Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Improving algorithmic trading consistency via human alignment and imitation learning

Yuling Huang, Chujin Zhou, Kai Cui, Xiaoping Lu [*]

School of Computer Science and Engineering, Macau University of Science and Technology, Taipa, Macao Special Administrative Region of China

ARTICLE INFO

Keywords:
Deep Reinforcement Learning
Advantage actor critic
Imitation learning
Human alignment
Algorithmic trading

ABSTRACT

Research on algorithmic trading using reinforcement learning has become increasingly popular in recent years. Although most of the current reinforcement learning methods are employed to train the agent for some kind of modeling or data problem, it is worthwhile to explore in aligning agents with human behavior in applications as crucial as financial trading. Achieving such consistency by incorporating human expert experience into agent behavior is a key for potential improvements in this field. Imitation learning learns directly from examples of humans or other agents performing tasks. However, using imitation learning alone suffers from the problem of transitionally fitting expert example data. By combining the advantages of imitation learning and the Advantage Actor–Critic method, the Human Alignment Advantage Actor–Critic (HA3C) algorithm is proposed, to enhance single-asset trading strategy. First, by adding daily and weekly frequency trading data as input features to TimesNet, which is specifically designed to extract correlated temporal patterns from time-series data, it can capture both short-term and long-term features, thus capturing time-series features more comprehensively. Second, an expert action labeling method is proposed to train a strategy prediction network through supervised learning of behavior imitation. Third, a pre-trained strategy network is transferred to balance the exploration and exploitation of the agent's behavior. Imitation learning techniques leverage finance-specific knowledge to enhance algorithmic trading consistency. This approach enables algorithms to mimic and adapt human decision-making patterns in finance, ultimately improving overall performance. This paper introduces a novel return-based function that efficiently balances short-term and long-term returns over flexible time horizons. It considers the maximum return from different positions and uses flexible time windows to capture trends while maximizing returns. Finally, evaluation on six commonly used datasets, such as DJI and SP500, demonstrates the advantages of the proposed HA3C algorithm compared with other classical and reinforcement learning-based strategies. Notably, on the HSI dataset, the HA3C strategy significantly outperforms other methods, achieving an impressive cumulative return of 681.55% and a Sharpe ratio of 5.07. These results show the superior performance of the HA3C algorithm in enhancing stock trading strategies and its potential to impact algorithmic trading consistency through aligning agent behavior with human expertise.

1. Introduction

The surge in machine learning development has ignited a growing interest in harnessing its capabilities to refine financial trading models and systems. This interest spans various machine learning paradigms, including supervised learning, unsupervised learning, and Reinforcement Learning (RL) (Chou & Nguyen, 2018; Ge, Qin, Li, Huang, & Hu, 2022; Hu et al., 2018; Liu, Zhang, Bao, Yao, & Zhang, 2023; Tran, Pham-Hi, & Bui, 2023; Tsai, Cheng, Tsai, & Shiu, 2018; Vishal, Satija, & Babu, 2021; Xiao, 2023). Notably successful in a number of domains, including video and board games, RL is able to learn through environmental interactions and continuously make decisions.

我也要这么做!

Sequential decision-making in stock trading can be formulated as a Markov Decision Process (MDP) for reinforcement learning to learn financial trading strategies. The environment is the object with which the agent interacts and the rules or mechanisms of the interaction process. However, the main challenge of financial trading is the unpredictability and complexity of market dynamics, which are influenced by numerous factors such as economic indicators, investor sentiment and the behavior of other market participants. In this environment, where traditional trading models often struggle to maintain performance over time due to the inability to adapt to new information or changing market conditions, Deep Reinforcement Learning (DRL) offers a convincing

[*] Corresponding author.
E-mail addresses: 2109853gia30003@student.must.edu.mo (Y. Huang), 3220002751@student.must.edu.mo (C. Zhou), 2109853nia30001@student.must.edu.mo (K. Cui), xplu@must.edu.mo (X. Lu).

https://doi.org/10.1016/j.eswa.2024.124350
Received 11 December 2023; Received in revised form 25 March 2024; Accepted 27 May 2024
Available online 29 May 2024

solution due to its ability to learn from interactions and optimize the decision-making process for cumulative rewards. It provides the adaptive and learning capabilities needed to navigate and exploit the complexity of financial markets to improve trading strategies.

Optimizing cumulative rewards stands as a potent strategy in RL, often hinging on immediate rewards defined by either humans or the environment. The reward function assumes a pivotal role in conveying intricate objectives to autonomous agents, demonstrating effectiveness across diverse domains such as board games, autonomous control, and algorithmic trading (Bellemare, Candido, Castro, Gong, Machado, Moitra, Ponda, & Wang, 2020; Chakole & Kurhekar, 2020; Cornalba, Disselkamp, Scassola, & Helf, 2022; Cui, Hao, Huang, Li, & Song, 2023; Dang, 2020; Felizardo et al., 2022; Huang, Cui, Song, & Chen, 2023; Huang, Li, Mei, & Gong, 2023; Huang, Lu, Zhou, & Song, 2023; Huang & Song, 2023; Huang, Wan, Zhang, & Lu, 2023; Huang, Zhou, Cui, & Lu, 2024; Li, Liu, & Wang, 2022; Lima Paiva, Felizardo, Bianchi, & Costa, 2021; Pavel, Muhtasim, & Faruk, 2021; Silver et al., 2018; Taghian, Asadi, & Safabakhsh, 2022; Théate & Ernst, 2021; Tran et al., 2023). In our previous study, trading strategies were explored critic-only RL algorithms (including State-Action-Reward-State-Action (SARSA) and Deep Q-Network (DQN) algorithms) by incorporating deep networks. Additionally, the integration of Long-Short Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Multi-scale Convolutional Neural Network (CNN), and attention mechanisms was separately investigated to further enhance trading strategies (Cui et al., 2023; Huang, Cui, et al., 2023; Huang, Lu, et al., 2023; Huang, Wan, et al., 2023; Huang et al., 2024). Three distinct methodologies exist within RL: the critic-only approach (Tran et al., 2023; Xiao, 2023), the actor-only approach (Xiao, 2023), and the actor–critic approach (Ge et al., 2022; Vishal et al., 2021). The critic-only approach integrates the Q-value function to learn the optimal action selection policy for discrete action spaces. Conversely, the actor-only approach involves agents independently learning the optimal policy, typically applied in continuous action spaces. The actor–critic model combines the strengths of both approaches, with the actor learning the policy and the critic learning the value function (Diederichs, 2019). This combination makes it well-suited to address challenges in complex dynamic scenarios.

However, the primary goal of trading is to maximize profits while avoiding risks. DRL achieves this by optimizing the maximum expected cumulative return based on future actions. Evaluating RL agents based on the profits generated ensures optimal performance. Therefore, the role of the reward function in communicating objectives to autonomous agents in the RL domain cannot be underestimated.

AlphaGo, a world-renowned example, benefitted from pre-training with 30 million expected moves (Silver et al., 2016), learning from expert experience through Imitation Learning (IL). This process, where agents acquire skills and behaviors by observing demonstrations, finds its inspiration and foundation in neuroscience and is an indispensable component of machine intelligence and human–computer interaction (Schaal, 1999). Learning by imitation allows agents to align with human experts' experiences. Despite the obvious differences between the dynamics of board games and financial trading, this paper draws from AlphaGo and the emerging notion that large language models should be aligned with the human mind, with the aim of providing insights into the potential of IL and DRL techniques in stock trading scenarios.

Motivated by IL and the Advantage Actor-Critic (A2C) approach, this paper proposes the Human Alignment A2C (HA3C) algorithm. It aims to enhance single-asset trading strategies by combining IL and the A2C approach. The proposed method defines expert action labels, trains a strategy prediction network through supervised learning of behavioral clones, and then migrates this network to DRL as an initialization parameter for the strategy network. This integration of IL techniques leverages financial domain knowledge to enhance trading intelligence. Additionally, the paper proposes a new reward-based function, considering both short-term and long-term returns, with flexibility to fine-tune for specific stock characteristics.

The study makes the following major contributions:

- Introducing a novel approach by incorporating daily and weekly trading data into the TimesNet model, which demonstrates its effectiveness in expert strategy networks and DRL applications. TimesNet model and imitation learning in trading. TimesNet is an advanced neural network designed for time series that excels at recognizing patterns and trends in financial markets and making more informed and efficient trading decisions.
- Exploring IL in single-asset trading, which utilizes the TimesNet expert strategy network to identify trading opportunities and adjust risk according to expert decision data. IL is a method to mimic the decisions of human experts, thereby improving the ability to recognize profitable trading opportunities and enhancing risk management strategies based on expert insights
- Combining IL and RL enables agents to utilize experiential data from human experts for exploration and exploitation, which accelerates the convergence of RL algorithms.
- Introducing a novel return-based function that effectively balances short-term and long-term returns over a flexible time horizon. The function considers the maximum returns of various positions and employs a flexible time window to capture trends from both short-term and long-term perspectives while maximizing returns.
- Evaluations on six commonly used datasets show that the proposed HA3C algorithm significantly outperforms various classical and DRL-based strategies. These results highlight the effectiveness of the algorithm in enhancing stock trading strategies and the potential to significantly impact algorithmic trading consistency by aligning agent behavior with human expertise.

This paper is structured as follows. In Section 2, previous studies on the topic are reviewed. Section 3 explains the proposed method in detail. Experimental results are described and analyzed in Section 4. Finally, Section 5 summarizes the results and shows future works.

2. Related work

Traditional trading strategies, commonly adopted by human traders, such as buy and hold (B&H), sell and hold (S&H), trend following with moving averages (TF), and mean reversion with moving averages (MR), have long been the dominant trading strategies in the financial markets. While these strategies have proven to be time-tested, they often fail to capture the variety of opportunities presented by rapidly changing market dynamics. In contrast, recent research has shown that trading strategies derived from DRL are significantly better than these traditional methods (Cui et al., 2023; Huang, Cui, et al., 2023; Huang, Lu, et al., 2023; Huang, Wan, et al., 2023; Huang et al., 2024). DRL strategies excel at adapting to complex market conditions, utilizing large data sets to uncover patterns and insights that traditional methods cannot reach. However, the rise of AI methods and other advanced AI approaches presents a serious challenge: the problem of interpretability. Despite the outstanding capabilities of these AI-driven strategies, they often operate in a "black box", making it difficult to discern the rationale behind their decisions. This lack of transparency is evident in state-of-the-art models such as GPT and other large-scale AI systems, highlighting the urgent need for a mechanism to bring these powerful tools in alignment with human understanding and logic. With these considerations in mind, this paper attempts to bridge this gap by exploring IL and DRL techniques in the field of stock trading. By merging DRL with IL, this paper aims to create innovative trading strategies that not only outperform traditional methods, but also resonate with human traders by providing greater transparency and understandability.

To facilitate an organized discussion of the literature related to the proposed approach, this section is divided into three distinct branches: imitation learning, human alignment, and reward designing. This categorization helps to present research in these different areas in a coherent and systematic way, thus allowing for a more focused and organized examination of related work.

2.1. Imitation learning

IL has emerged as a widely studied paradigm across various domains, including robotics, navigation, game-play, and other applications (Hussein, Gaber, Elyan, & Jayne, 2017). In recent advancements, neural network-based IL methods have gained prominence, with notable examples such as one-shot IL (Duan et al., 2017; Finn, Yu, Zhang, Abbeel, & Levine, 2017), third-person IL (Stadie, Abbeel, & Sutskever, 2017), and Generative Adversarial IL (GAIL) (Ho & Ermon, 2016). These methods leverage the power of neural networks to enhance learning from demonstrations.

The intersection of IL with financial tasks has yielded notable contributions. Goluža, Bauman, Kovačević, and Kostanjčar (2023) explored the application of imitation learning methods in the context of financial activities, indicating the versatility of IL approaches in addressing challenges within the financial domain. For instance, Liu, Liu, Zhao, Pan, and Liu (2020) presented iRDPG, a novel approach that integrates trading agents, DRL, and imitation learning techniques. This model showcases a fusion of DRL and IL to automatically develop quantitative trading strategies. The incorporation of imitation learning allowed the model to assimilate classical trading strategies, finding a delicate balance between exploration and exploitation in the dynamic trading environment. Park and Lee (2021) proposed SIRL-Trader, an algorithmic trading method designed for profitability through long positions. This approach integrated offline and online state representation learning with imitative reinforcement learning. In the offline state representation learning phase, robust features were extracted through dimensionality reduction and clustering. The online state representation learning phase involved co-training a regression model with a reinforcement learning model, providing accurate state information for decision-making. The imitative reinforcement learning component combined behavior cloning with the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, incorporating multi-step learning and dynamic delay to enhance TD3's performance.

Expanding the scope of IL to valuation mechanisms, Peng and Lee (2023) developed an imitation learning mechanism utilizing a teacher to demonstrate state–action pairs. This approach employs guided policy search with a Levy distribution ordered chronologically, capturing stages of growth, decay, and a long fat tail corresponding to a business life cycle. By learning the trajectory of free cash flow and optimizing its fitness with the chronological order of a Levy distribution, the method effectively monitored data for the correct stage, showing the versatility of IL in financial applications.

These studies collectively highlighted the multi-functionality and effectiveness of imitation learning techniques, showing their application across diverse domains and emphasizing their potential in enhancing decision-making processes in financial tasks and algorithmic trading.

2.2. Human alignment

The concept of human alignment in artificial intelligence (AI) research has garnered considerable attention for its effectiveness in guiding machines to emulate human behavior and decision-making. AlphaGo's landmark victory over world Go champion Lee Sedol marked an early triumph in this arena. This success was achieved through the integration of deep CNNs and RL, supplemented by learning from human game histories and supervised learning to predict expert moves. The strategy is further enhanced by self-pairing games, laying the foundation for subsequent developments in human alignment methods (Silver et al., 2016).

One such advancement is AlphaGo Zero, which unlike its predecessor's reliance on human expert data, uses an approach based purely on RL, devoid of any human-derived data, guidance, or domain-specific insights beyond the fundamental rules of Go (Silver et al., 2017). Despite this, the need for expert data persists in many fields, highlighting the ongoing challenge of aligning AI with human values—the problem of value alignment in large language models being one prominent issue.

Following these developments, the field has introduced a variety of human-aligned models that utilize human feedback for improvement. For example, the human aggregated actor–critic (COACH) algorithm and the incorporation of human preferences into complex RL tasks, have demonstrated the potential of incorporating human insights into the AI training process (Christiano et al., 2017; MacGlashan et al., 2017). In addition, the application of example datasets has promoted the acceleration of DQN frameworks, emphasizing the importance of human data in augmenting machine learning models (Hester et al., 2017).

Recent researches, such as InstructGPT and human-aligned trading models, have taken the human-aligned paradigm to new heights. These models utilized human feedback for fine-tuning and employ unique loss functions to explicitly incorporate human behavior into algorithmic trading (Ouyang et al., 2022; Ye & Schuller, 2023). In particular, the Human Aligned Trading (HAT) model proposed by Ye and Schuller (2023), which aimed to align machine trading agents with human traders, was noteworthy. The model had a unique multi-loss function that integrates supervised learning, single- and multi-step Q-learning, and imitation learning in both the training and trading phases. The results showed that HAT outperforms the baseline model, especially in small-cap trading, where it reduced trade frequency and associated costs regardless of market conditions.

Overall, these efforts highlight the intrinsic value of combining machine learning models with human insights and strategies to achieve superior performance. The development of human-alignment approaches in AI research has not only enhanced the capabilities of autonomous systems in a variety of domains but has also been remarkably successful in navigating complex financial markets.

2.3. Reward designing

The formulation and design of reward functions in the context of single-asset trading strategies have been a focal point in the existing literature, with researchers adopting diverse approaches to shape these functions (Chakole & Kurhekar, 2020; Chakraborty, 2019; Chen & Gao, 2019; Corazza, Fasano, Gusso, & Pesenti, 2019; Corazza & Sangalli, 2015; Cornalba et al., 2022; Dang, 2020; Gao, 2018; Huang, 2018; Jeong & Kim, 2019; Li et al., 2022; Liu, Zhang, et al., 2023; Ma, Zhang, Liu, Ji, & Gao, 2021; Si, Li, Ding, & Rao, 2017; Xiao, 2023).

Chakole and Kurhekar (2020), Chen and Gao (2019), Gao (2018), Huang (2018), Jeong and Kim (2019), Lei, Zhang, Li, Yang, and Shen (2019), Ma et al. (2021) adopted a reinforcement learning method and built a manually designed return as a reward function. The manually designed returns of these researchers were a key component of the reward function. This design choice was based on the fact that rewards incorporated the financial gains or losses associated with trading behavior and were a tangible and interpretable measure of strategy performance.

Conversely, a subset of researchers, including Chakraborty (2019), Dang (2020), Lei et al. (2019), Liu, Zhang, et al. (2023), Si et al. (2017), Xiao (2023), chose to focus on profit as the primary element of the reward function. By focusing on profits, these studies aimed to directly capture the monetary gains obtained through the implementation of trading strategies. This simplification was motivated by the direct link between profit and the primary goal of the trading strategy, i.e., generating financial payoffs.

A significant portion of the literature considers the problem from a more holistic perspective, incorporating both return and risk into the return function. This approach, exemplified by the use of Sharpe ratios (Corazza et al., 2019; Corazza & Sangalli, 2015; Cornalba et al., 2022; Li et al., 2022; Tran et al., 2023), reflected the recognition that effective trading strategies must not only generate returns but also manage the associated risks. The Sharpe Ratio, calculated as the ratio of a strategy's average return to its standard deviation, provided a balanced assessment of both the profitability and risk mitigation aspects of a trading strategy.

In summary, the diversity of approaches to reward function design in existing research emphasizes the nuanced considerations involved in developing appropriate metrics to assess the effectiveness of single-asset trading strategies. Researchers have drawn on manual designs, profit-centered approaches, and holistic measures such as Sharpe ratios, reflecting a concerted effort to capture the multifaceted nature of financial markets and trading dynamics.

Table 1
Actual trading operations based on the signal and the current account position.

Position (POS_t)	Signal (a_t^*)	Actual action (a_t)	Description
0	0	0	Hold the cash.
0	1	1	Open a long position.
0	−1	−1	Open a short position.
1	0	0	Hold the long position.
1	1	0	Hold the long position.
1	−1	−1	Close the long position.
−1	0	0	Hold the short position.
−1	1	1	Close the short position.
−1	−1	0	Hold the short position.

3. Method

In algorithmic trading, the selection and execution of trading strategies can be formulated as a reinforcement learning problem. Agents learn optimal trading strategies and execution methods by exploring various actions in the trading environment. As this paper delves into the complexity of stock trading decisions, it is important to recognize the need to make certain reasonable assumptions to ensure the validity of the conclusions drawn. First, and most importantly, this paper assumes that the market is fully efficient, meaning that market prices comprehensively incorporate all available information, independent of external factors affecting the market. Second, Due to the intricacies of stock market liquidity and the relatively small size of the total assets under consideration, especially in the context of retail investors. Given the relatively small impact of individual buy and sell orders on market prices, the model presented in this paper operates under the assumption that trade orders executed by agents have a negligible impact on market prices. In addition, there is a key assumption that there is no slippage in the execution of orders, which implies that all market assets are sufficiently liquid to complete trades at the final price. Only under these assumptions can the conclusions and results be considered reasonable and robust.

3.1. Overview of the proposed method

This section provides a brief overview of the proposed approach, Human Alignment Advantage Actor–Critic (HA3C). As shown in Fig. 1, the framework integrates imitation learning and reinforcement learning to create a unified model that facilitates training using standard reinforcement learning methods. The system is divided into three key parts. First, data preprocessing is performed to process daily trading data into daily and weekly trading data. Daily data captures micro-differences in short-term market movements and can identify immediate trends, volatility, and potential short-term trading opportunities. Weekly data summarizes daily movements over the course of a week to highlight long-term, ongoing trends. Secondly, labeling is performed through the proposed expert labeling method and the task of pre-training a network of strategies based on imitation learning. For this purpose, the A2C pre-trained policy network is trained on top of the TimesNet network by means of a classification-supervised learning approach. This initial phase lays the foundation for the subsequent phases of the method. Subsequently, the pre-trained strategy network is integrated into the A2C algorithm to facilitate the learning of effective trading strategies. This combination is inspired by the good results demonstrated by the A2C algorithm introduced by Mnih et al. (2016b), which is specifically adapted to single-asset decision problems.

3.2. Problem formulation

3.2.1. State

At each time step t, the agent views the state of the stock market as s_t. It is regarded as a series of information collected at previous n time steps and newly obtained at t. Here, the opening price, high price, low price, closing price and trading volume for the previous $n = 20$ days and weeks are taken as the state of t. The state s_t is denoted mathematically as Eq. (1):

$$s_t = \{O_{t'}^d, H_{t'}^d, L_{t'}^d, C_{t'}^d, V_{t'}^d, O_{t'}^w, H_{t'}^w, L_{t'}^w, C_{t'}^w, V_{t'}^w\}_{t'=t-n}^{t}, \tag{1}$$

where $O_{t'}^d, H_{t'}^d, L_{t'}^d, C_{t'}^d, V_{t'}^d$ are the stock market prices for the opening, highest, lowest, closing and total trading volume in the daily period $[t-n, t]$. Similarly, $O_{t'}^w, H_{t'}^w, L_{t'}^w, C_{t'}^w, V_{t'}^w$ are the stock market prices for the opening, highest, lowest, closing and total trading volume in the weekly period $[t-n, t]$.

3.2.2. Action

At each discrete time step t, the reinforcement learning agent first observes the current state of the environment s_t. Subsequently, guided by the RL strategy $\pi(a_t|s_t)$, the agent chooses an action a_t. In this study, it focuses on a trading strategy that involves buying and selling a single security in the discrete action space of a financial market. The strategy consists of different phases such as opening, holding and closing a position. As a result, the action is computed by Eq. (2):

$$a_t^* = \begin{cases} -1, & \text{if } \pi(a_t|s_t) = \text{Sell}; \\ 0, & \text{if } \pi(a_t|s_t) = \text{Hold}; \\ 1, & \text{if } \pi(a_t|s_t) = \text{Buy}. \end{cases} \tag{2}$$

where $\pi(a_t|s_t) = \text{Sell}$, $\pi(a_t|s_t) = \text{Hold}$, and $\pi(a_t|s_t) = \text{Buy}$ are determined by the policy network at each time step t.

Table 1 summarizes the actual trading operations determined by a combination of the current account positions and trading signals. When the account holds no position ($\text{POS}_t = 0$), a buy ($a_t^* = 1$) or sell ($a_t^* = -1$) signal leads to the establishment of a long or short position, respectively, while a neutral signal ($a_t^* = 0$) leads to holding cash. For accounts already holding a long position ($\text{POS}_t = 1$), a sell signal prompts the closing of the position, while a buy or neutral signal indicates maintaining a long position. Conversely, if holding with short positions ($\text{POS}_t = -1$), a buy signal triggers the closing of a short position, while a sell or neutral signal indicates the continuation of a short position. Here, POS_t denotes a potential short, neutral or long position, which is equivalent to −1, 0 and 1, respectively; and a_t denotes that the possible actions are sell, hold or buy, which is equivalent to −1, 0 and 1, respectively.

3.2.3. Reward

The significance of the reward function within the reinforcement learning framework is pivotal, quantifying the immediate feedback or reward received by a trading agent based on its actions in a given state. In algorithmic trading, this function typically serves to mirror the profitability or performance of a trading strategy. This paper introduces a novel return-based function that effectively balances short-term and long-term returns across a flexible time horizon. This function considers the maximum return of various positions and incorporates a flexible time window, facilitating the capture of trends from both short-term and long-term perspectives while maximizing overall returns. Remarkably, the reward function's flexibility enables fine-tuning for either short-term or long-term trends through the adjustment of the

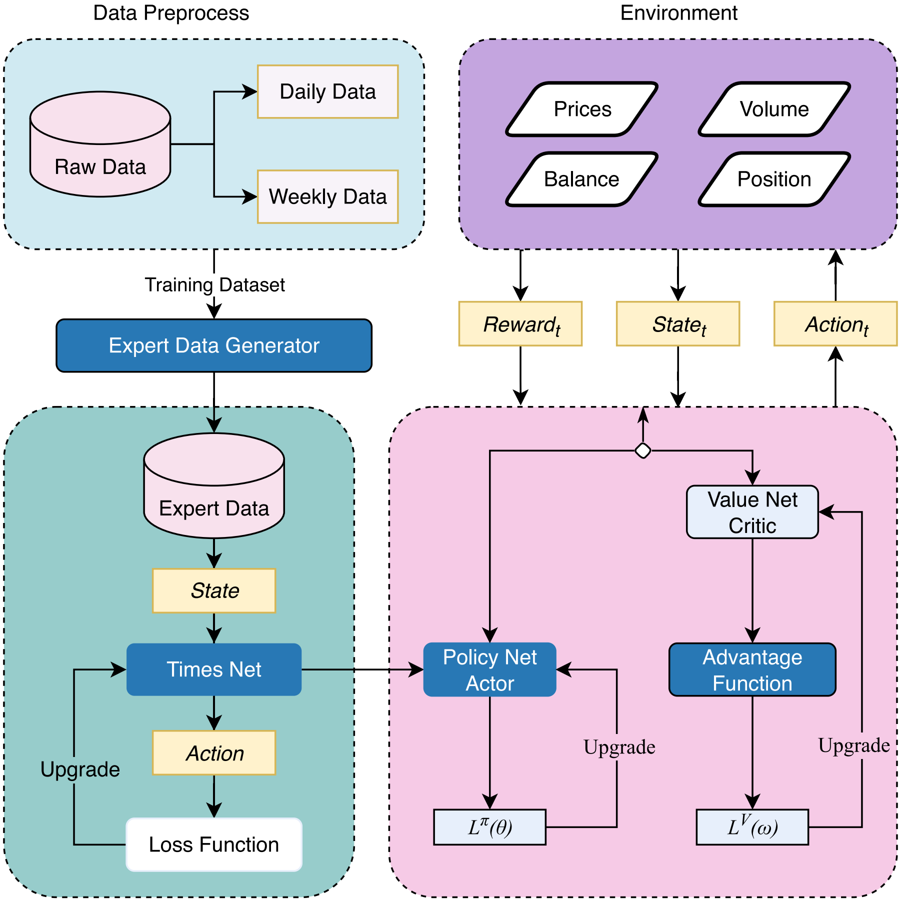


Fig. 1. The structure of the proposed HA3C.

parameter, time horizons m, tailoring it to the distinctive characteristics of the stocks under consideration. Furthermore, the designed reward function places emphasis on information pertinent to the agents' positions. Mathematically, the proposed reward function is expressed as Eqs. (3) to (5):

$$R_t = \begin{cases} \mathrm{POS}_t * \mathrm{maxRatio}, & \begin{array}{l}\mathrm{maxRatio} > 0 \text{ or} \\ \mathrm{maxRatio} + \mathrm{minRatio} > 0,\end{array} \\ \mathrm{POS}_t * \mathrm{minRatio}, & \begin{array}{l}\mathrm{minRatio} < 0 \text{ or} \\ \mathrm{maxRatio} + \mathrm{minRatio} < 0.\end{array} \end{cases} \tag{3}$$

where

$$\mathrm{maxRatio} = \begin{cases} \max(r_{t+1}, r_{t+2}, \dots, r_{t+m}), & \text{if } r_{t+i} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

$$\mathrm{minRatio} = \begin{cases} \min(r_{t+1}, r_{t+2}, \dots, r_{t+m}), & \text{if } r_{t+i} < 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where $r_{t+i} = \frac{p_{t+i} - p_t}{p_t} * 100$.

3.3. Pre-trained policy network

Behavioral cloning methods represent a category of direct RL techniques that establish a mapping from states to actions without the explicit recovery of the reward function (Bain & Sammut, 1995). Formulated as a supervised learning problem, these methods utilize a dataset $D = \{(s_i, a_i)\}_{i=1}^{N}$ consisting of state–action pairs obtained from an expert, aiming to derive the optimal policy π^* (Ross & Bagnell, 2010). In this context, the policy operates as a regressor or classifier. The efficiency of behavioral cloning methods lies in their capacity to learn without direct interaction with the environment, as they deviate from the traditional MDP formulation.

The application of IL in finance addresses the challenge of designing investment strategies across diverse assets and timeframes. This is achieved by leveraging historical trading data from experts, which may include human actors (such as fund managers or proprietary traders) or an oracle with complete information encompassing both past and future data (Dixon, Halperin, & Bilokon, 2020). The advancement of machine learning, coupled with increased computational power and data availability, has facilitated significant progress in imitation learning over the last decade. IL algorithms, designed to learn optimal policies from expert demonstrations, have proven effective in financial applications, outperforming other statistical approaches in identifying investment behavior.

In this paper, the TimesNet network is adapted to make it suitable for behavioral cloning scenarios. Based on TimesNet, the activation function in the last layer is modified to a Softmax function to facilitate state-to-action (Buy, Hold, Sell) mapping. The network structure is illustrated in Fig. 2. Daily and weekly data are both 5-dimensional data and have the same shape, so they can directly concatenate to become 10-dimensional data. Then, to enrich the detailed information of the data, the token embedding operation is used to expand the length of the data. At the same time, the position encoding information generated by the position embedding operation is also added to the token. Finally, dropout processing is used to randomly discard some information to increase the network's information judgment ability and reduce overfitting. The processed feature map will be extracted through TimesBlock. The final high-dimensional features are obtained through the GELU activation function and dropout layer, and are sent to the

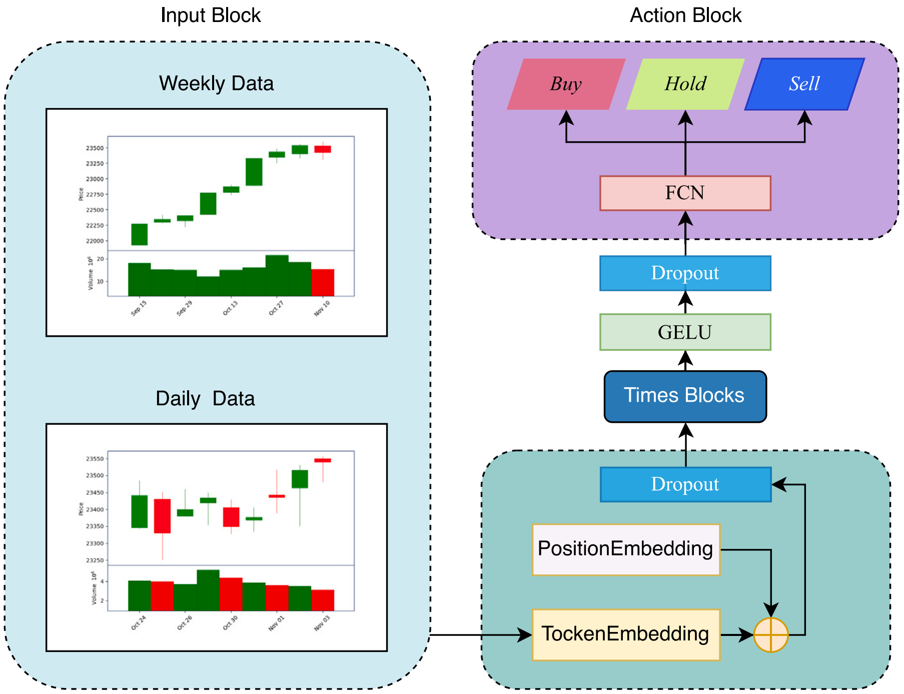


Fig. 2. The pre-trained policy network based on imitation learning.

fully connected layer to output the probability distribution of the buy, sell and hold.

In addition, a robust and rational approach is proposed for generating expert example data for applying IL. Through the utilization of the up-down labeling method and flexible parameter adjustments, the proposed approach captures both short-term and long-term stock price trends, producing high-quality expert example data for training the IL agent in effective trading strategies. Weekly data and daily data are regarded as inputs with the same weight, and the data length is exactly the same, so that they can be integrated into a whole feature input to the subsequent network. The state, denoted as s_t, includes the open price, low price, high price, and trading volume for the preceding ten days, forming a comprehensive representation for the agent to capture crucial market information. Subsequently, the action a_t is determined based on the observed state and predefined rules. The approach aligns with the widely used up-down labeling method for stock trend forecasting, categorizing stock price movements as "upward" (0) when the future stock price $(t + m)$ is higher than the current price (t) and as "downward" (1) when it is equal to or lower. For example, if maxRatio > 1.50, a buy action will be given from an expert's perspective. Importantly, this method accommodates various time horizons (m) to capture both short-term and long-term trends, offering flexibility by allowing adjustments to the parameter m for tuning sensitivity to short-term or long-term trends based on stock characteristics. Figs. 3 to 5 illustrate different trend-based labeling of expert actions for imitation learning. The action label is expressed as Eq. (6):

$$a_t = \begin{cases} 1\text{buy}, & \text{if maxRatio} > 1.50, \\ -1\text{sell}, & \text{if minRatio} < -1.50, \\ 0\text{hold}, & \text{otherwise.} \end{cases} \tag{6}$$

3.4. HA3C

In this paper, the HA3C method is proposed to explore single-asset trading strategies based on A2C. A2C is a synchronous variant of the Actor–Critic algorithm that employs multiple parallel agents to enhance learning efficiency and stability (Mnih et al., 2016b). A2C is situated

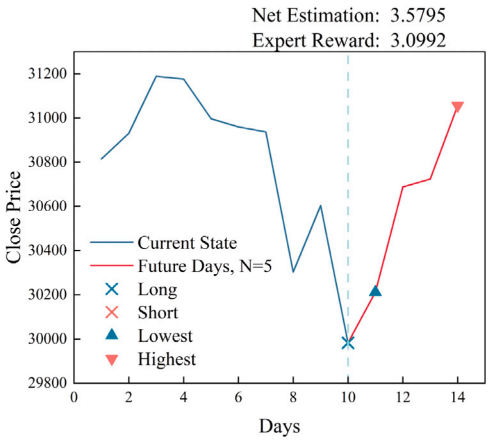


Fig. 3. Assigning labels to expert actions for imitation learning in upward market conditions.

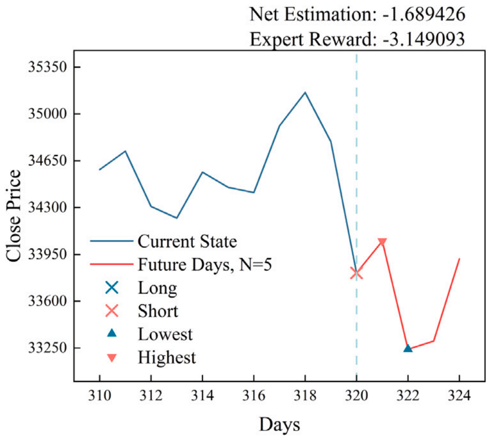


Fig. 4. Assigning labels to expert actions for imitation learning in stationary market conditions.

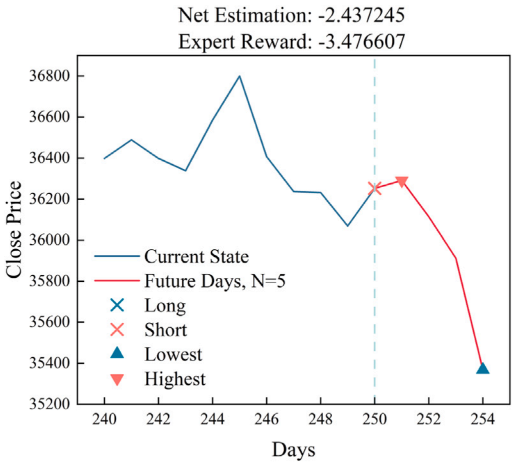


Fig. 5. Assigning labels to expert actions for imitation learning in downward market conditions.

within the broader field of reinforcement learning, where agents learn decision-making by interacting with an environment and receiving feedback in the form of rewards or punishments. The Actor–Critic architecture introduces a separation between the policy (actor) and the value function (critic). The actor selects actions based on the policy, while the critic evaluates actions by estimating expected rewards, thereby reducing the high variance associated with pure policy gradient methods. The Advantage function, representing the advantage of a specific action in a given state compared to the average action, plays a crucial role in addressing high variance. A2C combines advantages from both policy gradient methods and value-based methods by using the Advantage function to simultaneously update the policy and value function.

Advantage function: The advantage function $A_\pi(s_t, a_t)$ is the difference between the observed reward and the expected value of being in a particular state (Mnih et al., 2016a). Mathematically, it is defined as Eqs. (7) to (8):

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t), \tag{7}$$

and

$$Q_\pi(s_t, a_t) = r_t + \gamma \cdot V_\pi(s_{t+1}), \tag{8}$$

where the state value function $V_\pi(s_t)$ is the sum of the action value functions corresponding to all possible actions in state s_t multiplied by the probability of taking that action. γ is the discounted factor in reinforcement learning algorithm. The action value function $Q_\pi(s_t, a_t)$ is the value function corresponding to action a_t in state s_t. The relationship between action value function and state value function can be represented as Eq. (8) If $A_\pi(s_t, a_t) > 0$, the action is better than the average action; if $A_\pi(s_t, a_t) < 0$, the action is not as good as the average action.

Actor: The actor determines the policy, defining the probability distribution over actions given a state. In A2C, this is often represented as Eq. (9):

$$\pi(a_t|s_t; \theta) = \mathbb{P}(A_t = a_t|S_t = s_t; \theta), \tag{9}$$

where a_t is the action taken by the agent, s_t is the current state and θ is the parameter of actor network. The actor's parameter θ update is given by Eq. (10):

$$\theta \leftarrow \theta + \beta \cdot A_\pi(s_t, a_t) \cdot \nabla_\theta \ln \pi(a_t|s_t; \theta), \tag{10}$$

where β is the learning rate.

Critic: The critic estimates the value function, representing the expected cumulative reward of being in a state and following a policy. The critic is typically implemented as a neural network. The critic's weight update is given by Eq. (11):

$$w \leftarrow w - \alpha A_\pi^2(s_t, a_t) \cdot \nabla_w V(s_t; w), \tag{11}$$

where ω is the parameter of critic network and α is the learning rate.

In this paper, the TimesNet is adopted as Value function's, Actor's, and Critic's network, respectively. TimesNet is a modular neural network architecture that transforms 1D time series data into 2D tensors to effectively capture temporal variations using an efficient inception block. The TimesNet network uses CNN and Fourier transform techniques for feature extraction, specifically designed to extract relevant temporal patterns from data, thereby enhancing the generalization ability of our models across different market conditions. In this study, TimesNet is modified appropriately to fit the task, which is the output of the network. For the actor, the network needs to output a three-dimensional tensor corresponding to the agent's three actions: buy, sell and hold. Therefore, the output of TimesNet used in the actor has beshuen modified to be three-dimensional. For the critic, the network only needs to output a single Q-value, hence, the output of TimesNet used in the critic is one-dimensional. The A2C algorithm updates the policy (actor) and the value function (critic) simultaneously. The policy is updated using the advantage-weighted log probability of the chosen action, encouraging actions that lead to higher-than-expected rewards. The value function is updated to reduce the difference between the estimated and observed rewards.

3.5. HA3C training

The training procedure for this framework is presented in Algorithm 1. Initially, a dataset that consisting of state–action pairs obtained from an expert is introduced as the training set of behavioral cloning network θ. By calculating the loss of estimated action from behavioral network and actual action, the parameter of θ is updated with gradient descent. When the training of behavioral network is finished, the parameter of θ will be used to initialize the parameter of policy network. After that, a sequence of market features that matches the window length is randomly sampled from real data to create an initial condition for a trading event. Agent takes actions and receives corresponding rewards based on the current environment state. At the end of the trading process, the collected market states and trading behaviors are used to optimize the trading strategy by updating the parameters of the agent network.

4. Experiments

4.1. Dataset

In the experiment, six stock indices from diverse countries and regions. The selected country/region stock indices encompass the Hang Seng (HSI) in Hong Kong, CAC40 (FCHI) in France, Nikkei 225 (N225) in Japan, as well as the Nasdaq Composite (IXIC), S&P 500 (SP500), and Dow Jones Industrial Average (DJI) in the United States. The temporal scope of the datasets spans from January 1, 2007, to December 31, 2022, providing an extensive dataset for model training.

The entire dataset is partitioned into training and testing sets, each comprising five-dimensional time series data for every index. This includes daily opening, low, high, and closing prices, along with trading volume. Fig. 6 visually depicts the closing prices for each dataset. The training set encompasses the period from January 1, 2007, to December 31, 2020, serving as the interval for training the model parameters. On the other hand, the testing set spans from January 1, 2021, to December 31, 2022, facilitating the evaluation of the models' performance.

Algorithm 1: HA3C algorithm

```
Input: Learning rate $\alpha_\omega, \alpha_\theta, \beta$, discount factor $\gamma$, training set $D = \{(s_i, a_i)\}$, replay buffer $B$, buffer size $m$, minimal replay buffer size $M$;
Initialize behavioral cloning network parameter with random initial weights $\theta_p \leftarrow \theta_0$;
for $N = 0, 1, 2, \cdots$ do
    Select sample $(s_i, a_i)$ from dataset D;
    Predict $\hat{a}_i$ with $\theta(s_i)$;
    Calculate loss $L_\theta = \text{CrossEntropyLoss}(a_i, \hat{a}_i)$;
    Calculate gradient $\delta_\theta = \frac{\partial L_\theta}{\partial \theta}$;
    $\theta_p = \theta_p - \beta * \delta_\theta$;
end
Initialize policy network parameter with parameter of behavioral cloning network $\theta \leftarrow \theta_p$;
Initialize target network parameter with random initial weights $\omega \leftarrow \omega_0$;
for $k = 0, 1, 2, \cdots$ do
    Initialize state $s_t$;0
    Perform policy $\pi_\theta$, store $\{s_t, a_t, r_t, s_{t+1}\}$ into replay buffer $B$;
    if $m$>=$M$ then
        Sample trajectory $\{s_t, a_t, r_t, s_{t+1}\}$ from replay buffer $B$;
        Estimate advantage function: $\hat{A}_t = r_t + \gamma V_\omega^{\pi_\theta}(s_{t+1}) - V_\omega^{\pi_\theta}(s_t)$, where $V_\omega^{\pi_\theta}$ is the state-value function;
        Calculate $L_\theta = \sum_t \log\pi_\theta(a_t, s_t)\hat{A}_t$;
        Calculate gradient $\delta_\theta = \frac{\partial L_\theta}{\partial \theta}$;
        Calculate $L_{V_\omega^{\pi_\theta}}(\omega) = \sum_t \hat{A}_t^2$;
        Calculate gradient $\delta_\omega = \frac{\partial L_{V_\omega^{\pi_\theta}}(\omega)}{\partial \omega}$;
        $\omega = \omega - \alpha_\omega * \delta_\omega$;
        $\theta = \theta + \alpha_\theta * \delta_\theta$;
    end
end
```

4.2. Evaluation metrics

In the context of this paper, several key metrics to assess the performance of investments have been employed. Firstly, Cumulative Return (CR) signifies the total return on an investment over a specified time period. Secondly, Annualized Return (AR) denotes the average rate of return earned on an investment over a given time frame, expressed as a percentage per year. Thirdly, the Sharpe Ratio (SR) serves as a metric to evaluate the risk-adjusted return of an investment or investment portfolio. The Sharpe Ratio measures the excess return earned over a risk-free rate per unit of volatility or total risk. Lastly, Maximum Drawdown (MDD) indicates the potential downside risk of an investment or portfolio. It gauges the largest percentage drop from a peak to a trough in the value of an investment or investment portfolio over a specific period of time. Together, these evaluation metrics provide a comprehensive assessment of various facets of investment performance, covering aspects such as returns, risk, and volatility.

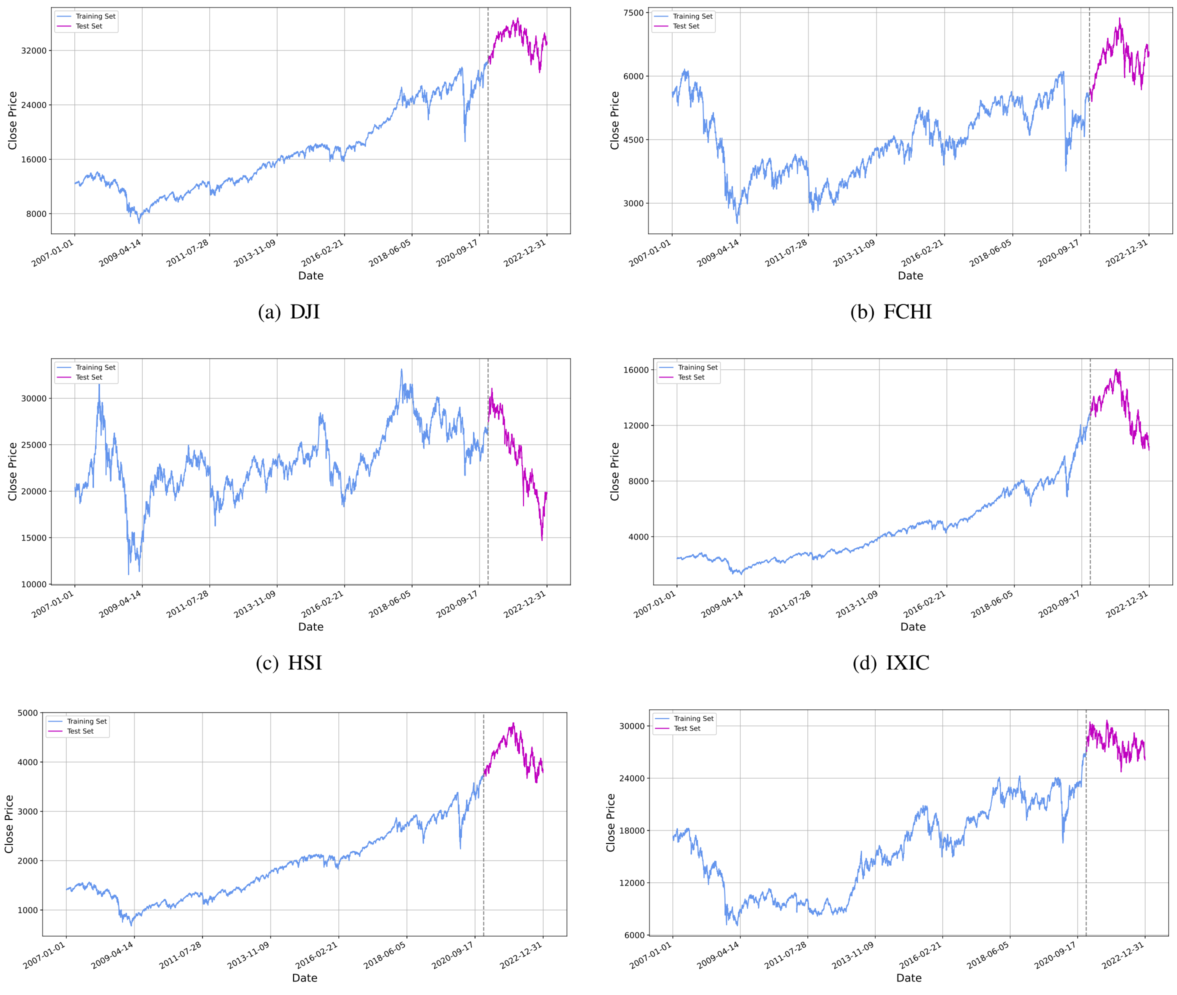
4.3. Baseline methods

To objectively evaluate the performance of the HA3C algorithm, it was compared with DRL methods including TDQN, MLP-Vanilla, DQN-Vanilla, and traditional trading strategies, including B&H, S&H, MR and TF.

- **B&H:** This strategy entails an investor selecting an asset and establishing a long position at the onset of the investment period. The asset is retained until the conclusion of the period, irrespective of any fluctuations in price.
- **S&H:** Analogous to B&H, this strategy involves an investor selecting an asset and initiating a short position at the beginning of the investment period, maintaining the position through to the period's end, regardless of price movements.
- **MR:** Based on the principle that asset prices eventually revert to historical averages, mean reversion strategies attempt to capitalize on this trend. This study uses a 10-day Simple Moving Average (SMA) to identify mean reversion opportunities.
- **TF:** TF represents a quantitative investment strategy within algorithmic trading, leveraging Moving Averages to identify potential trading opportunities. These strategies typically utilize moving averages as indicators of such trends, with this paper specifically employing both 10-day and 20-day SMAs for this purpose.
- **TDQN:** The TDQN model, introduced by Théate and Ernst (2021), addresses the challenge of ascertaining the optimal trading position within stock market transactions. It integrates a Deep Q-Network (DQN) augmented with a five-layer fully connected Q-network.
- **MLP-Vanilla:** Proposed by Taghian et al. (2022), the MLP-Vanilla approach generates trading rules based on stock Open-High-Low-Close (OHLC) data. The Q network of MLP-Vanilla is a encoder–decoder structure consist of a two-layer fully connected network.
- **DQN-Vanilla:** Similarly introduced by Taghian et al. (2022), the DQN-Vanilla model devises trading rules derived from stock OHLC data. The Q network of DQN-Vanilla is a two-layer fully connected network.

4.4. Experimental setup

The proposed method undergoes individual training on six distinct stock indices datasets, each corresponding to a different region. Subsequent to the training phase, the model's performance is evaluated using a dedicated test set associated with its respective stock index dataset. This evaluative approach enables the measurement of the model's accuracy in predicting diverse behaviors across various stock markets, achieved through training on multiple datasets. The finely tuned hyperparameters for each agent in this experiment are detailed in Table 2. The hardware and software used in this paper are as follows: processor is 13th Gen Intel(R) Core(TM) i7-13700KF, GPU is NVIDIA RTX 4090 with the 24 GB memory. The software versions used are: Python is v3.10, Pytorch is v1.12, and cuda is v12.1. The system is Windows 10.



(a) DJI

(b) FCHI

(c) HSI

(d) IXIC

(e) SP500

(f) N225

Fig. 6. Close price curve of six stock indices.

Table 2
The tuned hyperparameters.

Hyperparameter	HA3C
Number of CNN layers	13
Number of linear layers	2
Activation function	GELU
Learning rate (α)	0.001
Replay memory size	1000
Discount factor	0.9
Window size	20

4.5. Results and analysis

4.5.1. Comparison with baselines

To assess the effectiveness of the proposed approach, a comparative analysis of the testing results of HA3C agent against the previously mentioned baselines has been conducted. The metrics considered for this evaluation, as presented in Table 3, encompass CR, AR, MDD, and SR, collectively providing a comprehensive view of the performance across all the methods under consideration.

Notably, the proposed method, HA3C, demonstrates superior performance compared to all the baseline methods in terms of cumulative return, maximum drawdown ratio, and Sharpe ratio. The proposed method has consistently yielded CR exceeding 200% across all six datasets, whereas none of the baselines used for comparison achieved a CR surpassing 100%. The most significant disparity in CR occurs in the HSI dataset. In this context, HA3C achieves an impressive CR of 681.55%, while the best-performing baseline, MLP-Vanilla, lags significantly behind with a CR of only 65.92%. This outcome indicates that HA3C effectively assimilates relevant information during the training period, enabling it to derive optimal actions and consequently achieve enhanced performance. The observed outperformance across these key metrics highlights the efficacy of the proposed method in learning and leveraging actionable insights from the training data.

4.5.2. Analysis of pre-trained policy networks

As previously discussed, the proposed methodology involves the integration of a pre-trained behavioral cloning network into the actor network of the A2C method through the application of transfer learning. The primary responsibility of the behavioral cloning network lies in the accurate interpretation of the corresponding state representation and the prediction of ensuing actions. In order to make an informed selection, a comparative analysis of three networks renowned for their exceptional performance in financial time series applications—specifically, TimesNet (Wu et al., 2022), WFNet (Liu, Wu, et al., 2023), and DLinear (Zeng, Chen, Zhang, & Xu, 2023), has been conducted. This comparative evaluation was based on their respective abilities to predict trading actions across diverse datasets, ultimately guiding the final choice of network for implementation in the HA3C method.

Concurrently, a comparative assessment of these networks on the same dataset was undertaken, examining their performance with daily data as well as a combined dataset featuring both daily and weekly

Table 3
The performance of various trading approaches on six datasets.

Datasets	Metrics	Buy and Hold	Sell and Hold	MR	TF	TDQN	MLP-Vanilla	DQN-Vanilla	HA3C
DJI	CR	9.84%	−10.41%	−16.45%	−34.44%	7.57%	40.08%	31.67%	**217.64%**
	AR	9.35%	−5.95%	−12.75%	−33.49%	4.85%	24.44%	20.83%	85.75%
	SR	0.39	−2.00	−0.54	−1.40	0.31	1.42	0.99	4.50
	MDD	21.39%	−5.95%	−12.75%	40.68%	21.66%	10.00%	15.10%	4.34%
FCHI	CR	21.43%	−22.02%	−22.79%	−12.40%	45.16%	42.9%	41.23%	**243.17%**
	AR	18.17%	−14.41%	−19.00%	−8.51%	18.63%	27.64%	26.02%	93.72%
	SR	0.64	−0.33	−0.68	−0.32	1.15	1.32	1.24	4.47
	MDD	22.36%	35.59%	29.66%	28.97%	16.40%	8.46%	9.92%	3.30%
HSI	CR	−31.53%	30.95%	−34.21%	−39.17%	−40.45%	65.92%	1.30%	**681.55%**
	AR	−26.68%	23.73%	−30.03%	−38.11%	−26.03%	39.17%	2.58%	152.31%
	SR	−0.65	0.92	−0.72	−1.01	−0.88	1.45	0.12	5.07
	MDD	51.04%	12.15%	42.47%	43.03%	58.69%	13.07%	17.02%	3.59%
IXIC	CR	−20.23%	19.64%	−27.44%	−46.93%	−23.45%	−20.86%	26.55%	**653.90%**
	AR	−13.16%	18.32%	−21.44%	−49.87%	−10.85%	−12.83%	20.45%	142.76%
	SR	−0.33	0.51	−0.55	−1.28	−0.41	−0.33	0.61	5.02
	MDD	35.57%	24.17%	38.99%	50.62%	36.30%	35.72%	19.11%	4.53%
SP500	CR	2.46%	−3.06%	−17.00%	−43.76%	0.004%	18.28%	32.88%	**320.10%**
	AR	4.79%	1.80%	−12.70%	−46.53%	1.84%	13.47%	21.72%	105.05%
	SR	0.16	0.05	−0.47	−1.58	0.10	0.72	0.95	4.75
	MDD	25.41%	27.30%	25.48%	45.71%	25.38%	9.02	19.22%	5.89%
N225	CR	−5.67%	5.10%	−6.52%	−37.00%	−18.51%	55.35%	47.72%	**308.84%**
	AR	−2.12%	6.66%	−3.11%	−37.33%	−9.03%	34.21%	29.35%	107.72%
	SR	−0.07	0.237	−0.11	−1.343	−0.476	1.49	1.43	4.91
	MDD	18.66%	13.78%	18.96%	37.16%	24.68%	9.54	12.56%	3.24%

[1] CR: Cumulative return refers to the total amount of return on an investment over a period of time.
[2] AR: Annualized return refers to the average rate of return earned on an investment over a period of time, expressed as a percentage per year.
[3] SR: The Sharpe ratio used to evaluate the risk-adjusted return of an investment or investment portfolio. It measures the excess return earned over a risk-free rate per unit of volatility or total risk.
[4] MDD: Maximum drawdown indicates the potential downside risk of an investment or portfolio, measuring the largest percentage drop from a peak to a trough in the value of an investment or investment portfolio over a specific period of time.

Table 4
The prediction accuracy of pre-trained network on six datasets.

Indexes	Data Type	Dataset	TimesNet	WFNet	DLinear
DJI	Daily	Training	80.29%	74.80%	51.71%
		Test	41.00%	48.74%	50.21%
	Daily&Weekly	Training	**92.43%**	89.86%	56.53%
		Test	**61.72%**	58.16%	54.18%
FCHI	Daily	Training	**91.43%**	79.01%	42.85%
		Test	39.23%	41.15%	46.70%
	Daily&Weekly	Training	**83.63%**	83.15%	53.86%
		Test	59.06%	**60.13%**	49.68%
HSI	Daily	Training	85.56%	75.94%	42.21%
		Test	42.83%	44.54%	42.40%
	Daily&Weekly	Training	90.88%	**94.01%**	51.94%
		Test	**66.38%**	62.96%	58.89%
IXIC	Daily	Training	93.77%	84.00%	47.34%
		Test	42.89%	42.47%	44.14%
	Daily&Weekly	Training	**95.71%**	95.40%	56.40%
		Test	**62.30%**	62.13%	52.72%
SP500	Daily	Training	91.77%	72.97%	52.86%
		Test	46.44%	46.03%	46.65%
	Daily&Weekly	Training	**87.37%**	76.31%	59.03%
		Test	**62.97%**	59.00%	53.97%
N225	Daily	Training	95.01%	86.27%	40.49%
		Test	43.75%	43.75%	44.18%
	Daily&Weekly	Training	**96.75%**	84.87%	60.15%
		Test	59.48%	56.9%	**60.34%**
Average	Daily	Training	88.56%	77.34%	47.39%
		Test	42.48%	44.59%	46.02%
	Daily&Weekly	Training	**90.00%**	87.75%	55.57%
		Test	**62.49%**	60.47%	53.89%

information. Detailed results of these comparisons are presented in Table 4.

From Table 4, it is evident that DLinear exhibits the poorest performance within the same dataset. Its average accuracy across the test sets of the six indices is only slightly above 50%. Furthermore, DLinear's performance in the training set is notably lower than the other two networks. In contrast, both WFNet and TimesNet demonstrate consistently strong performance across all training sets, with predictive accuracies surpassing 80%. Particularly noteworthy is their superior performance in the daily&weekly dataset, where the average predictive accuracy significantly outperforms that of DLinear.

Upon closer comparison, one can observe that TimesNet performs slightly better than WFNet. Importantly, it can be seen that when daily&weekly data is utilized as input for the models, the average predictive accuracy of all three networks surpasses that achieved with only daily data as input. Take TimesNet as an example, the average prediction accuracy with daily data on test set was 42.48%. But with daily&weekly data, the average prediction accuracy became 62.49%. Following discussions, it can posit that the inclusion of daily&weekly data provides a more detailed depiction of financial market information, enabling the networks to better extract features of the state. Therefore, in subsequent experiments, TimesNet will be employed as the policy network for HA3C, while utilizing daily&weekly data as inputs for the model.

As HA3C is an extension of the A2C algorithm with the incorporation of a pre-trained behavioral cloning network, validating the effectiveness of this framework requires a comparison between the original A2C algorithm and the HA3C algorithm. Both A2C and HA3C use TimesNet as the architectural backbone for both the actor network and the critic network. For the actor network, TimesNet processes the current state to generate an action, whereas, for the critic network, it assists in estimating the value function of the current state and the corresponding action. HA3C extends the A2C architecture by incorporating pre-trained TimesNet expert networks into A2C-based actor networks. In contrast, A2C randomly initializes the parameters of TimesNet for training the actor network. This method utilizes the parameters of the expert network as initial values for the training phase of HA3C. Additionally, observing the curves of accumulated rewards during their respective training processes is essential to determine whether HA3C

Table 5
The performance of HA3C and A2C with Daily&Weekly data on six datasets.

Method	Datasets	CR	AR	SR	MDD
A2C	DJI	200.63%	82.02%	4.23	4.45%
	FCHI	234.10%	91.93%	4.25	3.41%
	HSI	557.98%	140.91%	4.57	7.62%
	IXIC	570.50%	136.91%	4.71	4.55%
	SP500	250.49%	92.89%	4.17	9.94%
	N225	297.77%	105.76%	4.84	4.57%
HA3C	DJI	217.64%	85.75%	4.50	4.34%
	FCHI	243.17%	93.72%	4.47	3.30%
	HSI	681.55%	152.31%	5.07	3.59%
	IXIC	635.90%	142.76%	5.02	4.53%
	SP500	320.10%	105.05%	4.75	5.89%
	N225	308.84%	107.72%	4.91	3.24%

Table 6
The performance of HA3C with different reward function on HSI.

Data Type	Metrics	Sharpe Ratio	Return	Max_Min
Daily	CR	2.35%	4.75%	**6.97%**
	AR	5.89%	8.05%	**9.21%**
	SR	0.17	0.22	**0.27**
	MDD	32.76%	**30.71%**	31.11%
Daily&Weekly	CR	659.16%	625.91%	**681.55%**
	AR	150.37%	147.04%	**152.31%**
	SR	5.01	**5.19**	5.07
	MDD	8.77%	3.60%	**3.59%**

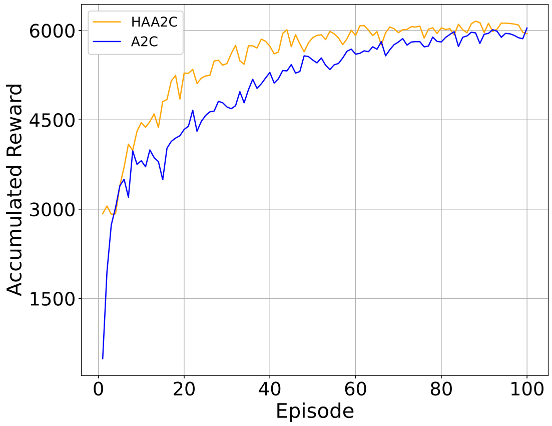


Fig. 7. Accumulated Reward curve of HA3C and A2C on DJI.

influences the training of A2C. Table 5 presents a performance comparison of the two algorithms under daily&weekly data across six datasets, while Figs. 7 to 12 contrast their accumulated reward curves.

From Table 5, it is evident that HA3C consistently outperforms A2C in terms of cumulative return across all six datasets. Moreover, upon comparing other metrics, one can observe that HA3C exhibits lower MDD and higher SR and AR compared to A2C. This suggests that the introduction of the pre-trained behavioral cloning network leads to a comprehensive improvement in the performance of A2C. On the other hand, by observing Figs. 7 to 12, it can be seen that HA3C converges significantly faster than A2C. In general, A2C tends to gradually converge after around 50 epochs, whereas HA3C achieves convergence by the 20th epoch. Also, HA3C can have a high accumulated reward on the first epoch, which helps it converge faster than A2C. Therefore, HA3C, both in training and testing, outperforms A2C, indicating the efficacy of the proposed framework.

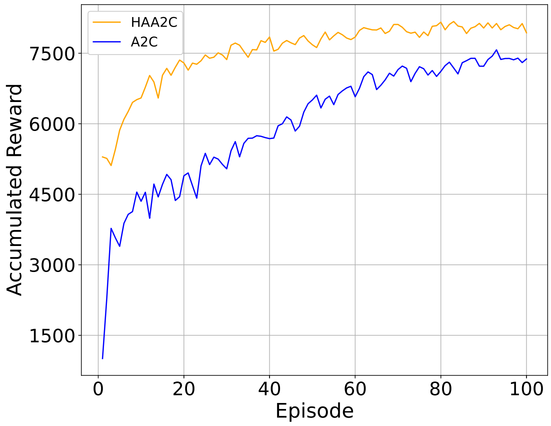


Fig. 8. Accumulated Reward curve of HA3C and A2C on FCHI.

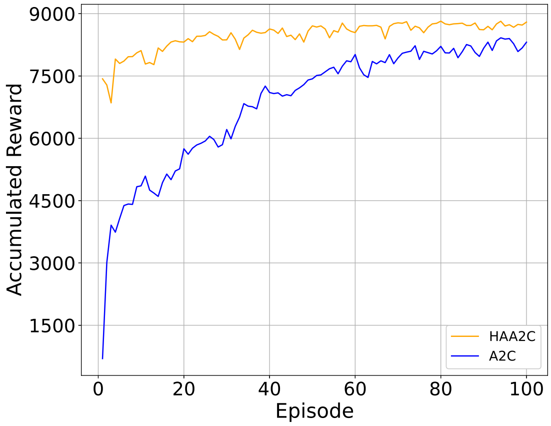


Fig. 9. Accumulated Reward curve of HA3C and A2C on HSI.

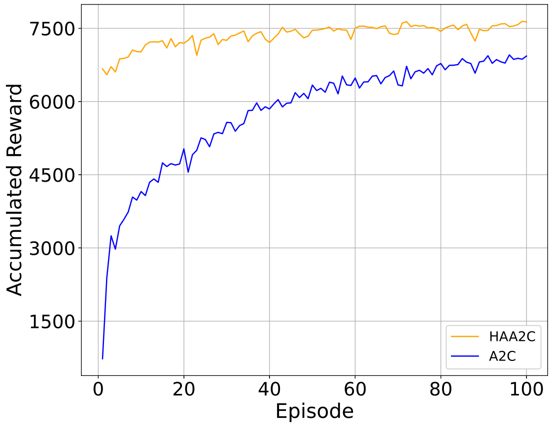


Fig. 10. Accumulated Reward curve of HA3C and A2C on IXIC.

4.5.3. *Analysis of reward functions*

The details of the reward function for HA3C have already been discussed in Section 3.2.3. However, in many reinforcement learning works related to finance, commonly employed reward functions include the Sharpe Ratio and Return (Corazza et al., 2019; Corazza & Sangalli, 2015; Cornalba et al., 2022; Huang, 2018; Li et al., 2022; Si et al., 2017; Tran et al., 2023). The Sharpe Ratio serves as a metric for evaluating the risk-adjusted return of an investment or investment portfolio, while Return focuses more on mid-term returns. The proposed reward function with the use of the aforementioned two reward functions in HA3C are compared within the HSI index. The specific experimental results are presented in Table 6.

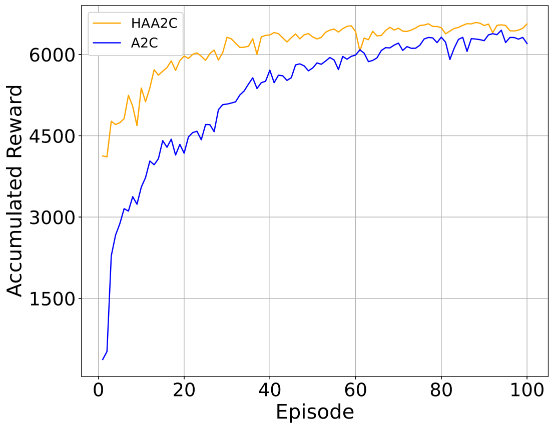


Fig. 11. Accumulated Reward curve of HA3C and A2C on SP500.

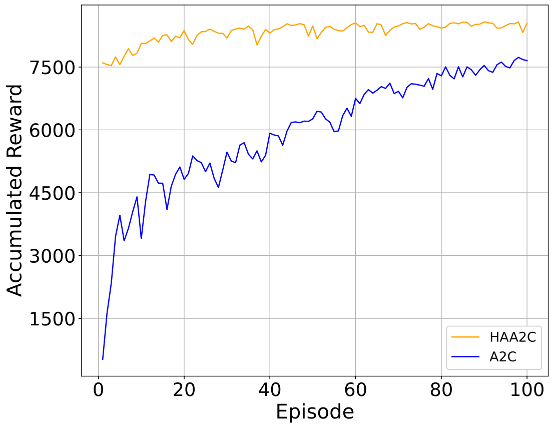


Fig. 12. Accumulated Reward curve of HA3C and A2C on N225.

In Table 6, the performance of three reward functions are compared under both daily and daily&weekly data. It is evident that the min_max reward function consistently achieves the highest cumulative return under both data settings. Notably, under daily&weekly data, the min_max reward function enables HA3C to achieve a cumulative return that is 20% higher than Sharpe Ratio and nearly 60% higher than Return.

Although under daily data, HA3C agent with min_max reward function have a Maximum Drawdown (MDD) higher and a Sharpe Ratio lower under daily&weekly data than H2AAC agent with Return reward function, these differences are marginal. Therefore, within the framework of HA3C, the proposed min_max reward function facilitates the agent in making better trading decisions, resulting in higher cumulative returns.

4.5.4. *Analysis of RL methods*

HA3C's core algorithm is A2C, wherein we incorporate a pre-trained behavioral cloning network as the actor network for A2C. This integration accelerates the convergence of A2C and enhances its overall performance. However, it is essential to note that the incorporation of a behavioral cloning network is not exclusive to A2C within reinforcement learning algorithms. Many algorithms rooted in the Actor–Critic paradigm, such as PPO, can similarly utilize a behavioral cloning network as their actor network. Additionally, in the realm of value-based learning, algorithms like DQN can employ a similar approach by using a behavioral cloning network as the initial Q network. Consequently, a comparative analysis of the performance of PPO, DQN, and A2C in HSI index within the proposed framework is conducted. Detailed comparison results are presented in Table 7.

Table 7
The comparison of HA3C, PPO and DQN on HSI.

Data Type	Metrics	PPO	DQN	HA3C
Daily	CR	−31.53%	−7.88%	**6.97%**
	AR	−26.68%	−2.76%	**9.21%**
	SR	−0.65	−0.08	**0.27**
	MDD	51.04%	31.74%	**31.11%**
Daily&Weekly	CR	562.83%	539.51%	**681.55%**
	AR	141.46%	138.98%	**152.31%**
	SR	4.55	4.52	**5.07**
	MDD	4.39%	8.37%	**3.59%**

From Table 7, it could be seen that HA3C consistently obtains the highest cumulative returns, both for daily data and for daily and weekly data. Moreover, after comparing various metrics, HA3C consistently outperformed PPO and DQN. Therefore, within the proposed framework, it could be concluded that the A2C algorithm enables agents to make superior trading decisions.

Furthermore, from the 4.5.2 to 4.5.4 part of the experiment, it was observed that the use of daily&weekly data as inputs to the behavioral cloning network and the RL algorithm produced better results compared to the use of daily data only. On the one hand, this improvement was attributed to the richer information content of daily&weekly data, which provided a more detailed description of financial market dynamics compared to using only daily data. On the other hand, it was considered that the daily/weekly data provided the model with more extractable features, thus improving the overall performance of the model. Therefore, it could be argued that maximizing the inclusion of data features could improve performance during the testing and training phases, as long as it is feasible within the constraints of the model.

4.5.5. *Analysis of trading strategy*

After analyzing the effectiveness of HA3C in different experiments, the observation of the actions taken by the agent in all datasets was continued. Figs. 13 to 18 showed the actions performed by the HA3C agent in the six datasets. These actions were labeled on the stock price trend charts to make it easier to understand how the agent made trading decisions based on price movements.

Firstly, trading actions performed by the HA3C agent in DJI and HSI datasets are depicted in Fig. 13 and Fig. 15. Detailed examination of the zoomed-in segments in these two graphs reveals that the HA3C agent exhibits sharp trading decisions. For instance, in Fig. 13, around the 420th day, the stock price reached a low point. At this juncture, the agent foresaw an imminent upward trend and promptly executed a "long" trading decision. Subsequently, by the 430th day, when the stock price approached a peak, the agent opted for a "short" trading decision. A comparable scenario is evident in Fig. 15, wherein the agent initiated a "long" trading decision at the lowest point around the 275th day. Comparable occurrences were observed in both Fig. 16 and Fig. 17, wherein the agent effectively navigated substantial upward or downward in the subsequent market prices, demonstrating a keen ability to make accurate trading decisions. These consistently judicious decisions resulted in profitable outcomes.

Secondly, Figs. 14 and 18 depict the trading behavior of HA3C on the FCHI and N225. From the zoomed sections of the two figures, it is evident that the fluctuation in the index price curve is quite pronounced. At such junctures, the agent adopts a high-frequency trading strategy. For instance, as illustrated in Fig. 14, around the 337th day, the agent strategically executed a "long" trading decision upon anticipating an imminent uptick in stock prices. Just a few days later, it promptly executed a "short" decision when prices were poised to decline. Subsequently, after the conclusion of the price decline, the agent swiftly made a "long" decision again. This implies that the HA3C agent tends to employ high-frequency trading during periods

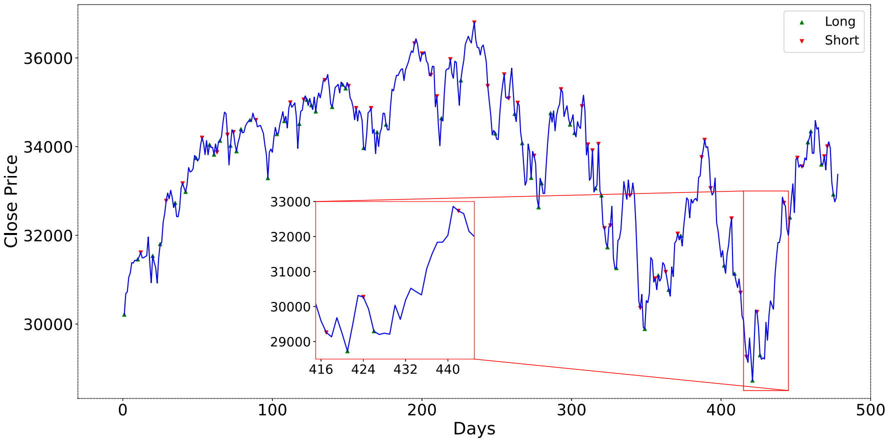


Fig. 13. Actions of HA3C Agent in DJI.

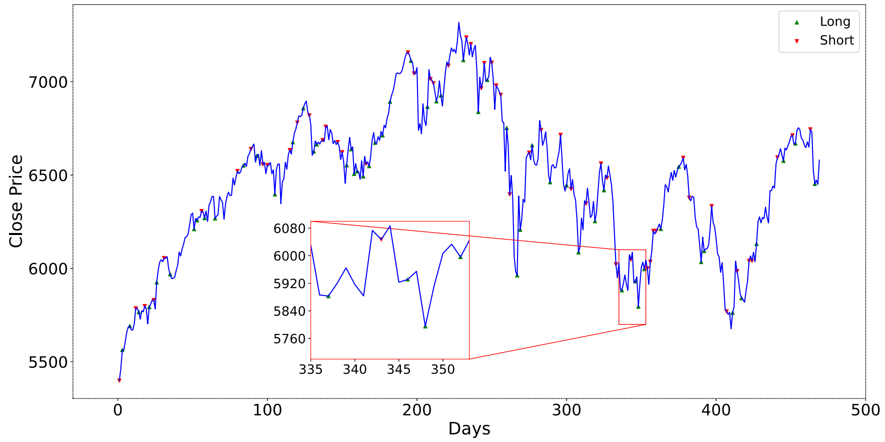


Fig. 14. Actions of HA3C Agent in FCHI.

of frequent stock price oscillations to maximize returns. A more illustrative example is depicted in Fig. 18, where the enlarged section illustrates a segment of high-frequency oscillations in stock prices. The agent adeptly pinpointed the optimal timing for each trade. In specific instances, the agent executed a "long" decision precisely on the day when the stock price decline concluded, and correspondingly, the "short" decision was impeccably timed to align with the termination of an upward price movement. With this high-frequency and precise trading, agents could earn higher returns.

4.5.6. Discussion

In this comprehensive analysis of HA3C through various experiments and perspectives, the proposed methodology consistently outperforms baseline strategies across all datasets in terms of cumulative return, annualized return, and Sharpe ratio. These results align with the theoretical foundation established in previous experiments.

1. The efficacy of HA3C has been validated across stock market indices from different regions worldwide. To further enhance its performance, the incorporation of the concept of large models can be explored. This involves pre-training a generalized strategy on multiple datasets and subsequently fine-tuning it on specific assets, aiming to create a more versatile model with improved results.

2. While the reward function of HA3C proves advantageous over the Sharpe ratio and yield, it is essential to acknowledge that the design of such reward functions relies on human experience. Consequently, it represents a complex task demanding substantial effort from researchers.

3. HA3C leverages IL for the pre-training of the strategy network, a validated approach. However, as IL resembles supervised learning, it lacks the conditions of the MDP assumption inherent in RL.

4. Notably, the inclusion of daily and weekly data significantly enhances the model's performance, offering a more comprehensive view of the financial environment and additional features for extraction. Consideration of other modal data, such as news and financial reports, could further enrich the features and enhance the model's capabilities.

5. Conclusion

In this paper, the Human Alignment Advantage Actor–Critic (HA3C) algorithm has been presented, motivated by the principles of IL and the dominance actor–critic approach. The algorithm aims to enhance single-asset trading strategies by integrating these two methodologies. The proposed method involves defining expert action labels, training a strategy prediction network through supervised learning on behavioral cloning, and subsequently transferring this network into DRL as an initialization parameter for the strategy network. This integration of imitation learning techniques capitalizes on financial domain knowledge to consolidate trading intelligence. The main contribution are as follows:

1. A novel financial trading framework is proposed, building upon the A2C algorithm and imitation learning. The notable enhancement in the convergence speed of the A2C algorithm is achieved by seamlessly integrating a pre-trained behavioral cloning network into the policy network of A2C. Importantly, in the context of single-asset trading, this approach not only optimizes the learning pace but also obtains

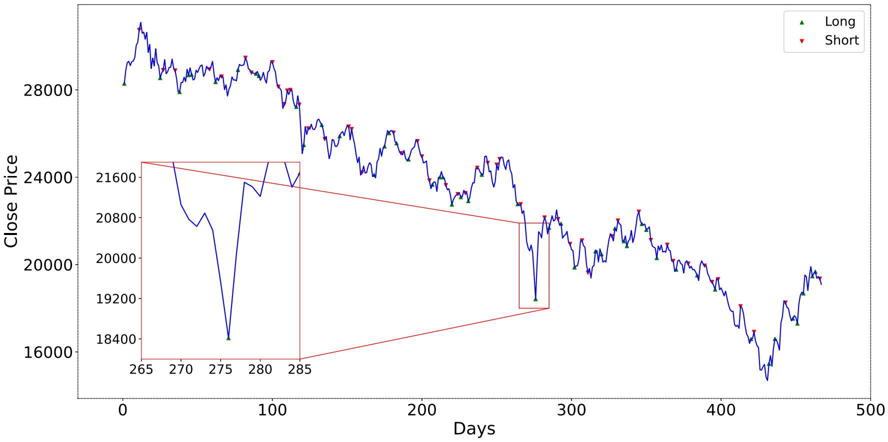


Fig. 15. Actions of HA3C Agent in HSI.

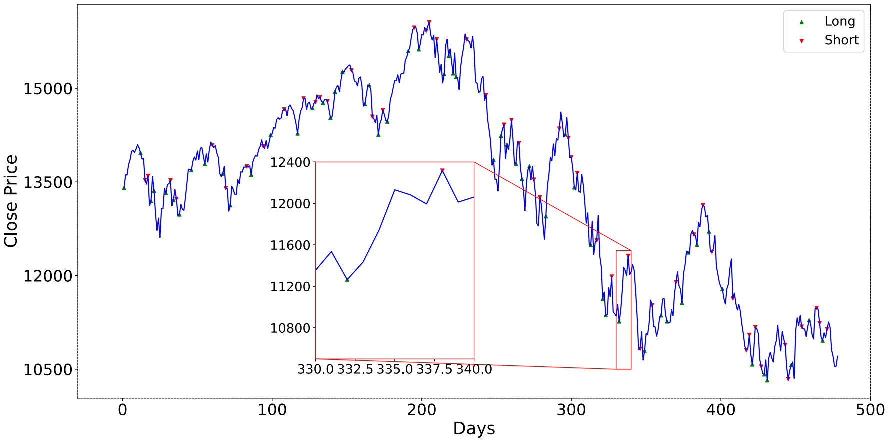


Fig. 16. Actions of HA3C Agent in IXIC.

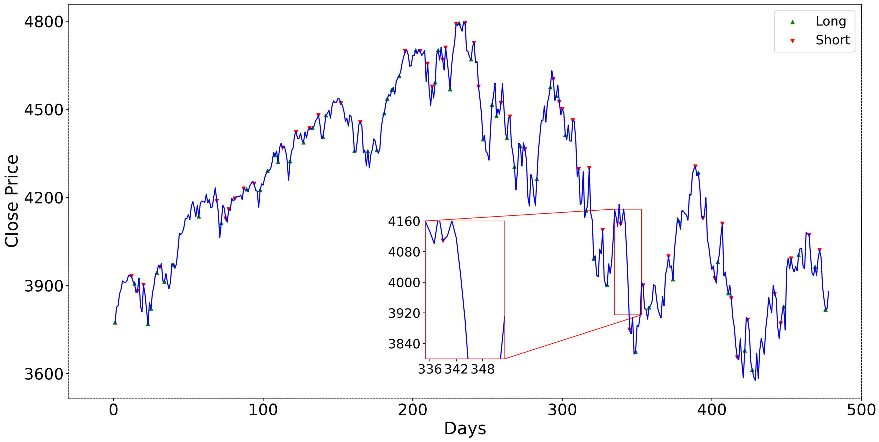


Fig. 17. Actions of HA3C Agent in SP500.

significantly higher cumulative returns. The introduction of this framework exhibits promising innovation and benefits in the field of financial trading.

2. A novel reward function, tailored for application in financial environments, is proposed. In contrast to commonly used reward functions in other relevant works, such as the Sharpe Ratio and Return, this specific reward function aims to prompt the agent under the HA3C framework to make a more extensive and superior range of trading decisions. As a result, it facilitates the attainment of higher cumulative returns.

3. The impact of daily data versus daily&weekly data on the performance of reinforcement learning in financial environments is also demonstrated in the paper. Experiments indicate that the inclusion of daily and weekly data, owing to its ability to offer a more comprehensive description of the financial environment and provide additional features for the model to extract, leads to a significant improvement in

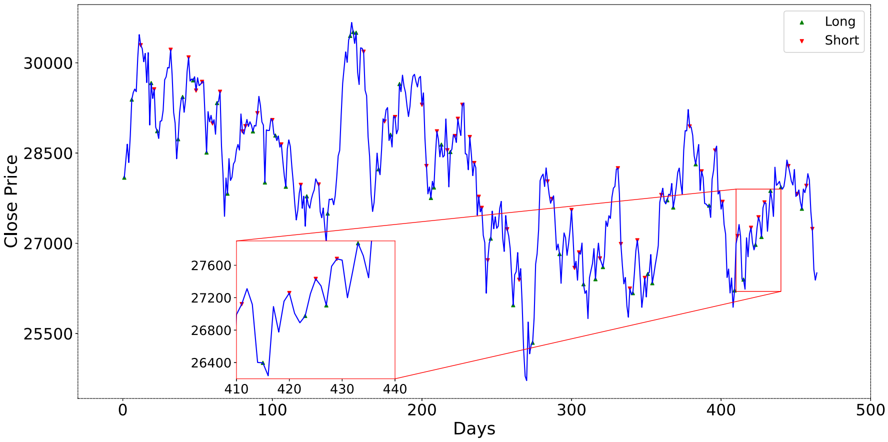


Fig. 18. Actions of HA3C Agent in N225.

model performance, at least under the experimental conditions outlined in this paper.

Indeed, there exist some limitations within the proposed framework. Firstly, the experimental results are tested on only a few indices and do not explore the possibility of making the model more generalizable. Second, while we compare the proposed reward function with two representative reward functions, there are still many other reward functions in the field of financial reinforcement learning. Finally, while experiments have demonstrated that the inclusion of daily and weekly data improves the performance of the model, it has not been investigated whether the inclusion of more features (e.g., news, financial reports, etc.) would further improve the effectiveness of the model.

Looking ahead, upcoming research efforts will involve more complex scenarios. First, exploring the construction of a composite dataset containing features from multiple datasets may prove beneficial. Training models on such a dataset may endow them with capabilities similar to those of a generalized base model. Secondly, the incorporation of inverse reinforcement learning into the current framework could also be investigated. This would involve collecting trading records and expert feedback on specific index or stock trades to facilitate the construction of neural networks that learn the underlying reward signals from the feedback. This approach promises to significantly reduce the time and effort required for reward function design. Additionally, exploring the use of multi-modality data as environmental states could be investigated. An exploration aimed at determining if the performance of the model can be further improved by leveraging additional relevant information.

CRediT authorship contribution statement

Yuling Huang: Methodology, Conceptualization, Investigation, Writing – original draft. **Chujin Zhou:** Methodology, Software, Validation, Visualization, Writing – original draft. **Kai Cui:** Methodology, Software, Visualization. **Xiaoping Lu:** Supervision, Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported in part by the Faculty Research, Macao Special Administrative Region of China Grants, Macau University of Science and Technology (Project no. FRG-22-001-INT).

References

Bain, M., & Sammut, C. (1995). A framework for behavioural cloning. In *Machine intelligence 15* (pp. 103–129).

Bellemare, M. G., Candido, S., Castro, P. S., Gong, J., Machado, M. C., Moitra, S., et al. (2020). Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, *588*, 77–82.

Chakole, J., & Kurhekar, M. (2020). Trend following deep Q-learning strategy for stock trading. *Expert Systems*, *37*(4), Article e12514.

Chakraborty, S. (2019). Capturing financial markets to apply deep reinforcement learning. *Computational Finance*, arXiv.

Chen, L., & Gao, Q. (2019). Application of deep reinforcement learning on automated stock trading. In *2019 IEEE 10th International Conference on Software Engineering and Service Science* (pp. 29–33).

Chou, J. S., & Nguyen, T. K. (2018). Forward forecast of stock price using sliding-window metaheuristic-optimized machine-learning regression. *IEEE Transactions on Industrial Informatics*, *14*(7), 3132–3142.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in neural information processing systems*: *vol. 30*.

Corazza, M., Fasano, G., Gusso, R., & Pesenti, R. (2019). A comparison among reinforcement learning algorithms in financial trading systems. University Ca'Foscari of Venice, Dept. of Economics Research Paper Series No 33.

Corazza, M., & Sangalli, A. (2015). Q-learning and SARSA: A comparison between two intelligent stochastic control approaches for financial trading. *SSRN Electronic Journal*.

Cornalba, F., Disselkamp, C., Scassola, D., & Helf, C. (2022). Multi-objective reward generalization: Improving performance of deep reinforcement learning for selected applications in stock and cryptocurrency trading. ArXiv, arXiv:2203.04579.

Cui, K., Hao, R., Huang, Y., Li, J., & Song, Y. (2023). A novel convolutional neural networks for stock trading based on ddqn algorithm. *IEEE Access*, *11*, 32308–32318.

Dang, Q. V. (2020). Reinforcement learning in stock trading. In *Advanced Computational Methods for Knowledge Engineering: Proceedings of the 6th International Conference on Computer Science, Applied Mathematics and Applications, ICCSAMA 2019 6* (pp. 311–322).

Diederichs, E. (2019). Reinforcement learning—A technical introduction. *Journal of Autonomous Intelligence*, *2*(2), 25.

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in finance*: *vol. 1170*, Springer.

Duan, Y., Andrychowicz, M., Stadie, B., Jonathan Ho, O., Schneider, J., Sutskever, I., et al. (2017). One-shot imitation learning. In *Advances in neural information processing systems*: *vol. 30*.

Felizardo, L. K., Paiva, F. C. L., de Vita Graves, C., Matsumoto, E. Y., Costa, A. H. R., Del-Moral-Hernandez, E., et al. (2022). Outperforming algorithmic trading reinforcement learning systems: A supervised approach to the cryptocurrency market. *Expert Systems with Applications*, *202*, Article 117259.

Finn, C., Yu, T., Zhang, T., Abbeel, P., & Levine, S. (2017). One-shot visual imitation learning via meta-learning. In *Conference on robot learning* (pp. 357–368). PMLR.

Gao, X. (2018). Deep reinforcement learning for time series: playing idealized trading games. arXiv, arXiv:1803.03916.

Ge, J., Qin, Y., Li, Y., Huang, y., & Hu, H. (2022). Single stock trading with deep reinforcement learning: A comparative study. In *2022 14th International Conference on Machine Learning and Computing* (pp. 34–43).

Goluža, S., Bauman, T., Kovačević, T., & Kostanjčar, Z. (2023). Imitation learning for financial applications. In *2023 46th MIPRO ICT and electronics convention* (pp. 1130–1135).

Hester, T., Vecerík, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., et al. (2017). Learning from demonstrations for real world reinforcement learning. ArXiv, arXiv:1704.03732.

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in neural information processing systems*: *vol. 29*.

Hu, G., Hu, Y., Yang, K., Yu, Z., Sung, F., Zhang, Z., et al. (2018). Deep stock representation learning: From candlestick charts to investment decisions. In *2018 IEEE international conference on acoustics, speech and signal processing* (pp. 2706–2710). IEEE.

Huang, C. Y. (2018). Financial trading as a game: A deep reinforcement learning approach. arXiv preprint arXiv:1807.02787.

Huang, Y., Cui, K., Song, Y., & Chen, Z. (2023). A multi-scaling reinforcement learning trading system based on multi-scaling convolutional neural networks. *Mathematics*, *11*(11), 2467.

Huang, Z., Li, N., Mei, W., & Gong, W. (2023). Algorithmic trading using combinational rule vector and deep reinforcement learning. *Applied Soft Computing*, *147*, Article 110802.

Huang, Y., Lu, X., Zhou, C., & Song, Y. (2023). DADE-DQN: Dual action and dual environment deep Q-network for enhancing stock trading strategy. *Mathematics*, *11*(17), 3626.

Huang, Y., & Song, Y. (2023). A new hybrid method of recurrent reinforcement learning and BiLSTM for algorithmic trading. *Journal of Intelligent & Fuzzy Systems*, *45*, 1939–1951.

Huang, Y., Wan, X., Zhang, L., & Lu, X. (2023). A novel deep reinforcement learning framework with bilstm-attention networks for algorithmic trading. *Expert Systems with Applications*, Article 122581.

Huang, Y., Zhou, C., Cui, K., & Lu, X. (2024). A multi-agent reinforcement learning framework for optimizing financial trading strategies based on TimesNet. *Expert Systems with Applications*, *237*, Article 121502.

Hussein, A., Gaber, M. M., Elyan, E., & Jayne, C. (2017). Imitation learning: A survey of learning methods. *ACM Computing Surveys*, *50*(2), 1–35.

Jeong, G. H., & Kim, H. Y. (2019). Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning. *Expert Systems with Applications*, *117*, 125–138.

Lei, K., Zhang, B., Li, Y., Yang, M., & Shen, Y. (2019). Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. *Expert Systems with Applications*, *140*, Article 112872.

Li, Y., Liu, P., & Wang, Z. (2022). Stock trading strategies based on deep reinforcement learning. *Scientific Programming*, *2022*.

Lima Paiva, F. C., Felizardo, L. K., Bianchi, R. A. d. C., & Costa, A. H. R. (2021). Intelligent trading systems: a sentiment-aware reinforcement learning approach. In *Proceedings of the Second ACM International Conference on AI in Finance* (pp. 1–9).

Liu, Y., Liu, Q., Zhao, H., Pan, Z., & Liu, C. (2020). Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 34*, (02), (pp. 2128–2135).

Liu, P., Wu, B., Li, N., Dai, T., Lei, F., Bao, J., et al. (2023). WFTNet: Exploiting global and local periodicity in long-term time series forecasting. arXiv preprint arXiv:2309.11319.

Liu, P., Zhang, Y., Bao, F., Yao, X., & Zhang, C. (2023). Multi-type data fusion framework based on deep reinforcement learning for algorithmic trading. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, *53*(2), 1683–1706.

Ma, C., Zhang, J., Liu, J., Ji, L., & Gao, F. (2021). A parallel multi-module deep reinforcement learning algorithm for stock trading. *Neurocomputing*.

MacGlashan, J., Ho, M. K., Loftin, R., Peng, B., Wang, G., Roberts, D. L., et al. (2017). Interactive learning from policy-dependent human feedback. In *International conference on machine learning* (pp. 2285–2294).

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., et al. (2016a). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928–1937). PMLR.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., et al. (2016b). Asynchronous methods for deep reinforcement learning. ArXiv, arXiv:1602.01783.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Park, D. Y., & Lee, K. H. (2021). Practical algorithmic trading using state representation learning and imitative reinforcement learning. *IEEE Access*, *9*, 152310–152321.

Pavel, M. I., Muhtasim, D. A., & Faruk, O. (2021). Decision making process of stock trading implementing DRQN and ARIMA. In *2021 IEEE madras section conference* (pp. 1–6). IEEE.

Peng, Y. L., & Lee, W. P. (2023). Valuation of stocks by integrating discounted cash flow with imitation learning and guided policy. *IEEE Transactions on Automation Science and Engineering*.

Ross, S., & Bagnell, D. (2010). Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 661–668). JMLR Workshop and Conference Proceedings.

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, *3*(6), 233–242.

Si, W., Li, J., Ding, P., & Rao, R. (2017). A multi-objective deep reinforcement learning approach for stock index future's intraday trading. In *2017 10th International symposium on computational intelligence and design (ISCID)*: *vol. 2*, (pp. 431–436).

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, *362*, 1140–1144.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, *550*(7676), 354–359.

Stadie, B. C., Abbeel, P., & Sutskever, I. (2017). Third-person imitation learning. arXiv, arXiv:1703.01703.

Taghian, M., Asadi, A., & Safabakhsh, R. (2022). Learning financial asset-specific trading rules via deep reinforcement learning. *Expert Systems with Applications*, Article 116523.

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, *173*, Article 114632.

Tran, M., Pham-Hi, D., & Bui, M. (2023). Optimizing automated trading systems with deep reinforcement learning. *Algorithms*, *16*, 23.

Tsai, M. C., Cheng, C. H., Tsai, M. I., & Shiu, H. Y. (2018). Forecasting leading industry stock prices based on a hybrid time-series forecast model. *PLoS One*, *13*(12), Article e0209922.

Vishal, M., Satija, Y., & Babu, B. S. (2021). Trading agent for the Indian stock market scenario using actor-critic based reinforcement learning. In *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions* (pp. 1–5).

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., & Long, M. (2022). Timesnet: Temporal 2d-variation modeling for general time series analysis. arXiv preprint arXiv:2210.02186.

Xiao, X. (2023). Quantitative investment decision model based on PPO algorithm. *Highlights in Science, Engineering and Technology*, *34*, 16–24.

Ye, Z. J., & Schuller, B. W. (2023). Human-aligned trading by imitative multi-loss reinforcement learning. *Expert Systems with Applications*, *234*, Article 120939.

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*: *vol. 37*, (9), (pp. 11121–11128).