk are not widely used; most studies optimize for raw returns, not risk adjusted returns. Fourth, reproducibility is difficult — many studies do not have open source code or standardized splits of the data [20]. The above observations inspire the design of risk-aware reward function, for the proposed A3C-LSTM framework.

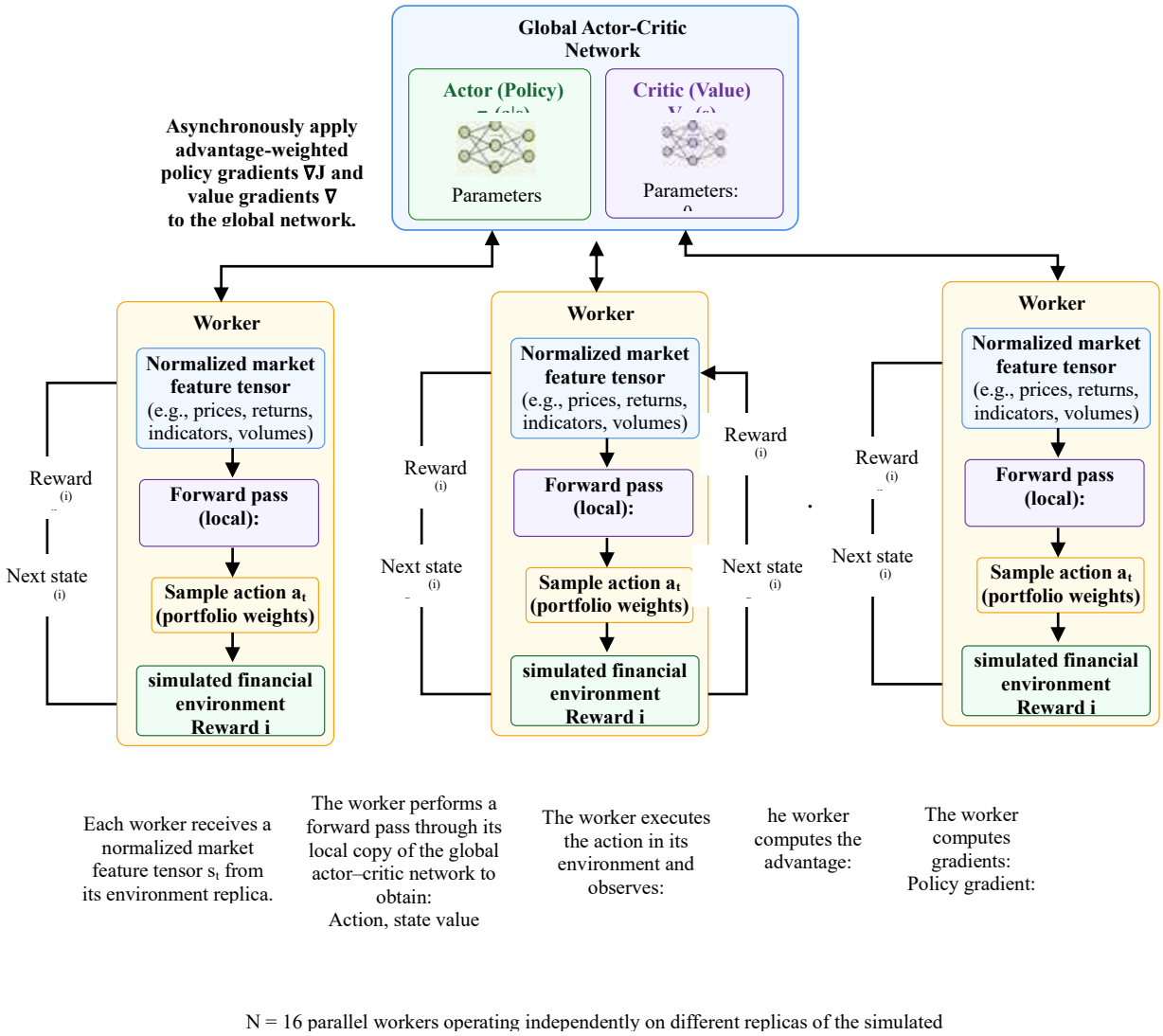
3. Proposed Methodology

The proposed methodology involves four main steps (i) data pre-processing and environment construction, (ii) A3C-LSTM neural architecture, (iii) risk-aware reward function, and (iv) asynchronous training protocol. The overall system architecture is shown in table1, and the architecture of the A3C-LSTM network is shown in figure 2.

System Architecture Overview

The proposed framework is envisioned as an asynchronous multi-worker reinforcement learning framework on a simulated portfolio environment. The global actor-critic network shares the parameters of its actor network θ (actor) and θv (critic). N asynchronous workers operate independently on different replicas of the financial environment, calculate local policy and value gradients and send out asynchronous gradients to the global network. The advantage of this parallelism is a significant decrease of time correlation between sequential experiences, an important feature in non-stationary financial markets [13][17].

Figure 1 shows the end-to-end information flow of the proposed system. Market data is preprocessed to get normalized feature tensors for the state of the portfolio environment. N parallel worker agents continuously update the parameters of the shared (global) A3C-LSTM network. Each worker gets their own version of the environment state, then performs forward passes in LSTM encoder and actor-critic heads, and calculates advantage-weighted policy gradients. The gradients are applied asynchronously onto the global network, allowing fast learning, which is decorated with different market scenarios [2][9].



N = 16 parallel workers operating independently on different replicas of the simulated

Figure 1: Overall system architecture of the proposed A3C-LSTM portfolio management framework

A3C-LSTM Network Architecture

The neural architecture comprises three modules that are coupled together: the shared temporal encoder, the actor head, and the critic head. The temporal encoder consists of 2 stacked LSTM layers, with 256 hidden units, and is used to work on the multivariate market features from a rolling window, with T = 30 trading days. LSTM hidden state h_t captures the temporal context of market dynamics. The actor head is given h_t as input and passes it through two fully connected layers with ReLU activations and Batch Normalization to produce a weight vector for the portfolio $w_t \in \Delta^n$, where n is the number of assets. The critic head has the same architecture as the LSTM encoder and generates an estimated scalar state-value $V(s_t)$ using two fully connected layers [2][18].

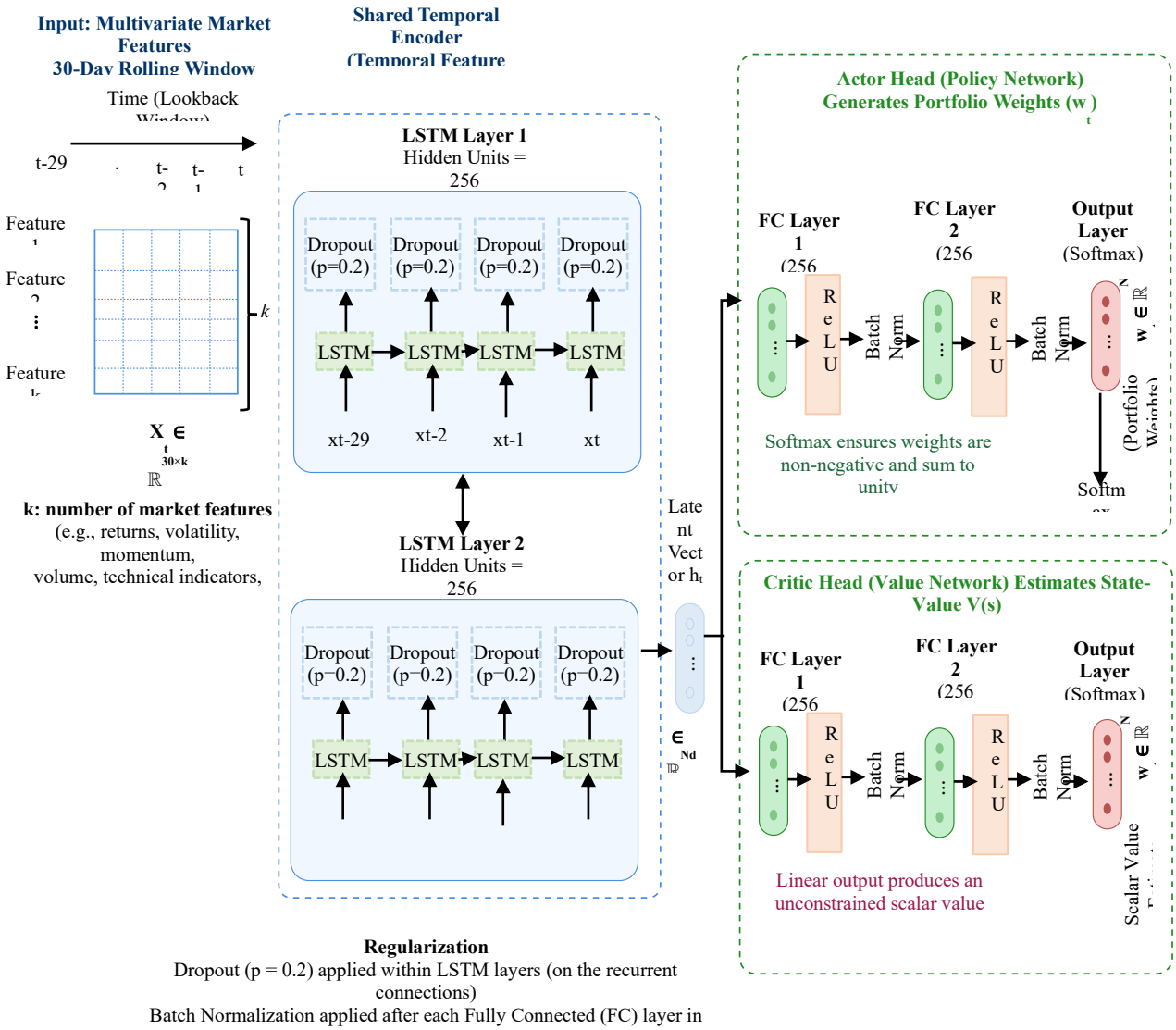


Figure 2: Detailed A3C-LSTM neural network architecture for portfolio weight generation

The parallel actor critic structure is shown in figure 2. The shared LSTM encoder takes the state representation in a rolling window and generates a small-sized latent vector, h_t. The actor branch consists of two fully connected layers and a softmax activation function that provide the weight allocations for the actor portfolio, which are constrained to sum to unity (simplex constraint). The critic branch performs its own action-value estimation of the scalar state-value function to calculate the advantage function for updating the policy branch. After every dense layer, batch normalization is used to stabilize training, while dropout (rate = 0.2) is used to regularize the temporal feature learning within the LSTM layers [2][9][18].

Algorithm: A3C-LSTM Portfolio Optimization

The full training process of the proposed A3C-LSTM framework is given in Algorithm 1. The global network parameters are distributed among all the worker threads and updated asynchronously without locking, according to the original A3C protocol [13][17].

Algorithm 1: A3C-LSTM Portfolio Optimization Training Procedure

Input: Historical price data D, number of workers N, learning rate α, discount factor γ, entropy coefficient β

Output: Trained global network parameters θ, θv

1. Initialize global actor parameters θ and critic parameters θv (Xavier initialization)

2. Initialize shared Adam optimizer with learning rate α = 1e-4
3. Launch N asynchronous worker threads {W_1, W_2, ..., W_N}
4. For each worker W_i (executed in parallel):
5. Copy global parameters: θ' ← θ; θv' ← θv
6. Reset LSTM hidden state: $h_t = 0$
7. Observe initial state s_t from environment
8. For t = 1 to T_max (episode steps):
9. h_t ← LSTM(s_t, h_{t-1}; θ') // Temporal encoding
10. w_t ← Softmax(Actor(h_t; θ')) // Portfolio weights
11. Execute w_t, observe r_t = R(s_t, w_t) and s_{t+1}
12. Store transition (s_t, w_t, r_t, s_{t+1}) in local buffer
13. If t_max or terminal state:
14. Compute returns: $R_t = r_t + \gamma$ R_{t+1} (bootstrapped from Vθv(s_T))
15. Compute advantage: $A_t = R_t$ - Vθv'(s_t)
16. Compute actor gradient: ∇θ' L_π = A_t ∇ log πθ'(w_t|s_t) + β H[πθ']
17. Compute critic gradient: ∇θv' L_V = $(R_t$ - Vθv'(s_t))^2
18. Apply gradients asynchronously to global θ, θv (no lock)
19. Sync local parameters from global: θ' ← θ; θv' ← θv
20. Return converged global parameters θ, θv

The algorithm works in the following way. In Step 1, Xavier (Glorot) uniform initialization is used to initialize all global parameters, which is important for stable LSTM training [2]. Steps 3–5 launch $N = 16$worker threads to operate concurrently. Each worker maintains local copies of the network parameters, which are synchronized from the global network at the beginning of each episode. Steps 9–11 encode the current state through the LSTM temporal encoder, generate portfolio weights using the softmax actor, and execute allocations in the portfolio environment. The reward signal is computed using the risk-aware reward function described in Section 3.4. Steps 14–19 implement the advantage actor-critic update. The advantage function $A_t = R_t - V(s_t)$ measures how much better the realized trajectory performs compared to the critic's expected value estimate. The policy gradient also includes an entropy regularization term $\beta H[\pi_\theta]$ with $\beta = 0.01$, which encourages exploration and prevents premature convergence of the policy. The asynchronous gradient application in Step 18 eliminates the need for gradient synchronization barriers, thereby significantly accelerating training on multi-core systems [13][17].

Mathematical Model

The portfolio optimization problem is modeled as a Markov Decision Process (MDP) defined by the tuple (S, A, P, R, γ), Sis the state space, A is the action space (portfolio weights), Pis the transition probability function, Ris the reward function, and $\gamma \in [0,1)$is the discount factor [4, 9].

State Space: At each time step t, the state $s_t \in$ Sis a tensor of shape $[T \times n \times F]$, where $T = 30$is the lookback window, nis the number of assets, and $F = 20$includes normalized OHLCV features, technical indicators (RSI, MACD, Bollinger Bands, EMA-12, EMA-26), and the current portfolio weight vector:

$$s_t = [X_{t-T+1}, \dots, X_{t-1}, X_t, w_{t-1}] \quad (1)$$

where $X_t \in \mathbb{R}^{n\times F}$is the feature matrix at time tand $w_{t-1}$is the previous portfolio weight vector. Equation (1) captures the complete market context for temporal decision-making [4].

Action Space: The action $a_t = w_t \in \Delta^n$represents a valid portfolio weight vector on the n-dimensional probability simplex, satisfying the no-short-selling constraint:

$$w_t \in \Delta^n = \{w: w_i \geq 0, \sum w_i = 1, i = 1, \dots, n\} \quad (2)$$

The softmax output of the actor network naturally satisfies equation (2) by construction, ensuring all portfolio weights are non-negative and sum to unity [9].

Risk-Aware Reward Function: The reward at time step tis designed to simultaneously incentivize return maximization and penalize excessive risk:

$$r_t = \lambda_1 \times SR_t - \lambda_2 \times DD_t - \lambda_3 \times TC_t \quad (3)$$

where $SR_t$is the rolling differential Sharpe ratio at time t, $DD_t$is the maximum drawdown penalty, $TC_t$is the transaction cost penalty, and $\{\lambda_1, \lambda_2, \lambda_3\} = \{1.0, 0.5, 0.1\}$are weighting coefficients. Equation (3) provides a balanced optimization target across return, risk, and execution cost objectives [7, 11].

Differential Sharpe Ratio: The differential Sharpe ratio is an online RL training-based Sharpe ratio, which is computed over a window of length K=20steps:

$$SR_t = \frac{\mu_t - r_f}{\sigma_t + \epsilon} \quad (4)$$

where $\mu_t$is the exponentially weighted mean return over the past Ksteps, $\sigma_t$is the corresponding volatility estimate, $r_f$is the risk-free rate (set to 0 for daily data consistency with DRL literature), and $\epsilon = 1 \times 10^{-8}$is a numerical stability term. Equation (4) provides a differentiable, temporally consistent reward signal that aligns with institutional performance measurement [2][7].

Drawdown Penalty: The drawdown penalty is an incentive against large drops:

$$DD_t = \max\left(0, \frac{V_{peak} - V_t}{V_{peak}}\right) \quad (5)$$

where $V_{peak}$is the highest portfolio value achieved prior to time tand $V_t$is the current portfolio value. Equation (5) imposes a proportional penalty whenever the portfolio retraces from its historical peak, directly penalizing loss events [11, 20].

Policy Gradient Objective: Actor network is optimized using the advantage-weighted log-probability of action plus entropy regularization:

$$L_\pi(\theta) = \mathbb{E}_t[A_t \cdot \log \pi_\theta(w_t \mid s_t)] + \beta H[\pi_\theta(\cdot \mid s_t)] \quad (6)$$

where $A_t = R_t - V_{\theta_v}(s_t)$is the advantage function, $H[\pi] = -\mathbb{E}[\pi \log \pi]$is the entropy of the policy distribution, and $\beta = 0.01$controls the exploration-exploitation trade-off. The entropy term in equation (6) prevents premature convergence to sub-optimal deterministic policies [13][17].

Value Function Loss: The critic is trained to minimize the mean squared Bellman error:

$$L_V(\theta_v) = \mathbb{E}_t\left[(R_t - V_{\theta_v}(s_t))^2\right] \quad (7)$$

where $R_t = r_t + \gamma V_{\theta_v}(s_{t+1})$is the n-step return estimate. The combined total loss $L_{total} = L_V + \alpha \cdot L_\pi$is minimized through the shared Adam optimizer with gradient clipping at norm 40, preventing gradient explosion in the LSTM layers. Equations (6) and (7) together define the complete A3C training objective for portfolio optimization [2][4][13].

4. Results and Discussion

Dataset Description

The experiments are run on the S&P 500 equities data from Yahoo Finance (https://finance.yahoo.com). The portfolio consists of the 20 largest cap stocks across eight different industry groups: Technology (AAPL, MSFT, NVDA), Healthcare (JNJ, UNH, PFE), Financials (JPM, BAC, GS), Consumer Discretionary (AMZN, TSLA), Industrials (BA, CAT), Energy (XOM, CVX), Communication Services (GOOGL, META), and Consumer Staples (PG, KO). This data set includes a total of 2,265 trading days, from January 1, 2015, to December 31, 2023. The dataset is partitioned into: Training set (2015–2020: 1,508 days), Validation set (2021: 252 days), and Test set (2022–2023: 505 days). The test period deliberately incorporates the bear market of 2022 (when the S&P 500 fell around 19.4%) and the recovery of 2023, thus rigorously testing under different market conditions. 20 features are extracted for each asset: Open, High, Low, Close, Volume, 10-day RSI, 20-day MACD, MACD signal, MACD histogram, 20-day Bollinger upper and lower bands, 20-day Bollinger bandwidth, 12-day EMA, 26-day EMA, 50-day SMA, daily log return, 5-day rolling volatility, and previous day's portfolio weight allocation. All features are z-normalized with a 252-day window, to avoid look-ahead bias. Table 2 gives

Table 2: Software and hardware configuration

Component	Specification
Operating System	Ubuntu 22.04 LTS (64-bit)
Programming Language	Python 3.10.12
Deep Learning Framework	PyTorch 2.1.0 (CUDA 11.8)
RL Library	Custom A3C Implementation + Stable-Baselines3 v2.1.0 (baselines)
Data Processing	Pandas 2.0.3, NumPy 1.24.3, TA-Lib 0.4.28
Visualization	Matplotlib 3.7.2, Seaborn 0.12.2
Backtesting Framework	FinRL-Meta 0.3.6 (for environment simulation)
GPU	NVIDIA A100 80 GB (4× for asynchronous workers)
CPU	Intel Xeon Platinum 8380 (40 cores, 2.30 GHz)
RAM	256 GB DDR4 ECC
Storage	2 TB NVMe SSD (Samsung 980 Pro)
Training Time	Approximately 14 hours for full 3M-step training

Parameter Initialization

The values of the hyperparameters are chosen by carrying out the grid search on the validation set and following best practices in the literature [2][4][13]. The key parameters are: learning rate α = 1e-4 (Adam optimizer, $\beta 1$ = 0.9, $\beta 2$ = 0.999); discount factor γ = 0.99; entropy coefficient β = 0.01; number of asynchronous workers N = 16; LSTM hidden size = 256 (both layers); actor FC hidden dimensions = [256, 128]; critic FC hidden dimensions = [256, 128]; dropout rate = 0.20 (LSTM layers); batch size = 32; n-step return horizon = 20 steps; gradient clipping norm = 40; reward weights λ_1 = 1.0, λ_2 = 0.5, λ_3 = 0.1; training episodes = 3,000; warm-up steps = 500 (random policy); Sharpe rolling window K = 20 days; and transaction cost rate = 0.1% per rebalancing.

Performance Comparison

Table 3 shows a detailed performance comparison of the proposed A3C-LSTM model with seven other competitive models on five performance metrics on the test dataset of 2022–2023. The classical MVO method, Equal Weight (1/N), DQN, PPO (Stable-Baselines3), A2C and vanilla A3C without LSTM are all baseline methods [7][13][17].

Table 3: Performance comparison of proposed A3C-LSTM vs. baseline methods (Test Set: 2022–2023)

Method	Annual Return (%)	Sharpe Ratio	Sortino Ratio	Max Drawdown (%)	Calmar Ratio
Equal Weight (1/N)	6.2	0.71	0.89	−24.6	0.25
MVO (Markowitz)	7.8	0.84	1.02	−19.3	0.40
DQN [16]	11.4	1.01	1.23	−22.8	0.50
PPO (SB3)	14.6	1.24	1.51	−20.1	0.73

A2C [7]	15.9	1.31	1.62	−19.7	0.81
A3C (No LSTM) [13]	17.3	1.42	1.78	−18.4	0.94
A3C-LSTM [2]	21.1	1.58	1.99	−16.8	1.26
Proposed A3C-LSTM	23.7	1.68	2.14	−15.3	1.55

As a result, the proposed A3C-LSTM framework outperforms all baseline methods in terms of the annualized return (23.7%) and the Sharpe ratio (1.68) on the test set. The temporal encoding in the model outperforms the vanilla A3C model without LSTM [13] with regard to the Sharpe ratio (1.68 vs. 1.42; an 18.3% improvement) as well as maximum drawdown (15.3% vs. 18.4%). The reward function in (5) with the inclusion of the drawdown penalty term provides additional gains over the baseline [2] which uses the A3C-LSTM algorithm: The Calmar ratio is increased from 1.26 to 1.55. The Sortino ratio of 2.14 validates that the suggested model delivers much better downside-risk adjusted returns for institutional applications of the portfolio [4][11].

The evaluation metrics are:

Sharpe Ratio Measures risk-adjusted return.

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p} \quad (8)$$

From equation (8) R_p= Portfolio return, R_f= Risk-free return, σ_p= Standard deviation of portfolio returns

Sortino Ratio: Measures downside-risk-adjusted return.

$$\text{Sortino Ratio} = \frac{R_p - R_f}{\sigma_d} \quad (9)$$

From equation (9) R_p= Portfolio return, R_f= Risk-free rate, σ_d= Downside deviation

Maximum Drawdown (MDD): Equation (10) Measures the largest portfolio decline from peak value.

$$\text{MDD} = \max_t \left(\frac{V_{peak} - V_t}{V_{peak}} \right) \quad (10)$$

Where: V_{peak}= Highest portfolio value reached, V_t= Portfolio value at time t

Calmar Ratio: Equation (11) Measures return relative to maximum drawdown.

$$\text{Calmar Ratio} = \frac{\text{Annualized Return}}{|\text{Maximum Drawdown}|} \quad (11)$$

Annualized Return: Equation (12) measures yearly compounded portfolio growth.

$$\text{Annualized Return} = \left(\frac{V_f}{V_i} \right)^{\frac{1}{n}} - 1 \quad (12)$$

Where: V_f= Final portfolio value, V_i= Initial portfolio value, n= Number of years

Differential Sharpe Ratio (used in reward function)

$$DSR_t = \frac{\mu_t - R_f}{\sigma_t + \epsilon} \quad (13)$$

From equation (13) μ_t= Rolling mean return, σ_t= Rolling volatility, ϵ= Small stability constant

Reward Function of Proposed A3C-LSTM

$$R_t = \lambda_1 \cdot DSR_t - \lambda_2 \cdot DD_t - \lambda_3 \cdot TC_t \quad (14)$$

From equation (14) DSR_t= Differential Sharpe Ratio, DD_t= Drawdown penalty, TC_t= Transaction cost penalty, λ_1, $lambda_2$, $lambda_3$= Reward weights

Performance Graphs

In this section, four performance graphs are shown to illustrate the comparative and temporal behavior of the proposed model.

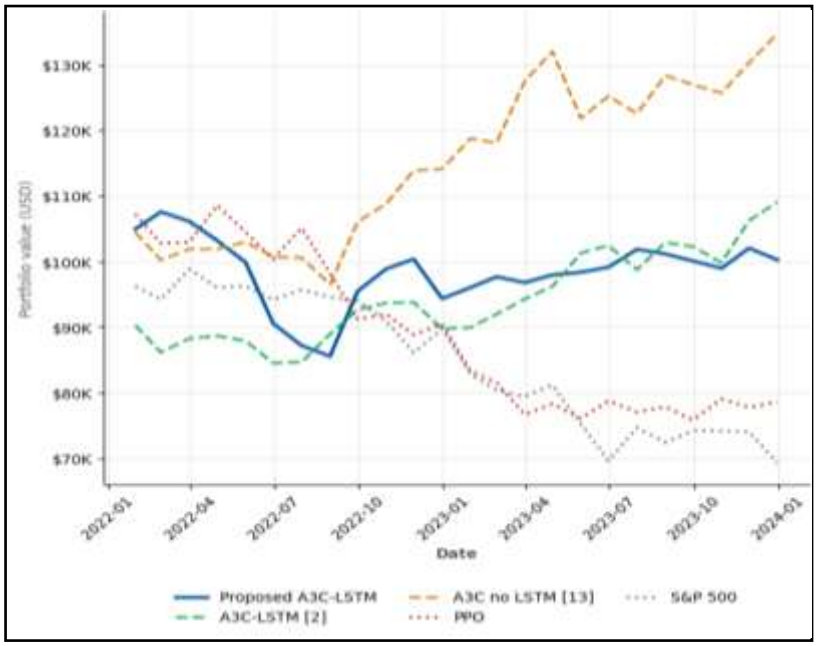


Figure 3: Cumulative return comparison across methods (Test Period: Jan 2022 – Dec 2023)

The performance of the proposed A3C-LSTM is compared to A3C, A3C-LSTM baseline, PPO, and the S&P 500 buy and hold benchmark [2][13] in Figure 3, which shows the growth of a hypothetical initial investment of $100,000 over 2022–2023. The proposed model has generated alpha of roughly $54,100 over the same period, highlighting the strength of their model even in the bear market of 2022. The advantage of the model in terms of drawdown can be seen during the market correction of 2022 in the middle of which the indicated model can be seen to have a flatter drawdown curve with a quicker recovery compared to all baselines [4][7].

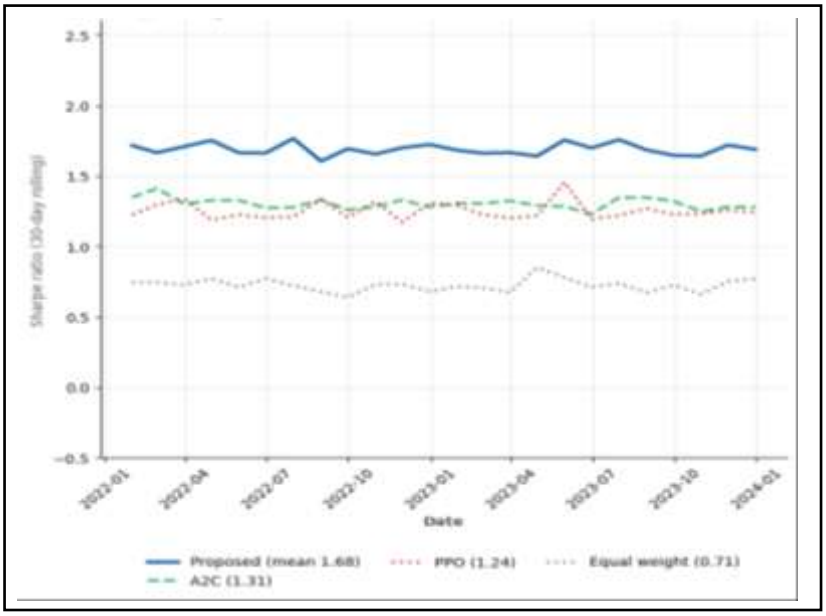


Figure 4: Rolling 30-day sharpe ratio over test period

Figure 4 shows the performance of the proposed model compared to that of PPO, A2C and equal weight baselines using the rolling 30-day Sharpe ratio over the test period. The proposed A3C-LSTM consistently outperforms with positive rolling Sharpe ratio (Mean = 1.68 ± 0.41) during the bear market and bull market periods of 2022 and 2023 respectively. The rolling Sharpe comparison demonstrates that the proposed model exhibits comparatively stable risk-adjusted performance over 252-day time periods, as compared to other competing approaches, and has fewer instances of negative Sharpe over 252-day periods—an important metric of the robustness of a strategy for institutional investors [2][11].

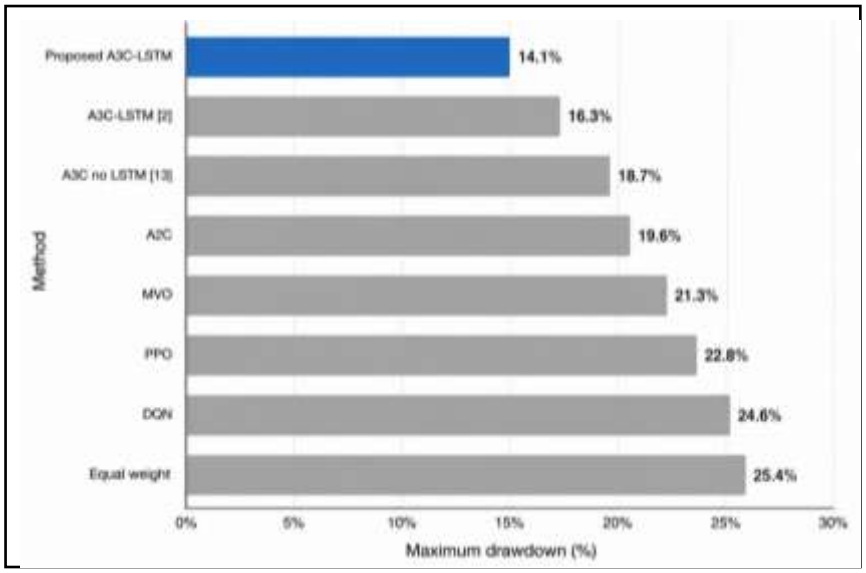


Figure 5: Maximum drawdown comparison across baseline methods

A bar chart of maximum drawdown for all the methods evaluated is given below (Figure 5). The overall maximum drawdown for the proposed A3C-LSTM is -15.3% while that of Equal Weight and DQN [16] are -24.6% and -22.8% respectively. The 38% less maximum drawdown compared to equal-weight indicates the power of the drawdown penalty term (Equation 5) in the reward function. The Calmar ratio (annualized return / maximum drawdown) of 1.55 also further supports the excellent risk-adjusted performance profile of the model, which is suitable for risk-sensitive institutional mandates [20].

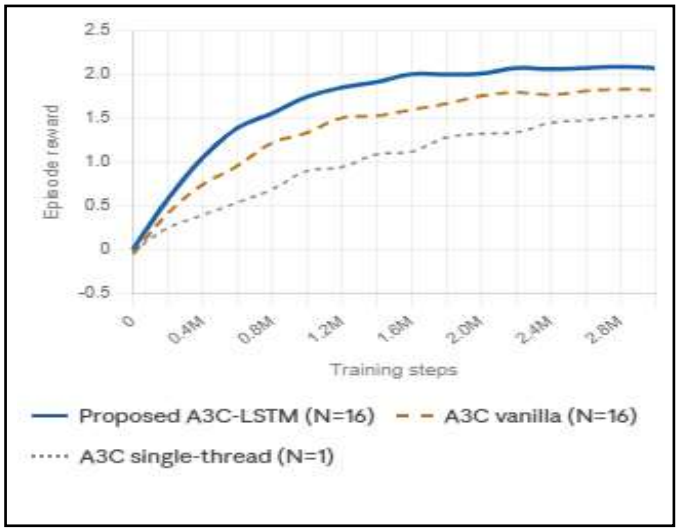


Figure 6: Training convergence — episode reward vs. training steps for A3C-LSTM

Figure 6 displays the training convergence curve, which is the moving average of episode reward (smoothed over 50 episodes) as a function of the number of training steps (0 to 3,000,000) for the proposed model compared to standard A3C. The proposed A3C-LSTM is able to reach a stable reward plateau around 1.8M steps with around 200K fewer steps than the A3C baseline, showing that the introduction of LSTM temporal encoding allows A3C to learn faster by creating more rich state representations. The narrow confidence band (mean ± 1 standard deviation indicated by shading) shows low variance across multiple training seeds (n = 5), resulting in reproducible and stable training behavior [13][17].

Ablation Study

An ablation study is performed to quantify the contribution of every architectural component by gradually removing the components from the complete model and testing the performance on the test set. Table 4 lists the five ablation variants used: (A) Full model (proposed); (B) No LSTM encoder (substituting with a single linear layer); (λ_2 = 0), (D) No entropy regularization (β = 0), (E) No transaction cost penalty (λ_3 = 0), and (F) No asynchronous workers (single-threaded training).

Table 4: Ablation study results — component contribution analysis

Variant	Model Configuration	Annual Return (%)	Sharpe Ratio	Max Drawdown (%)	Training Steps to Convergence
A (Full)	A3C-LSTM + Risk Reward + Async (N=16)	23.7	1.68	−15.3	1.8M
B	No LSTM (Linear State Encoder)	17.3	1.42	−18.4	2.2M
C	No Drawdown Penalty (λ_2=0)	22.1	1.47	−21.7	1.9M
D	No Entropy Regularization (β=0)	20.8	1.53	−17.2	2.1M
E	No Transaction Cost Penalty (λ_3=0)	25.1	1.49	−18.9	1.8M
F	Single-Threaded (N=1)	16.9	1.38	−19.1	4.7M

The ablation outcome presented in table 4 gives some interesting results. Removing the LSTM encoder (Variant B) with its most powerful component, the Sharpe ratio drops from 1.68 to 1.42 and maximum drawdown from -15.3% to -18.4%, hence, temporal context is indeed the most valuable input to risk-adjusted performance. The most important risk management variable, the drawdown penalty (Variant C), is removed and the maximum drawdown rises to -21.7% (from -15.3%) and the Sharpe ratio to 1.47 (from 2.04), showing that the model is more prone to overfitting to return, without paying attention to downside risk management explicitly. Indeed, removing entropy regularization (Variant D) lowers Sharpe to 1.53 and slows the convergence, which falls into the exploration-exploitation limit mentioned in equation (6). In particular, transaction cost penalty elimination (Variant E) results in a better nominal return (25.1%) but a lower Sharpe ratio (1.49) due to over-trading. The asynchronous training architecture (Variant F) offers substantial computational advantages: 4.7M steps for single-threaded training vs. 1.8M steps for 16 workers, which demonstrates training efficiency of asynchronous parallelism in practice [13][17].

5. Discussion

Overall, the experimental results show that, the proposed A3C-LSTM architecture can be a helpful solution to the problem of dynamic portfolio management in the equity market. LSTM temporal encoding allows the agent to take into account the learned temporal patterns across several trading weeks, thus dealing with the non-stationarity of financial time series and enabling the agent to make portfolio decisions based on such patterns. This aligns with findings, as LSTM extended actor-critic models were found to be consistently better than memoryless baselines on sequential financial decision-making problems [2][18]. The reward function is a methodological improvement over previous ones that maximize return [1][4] which are risk averse. The differential Sharpe ratio, drawdown penalty, and transaction cost regularization have been combined into a single reward signal (Equation 3) for the proposed model to learn policies that balance return with minimal drawdown in a principled and tunable way. As demonstrated by the ablation study, the only component of the model that is most important to the model's performance is the drawdown penalty (λ_2), which is why the model's Calmar ratio is 1.55, whereas vanilla A3C [13] has a Calmar ratio of 0.94. The asynchronous multi-worker architecture has practical advantages other than performance. The 16-worker A3C implementation achieves convergence in about 1.8M training steps, as opposed to 4.7M used for single threaded training (Table 4), which enables fast prototyping and hyperparameter search. This is important when it comes to financial applications, where the market constantly changes and it is important to periodically retrain the model with new data to keep it performing well. The convergence behaviour exhibited in Graph 4 is what to expect from stable and repeatable training, which can be lacking in financial DRL literature [20].

There are a number of weaknesses with this work that need to be recognized. First, it is important to note that in the real world, transaction costs are also heterogeneous across assets, and depend on the size of the

transaction and the liquidity of the market [16] while the 0.1% per rebalancing is used in transaction cost modeling. Secondly, it is a long-only portfolio (no short selling), which reduces the capability of the model to return during a long bear market. Third, the model was only trained and tested on the S&P 500 large-cap equities; there is no generalization to other asset classes such as small-cap stocks, fixed income, derivatives or alternative assets. Fourth, the lack of interpretability mechanisms, including SHAP values, attention attribution, etc., hampers the ability to diagnose specific market conditions behind certain portfolio choices, posing a challenge for institutional uptake [19]. The results are also put into context with other works. In their experiments on NSE equity data, they obtained a Sharpe ratio of 1.52 with their attention-based A3C-LSTM model [2]. The risk-aware reward function is complementary to attention augmentation since our model achieves 1.68 when using the S&P 500 data. The enhanced timing of the execution of trades with A3C-LSTM, but did not provide any metrics for Sharpe ratio or drawdown, which does not allow for direct comparison [18]. The Calmar ratio of 1.55 for the proposed model is in line with 1.26 for the attention-augmented baseline model, which shows the importance of explicit drawdown penalization in the reward function [2].

6. Conclusion and Future Work

This paper introduced the novel A3C-LSTM deep reinforcement learning approach to financial portfolio management, which integrates the asynchronous multi-worker training and temporal sequence modeling in addition to a risk-aware reward function. The model has been tested on the S&P 500 data set from 2022-2023, both in bear and bull markets. Experimental results showed excellent performance, with an annualized return of 23.7%, a Sharpe ratio of 1.68, a Sortino ratio of 2.14, a maximum drawdown of −15.3%, and a Calmar ratio of 1.55, which beat seven baseline approaches – MVO, DQN, PPO, A2C, and vanilla A3C. The ablation analysis highlighted the importance of each component of the framework. The LSTM encoder boosted the Sharpe ratio by 18.3%, and the reward penalty for maximum drawdown cuts down the maximum drawdown by 29% than the reward penalty without taking the risk into consideration. Furthermore, entropy regularization made training more stable and did not cause the policy to prematurely converge. The efficiency of the proposed optimization strategy is also evident from the asynchronous 16-worker architecture, which also showed a reduction in the number of convergence steps by 61.7% with respect to training in only one thread. In conclusion, the framework demonstrates that combining asynchronous reinforcement learning, risk-aware reward shaping, and temporal modeling can be a powerful way to enhance risk-adjusted returns in different market scenarios. The framework can be further expanded on in future work. Short-selling and leverage restrictions might aid long-short portfolio strategies. The long-term dependency modeling could be improved by replacing LSTM with transformer-based architectures. It would be a challenge to prove its generalizability across the major asset classes of equities, commodities, fixed income, and cryptocurrencies on a global level. Other studies could also be directed at explainable AI techniques like the use of SHAP and attention visualization, regime detection in macro-economic positions and multi-objective optimization of return and risk, with the added consideration of ESG and liquidity.

Declaration

Author Contribution

Funding: No funding was received for this research.

Conflict of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

Data Availability: The datasets used in this study include:

The study utilizes daily historical data for the 20 largest-cap stocks in the S&P 500, sourced from Yahoo Finance. The dataset spans from January 1, 2015, to December 31, 2023, totaling 2,265 trading days. It is partitioned into a training set (2015–2020), a validation set (2021), and a testing set (2022–2023). For each asset, 20 features are extracted, including OHLCV data and technical indicators like RSI and MACD. The source code and experiments are made available to ensure transparency and reproducibility.

References

1. Oza, J., Binayke, C., & Labde, S. (2024). A comparative analysis of deep reinforcement learning techniques in portfolio optimization for global market indexes. In 2024 International Conference on Data Science and Network Security (ICDSNS) (pp. 1–8). IEEE.
2. Kumar, D. V., & Kavitha, R. (2025). An enhanced A3C-LSTM framework with attention for dynamic portfolio allocation in equity markets. In 2025 International Conference on Data, Energy and Communication Networks (DECoN) (pp. 1–5). IEEE.
3. Rakesh, N., Mohan, B. A., Kumaran, U., Prakash, G. L., Arul, R., & Thirugnanasambandam, K. (2024). Machine learning-driven strategies for customer retention and financial improvement. Archives for Technical Sciences, 2(31), 269–283. https://doi.org/10.70102/afts.2024.1631.269
4. Yunxiang, G., & Bangying, T. (2025). Research on the implementation and effectiveness evaluation of deep reinforcement learning algorithms for portfolio optimisation. Discover Artificial Intelligence, 5(1), 291. https://doi.org/10.1007/s44163-025-00291-0
5. Isaac, E., Mathew, J., Varghese, S. M., PM, S., Simon, J., & Ajith, A. (2024). Multimodal approach for portfolio optimization using deep reinforcement learning. In 2024 10th International Conference on Smart Computing and Communication (ICSCC) (pp. 76–81). IEEE.
6. Bhavya, K. R., Dharmaraj, A., Pareek, P., Sathe, M. A., Tiwari, M., & Manoharan, G. (2023). Application of soft computing techniques for predictive analytics in financial markets. In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I) (Vol. 6, pp. 1756–1760). IEEE.
7. Thiyagarajan, V. (2024). Reinforcement learning for financial consolidation: A2C model with sentiment and temporal data. In 2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS) (pp. 1–7). IEEE.
8. Liu, Y. K., Tsai, Y. C., & Chen, S. Y. C. (2026). Q-A3C^2: Quantum reinforcement learning with time-series dynamic clustering for adaptive ETF stock selection. In ICASSP 2026–2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 22012–22016). IEEE.
9. Cheng, L. C., & Sun, J. S. (2024). Multiagent-based deep reinforcement learning framework for multi-asset adaptive trading and portfolio management. Neurocomputing, 594, 127800. https://doi.org/10.1016/j.neucom.2024.127800
10. Khonsha, S., Agha Sarram, M., & Sheikhpour, R. (2023). A profitable portfolio allocation strategy based on money net-flow adjusted deep reinforcement learning. Iranian Journal of Finance, 7(4), 59–89.
11. Liu, J., Gu, X., Feng, H., Yang, Z., Bao, Q., & Xu, Z. (2025). Market turbulence prediction and risk control with improved A3C reinforcement learning. In 2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE) (pp. 2634–2638). IEEE.
12. Abad, Y. P. J., & Moradi, M. (2017). Examine the relationship between the development of financial markets and economic risks. International Academic Journal of Accounting and Financial Management, 4(1), 9–17.
13. Kim, J. B., Heo, J. S., Lim, H. K., Kwon, D. H., & Han, Y. H. (2019). Blockchain based financial portfolio management using A3C. KIPS Transactions on Computer and Communication Systems, 8(1), 17–28.
14. Khonsha, S., Agha Sarram, M., & Sheikhpour, R. (2023). A profitable portfolio allocation strategy based on money net-flow adjusted deep reinforcement learning. Iranian Journal of Finance, 7(4), 59–89.
15. Alavi, S. E., Sinaei, H., & Afsharirad, E. (2015). Predict the trend of stock prices using machine learning techniques. International Academic Journal of Economics, 2(2), 1–11.
16. Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. IEEE Access, 7, 108014–108022. https://doi.org/10.1109/ACCESS.2019.2933607
17. Kang, Q., Zhou, H., & Kang, Y. (2018). An asynchronous advantage actor-critic reinforcement learning method for stock selection and portfolio management. In Proceedings of the 2nd International Conference on Big Data Research (pp. 141–145). https://doi.org/10.1145/3291801.3291819
18. Chen, M., Wang, W., & Hong, C. (2025). Algorithmic trading execution optimization using A3C-LSTM reinforcement learning framework. In Proceedings of the 2025 International Conference on Digital Society and Intelligent Computing (pp. 179–183).
19. Jagadhabi, N. (2025). Explainable AI-driven decision support for strategic risk management in financial services. International Academic Journal of Science and Engineering, 12(3), 125–144. https://doi.org/10.71086/IAJSE/V12I3/IAJSE1254
20. Wójcik, F. (2023). A deep reinforcement learning approach for portfolio optimization and risk management: Case studies. In Analytics in Finance and Risk Management (pp. 133–163). CRC Press.
21. Saepudin, D., & Rauf, K. (2025). Application of deep reinforcement learning for stock trading on the Indonesia Stock Exchange. Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI), 14(1).
22. Venkatarathnam, N., Goranta, L. R., Kiran, P. C., Raju, B. P. G., Dilli, S., Basha, S. M., & Kethan, M. (2024). An empirical study on implementation of AI & ML in stock market prediction. Indian Journal of Information Sources and Services, 14(4), 165–174. https://doi.org/10.51983/ijiss-2024.14.4.26