IFA
IRAN FINANCE ASSOCIATION

Iranian Journal of Finance

Print ISSN 2676-6337
Online ISSN 2676-6345

# A Profitable Portfolio Allocation Strategy Based on Money Net-Flow Adjusted Deep Reinforcement Learning

**Samira Khonsha**

Ph.D. Candidate in Computer Engineering, Department of Computer Engineering, Yazd University, Yazd, Iran. (Email: khonsha.samira@gmail.com)

**Mehdi Agha Sarram***

*Corresponding Author, Associate Prof., Department of Computer Engineering, Yazd University, Yazd, Iran. (Email: mehdi.sarram@yazd.ac.ir)

**Razieh Sheikhpour**

Assistant Prof.,, Department of Computer Engineering, Faculty of Engineering, Ardakan University, P.O. Box 184, Ardakan, Iran. (Email: rsheikhpour@ardakan.ac.ir)

Iranian Journal of Finance, 2023, Vol. 7, No.4, pp. 59-89.
Publisher: Iran Finance Association
doi: https://doi.org/ 10.30699/IJF.2023.364455.1369
Article Type: Original Article
© Copyright: Author(s)
Type of License: Creative Commons License (CC-BY 4.0)

Received: November 11, 2023
Received in revised form: January 08, 2023
Accepted: March 01, 2023
Published online: December 12, 2023

## Abstract

Portfolio allocation with Deep Reinforcement Learning (DRL) has been the focus of many researchers. In investing, a portfolio optimization strategy is selecting assets that maximize return on investment while minimizing the risk. Asset optimization involves balancing risk and return, where stock returns are

profits over time, and risk is the standard deviation value of the asset's return. Many of the existing methods for portfolio optimization are essentially the expansion of diversification methods for assets in the investment. Signiant drawdowns and early entry into the share are still challenging in portfolio construction. The idea is that having a portfolio based on net money flow is less risky than allocating a portfolio based on historical data only and turbulence as risk aversion. This paper proposes a profitable stock recommendation framework for portfolio construction using the DRL model based on the net money flow (MNF) indicator. We develop a new risk indicator based on the intelligent net-flow behavior of smart money to help determine the optimal market timing for buying and selling. The experimental results of real-world trading scenario validation show that the model outperforms all the considered baselines and even the conventional Buy-and-Hold strategy. Moreover, in this paper, the effect of defining different environments made of various information with hyper parameter optimization on the performance of models has been investigated, and the performance of DRL-driven models in different markets and asset positions has been investigated. The empirical results show the dominance of DRL models based on MNF indicators.

**Keywords:** Portfolio Optimization Strategy, Automate Trading, Deep Reinforcement Learning, Money Net Flow Indicator

## Introduction

In the last decade, there has been an exponential increase in the application of artificial intelligence, including machine learning and deep learning algorithms for automated trading in stock markets. Deep reinforcement learning (DRL) has recently attracted considerable attention due to its remarkable and specific achievements in video games (Mnih et al., 2015) and board games (Silver et al., 2016). After that, DRL has received more attention in algorithmic trading because it has both the perception ability of deep learning and the decision-making ability of reinforcement learning. Automatic algorithmic trading enables investors to execute complex trading strategies without human intervention.

DRL is a goal-oriented learning system that optimizes a financial measure without explicitly predicting future price movements. DRL performs the two main stages of business, market analysis and decision-making, simultaneously (Francis, 2022). Optimizing a measure of financial performance is the investor's ultimate goal, usually by considering the potential return against any

risk associated with that investment. Under the Modern Portfolio Theory (MPT) proposed by Markowitz, (2002), the main objective is to maximize the expected return on investment while minimizing its variance, which he refers to as the potential risks of capital. Risk aversion indicates whether the investor prefers to protect capital or not. It also affects one's trading strategy when faced with different levels of market volatility. To control the risk in the worst-case scenario and unexpected fluctuations, such as the Corona pandemic, we should use a volatility index to measure the extreme volatility of asset prices. Since then, the concept of relating the variance of returns to investment risks has become popular among researchers and practitioners. Financial turbulence indicator (Salisu, 2022) measures unusual asset price patterns, including extreme fluctuations or decoupling of correlated assets, that are uncharacteristic compared to past observed patterns. The criterion introduced in this paper for detecting investment risk is different, and it considers the flow of smart money in the market as the risk of entering or exiting the market. In the experiments conducted in this paper, the effect of this net money flow (MNF) index on investment returns and risk is shown with a measure such as the Sharpe ratio.

Based on our latest knowledge, this paper is the first to study a deep reinforcement learning model based on actual and legal traders' behavior in the capital market. Accordingly, we introduce a DRL model based on a new market timing indicator that monitors the net flow of real traders' money in the Tehran Stock Exchange (TSE) market and forms a portfolio based on it. The algorithmic trading considered in this paper includes designing a trading system that can buy or sell a variable volume of stocks proposed by the designed model in the TSE. This algorithm aims to maximize the return on the stock's portfolio. We use the historical data of 30 large shares of the TSE, which in addition to the price data including the opening price (O), the highest price (H), the lowest price (L), the closing price (C) and the volume (V), in short OHLCV, of daily transactions, the data related to the transactions of factual and legal stock traders are also used. We consider transaction fees. In addition, we introduce and evaluate the efficiency of the reinforcement agent in different environments that are made of features such as Google Trends, technical indicators, expert-defined features, and others. In the Experiments section, we empirically show that portfolio returns depend on the variation of the environment defined for the agent. Results confirm that an environment with MNF as a risk aversion indicator has better returns than other environments.

## Deep Reinforcement Learning

Imagine a robot (agent) who wants to be a trader. This robot has not received any explicit training for trading. However, it can observe the returns on shares sold (rewards). Maximizing these rewards is the goal of the robot. To achieve this goal, the robot must have access to as much information as possible about the stock, such as price, fundamentals, trading volume, number of shares bought and sold on the stock by factual and legal traders (environment), and information that a professional trader pays attention to, this information is as the robot's sensors. The robot tries different trades (actions) to increase its profit, after observing the result of each transaction, improves its trading strategy (policy). Due to the noise of raw historical price data that may not precisely reflect the future of markets, some agents with different data and strategies are needed to simulate the market better. Figure 1 shows this explained reinforcement learning for trading.

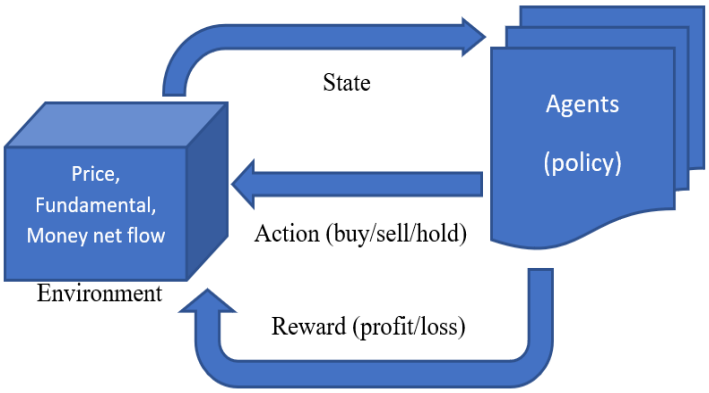


**Figure 1. Reinforcement learning trading**

In the following, the functioning of reinforcement learning and its components and types will be explained in more detail.

## Markov Decision Process

A Markov decision process (MDP) is a discrete-time stochastic control process in mathematics. It provides a mathematical framework for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision-maker. MDP is mainly used to describe the environment in Reinforcement Learning (RL), and RL models are a type of state-based model that utilizes the MDP. Briefly and schematically, RL trains an agent

based on reward and punishment. The RL agent acts as an environment by observing the current state and receiving a reward corresponding to that action, and performing this action puts the environment in the next state. Figure 2 schematically shows the MDP process in RL.

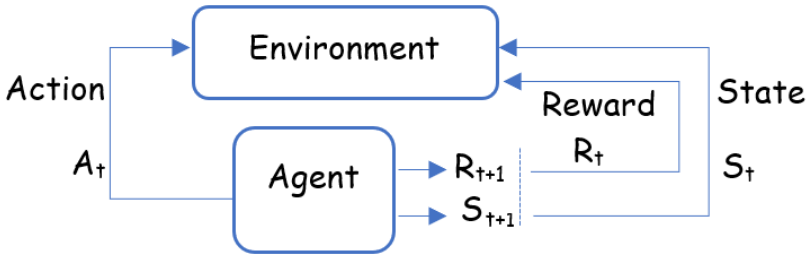


**Figure 2. MDP process in RL (Sutton et al., 2018)**

In trading RL algorithm:

- At time *t*, the agent (reinforcement algorithm) observes the current state ($s_t$) of the environment, which can include various information such as cash balance, stock price in the portfolio, the number of each share, the time elapsed since we bought the share, Technical indicators, the fundamental parameters of the share, the share board including the number of factual and legal buyers and sellers, the volume purchased by factual and legal traders and other features that we can define and consider as the current state of the environment.
- Among the allowed actions (buy/sell/hold), the agent chooses the best action ($a_t$).
- The environment changes to the new state ($s_{t+1}$).
- The environment produces a reward ($r_t$). The reward can be the change in the portfolio's value (increase/decrease/no change).

The policy function ($\pi$ (st) = at) decides the action based on the state. The policy is state-action mapping. In reinforcement learning, we always have an environment with a set of states, actions, a policy function (used to transition between states), and numerical values as rewards. The reinforcement learning agent observes the current state in an iterative cycle, chooses the best action based on the output of the policy function, and receives a reward for this action. This iteration continues until reaching an end state. The goal of the reinforcement learning agent is to achieve the maximum reward. An optimal

policy maximizes the reward. Reinforcement learning algorithms are very diverse and based on what components they use to produce the workflow of Figure 2. To reach the optimal policy, they are classified in Figure 3.

In recent years, multi-agent models (Lin et al. 2022), ensemble models (Faturohman, 2022), and models that use auto encoder (Yue, 2022) for portfolio optimization have also been presented.

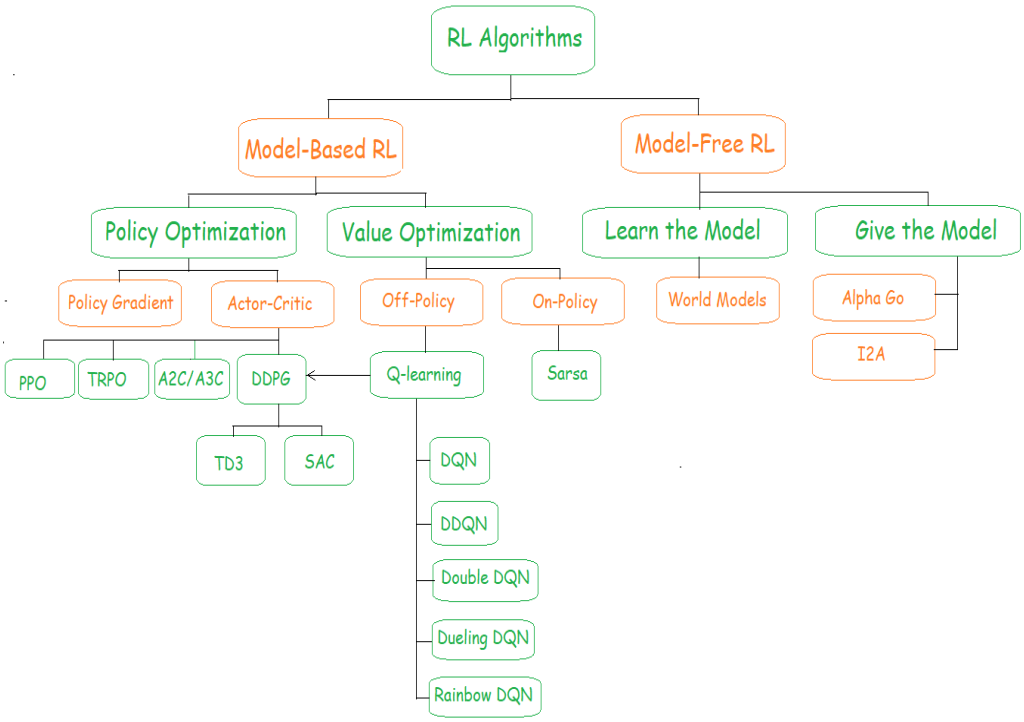


**Figure 3. RL Algorithms**

The working flow of all types of reinforcement learning algorithms includes the following steps:

1- Initialize the policy ($\pi$) with random parameters.

2- With the existing policy, we choose the action ($a$) that has the maximum probability, and the reward ($r$) obtained is stored in the experience memory (D) along with the state before ($s$) and after ($s_{t+1}$) the action.

3- Choose a model that improves the policy.

4- Go to 2 and gather more experience with the improved policy and improve policy.

In other words, a policy iteration is a common approach to finding an optimal policy that maximizes each state's expected cumulative discounted reward. Using these techniques is useful when we have several options, and each has different rewards and risks. Policy iteration is a two-stage process alternating between policy evaluation and policy improvement.

In the policy evaluation stage, we intend to find the exact value function for our current policy. We repeat the Bellman equation defined as Eq. (1) over and over time to achieve this goal.

$$V_{\pi}(s) = \sum_{s',r} \; p\,(s',\, r|s, \pi(s))[\, r + \gamma V_{\pi}(s')] \tag{1}$$

where $s'$ denotes the next state and, $\pi(s)$ is action from state $s$ under policy $\pi$, $p$ denotes transition probability from state $s$ to next state $s'$ when doing action $\pi(s)$ using policy $\pi$ and receiving reward $r$. The $\gamma \in (0,1)$ is a discount factor. In fact, it can be said that the reward may not be given to the agent instantly. Early rewards are more likely to occur because they are more predictable than long-term future rewards. In a sequence, nearby rewards, even if larger, are discounted because the agent is not sure that it will be able to receive them. A discount rate called gamma is defined as discounting rewards. This value must be between 0 and 1. The larger the gamma, the smaller the discount. This means that the agent gives more importance to long-term rewards. On the other hand, the smaller the gamma, the greater the discount. This means that the agent pays more attention to short-term rewards.

The policy improvement stage (Eq. (2)) is carried out by repeatedly applying the Bellman optimality operator:

$$\pi'(s) = argmax_a \sum_{s',r} \; p\,(s',\, r \mid s, a)[\, r + \gamma V_{\pi}(s')] \tag{2}$$

Algorithm 1 shows the summary of policy iteration, which usually converges with a small number of iterations to the optimal policy ($\pi^*$) and value function ($v^*$).

| Algorithm 1. Policy Iteration for estimating $\pi \approx \pi^*$ |
|---|
| 1. Initialization |
| $V(s) \in R$ and $\pi(s) \in A(s)$ arbitrarily for all $s \in S$ |
| 2. Policy Evaluation |
| Loop: |
| $\triangle \leftarrow 0$ |
| Loop for each $s \in S$: |
| $v \leftarrow V(s)$ |
| $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))[r + \gamma V(s')]$ |
| $\triangle \leftarrow (\triangle, \lvert v - V(s) \rvert)$ |
| until $\triangle < \theta$ ($a\ small\ positive\ number\ determining\ the\ accuracy\ of\ estimation$) |
| 1. Policy Improvement |
| Policy-stable $\leftarrow true$ |
| For each $s \in S$: |
| Old-action $\leftarrow \pi(s)$ |
| $\pi(s) \leftarrow argmax_a \sum_{s',r} p(s', r \mid s, a)[r + \gamma V(s')]$ |
| If Old-action $\neq \pi(s)$, then Policy-stable $\leftarrow false$ |
| If Policy-stable, then stop and return $V \approx v^*$ and $\pi \approx \pi^*$; else, go to 2 |

Similarly, the value of choosing action $a$ in state $s$ under policy $\pi$ is expressed as $Q_\pi$(s, a), which is the expected recursive reward of taking action $a$ in state $s$, and further actions are also chosen by the policy $\pi$ (Eq. (3)).

$$Q_\pi(s, a) = \sum_{s',r} p(s', r \mid s, a)[r + \gamma Q_\pi(s', a')] \quad (3)$$

The $Q_\pi$ is called the value-action function for the policy $\pi$. The value function of $V_\pi$ and $Q_\pi$ can be estimated with repetitive experiments. For example, suppose an agent follows a policy and averages the amount it receives from experiences for each situation after infinite repetitions. $V_\pi(s)$ will converge to the actual value in that case. If this average is kept for each state-action pair separately, then $Q_\pi(s,a)$ will be estimated and stored in the table. Such estimation methods are called Monte Carlo methods, which include averaging over many random samples of the absolute return reward.

It is evident that for complex and dynamic problems such as stock trading and portfolio optimization with high dimensions and continuous state-action spaces, it is often impossible to find the exact optimal answer via explained lookup table-based method in Algorithm 1. Therefore, we must use rough approximates such as neural networks. The neural network consists of several layers that receive the input layer of the state ($s$) vector, and the output layer determines the action ($a$). Figure 4 shows the training process of an agent based on Q Network with experience replay memory. This architecture consists of three main parts:

- Q-network Q($s$, $a$;θ) where θ determines the agent's behavioral policy,
- Q-target network Q($s'$, $a'$;θ′), which is used to obtain the Q values for the error part of the Deep Q-Network (DQN) and
- Experience Replay Memory, which the agent uses to transfer samples to train the Q network randomly.

The idea of using the replay memory part is that considering that consecutive examples in the problem usually have a high correlation, using them consecutively for training the network reduces convergence. To solve this problem by using this memory (representation of experience) transitions from one state to another state with a specific action and associated reward are stored in a memory and are randomly selected from the samples in this memory. It is used for network training. Because these obtained samples are valuable, this part allows us to use them several times. The target network is based on the same structure as the leading network, which is copied to the target network $\theta'$ after a fixed number of steps. This will reduce the negative impact of network fluctuations and make training more stable and converge faster.

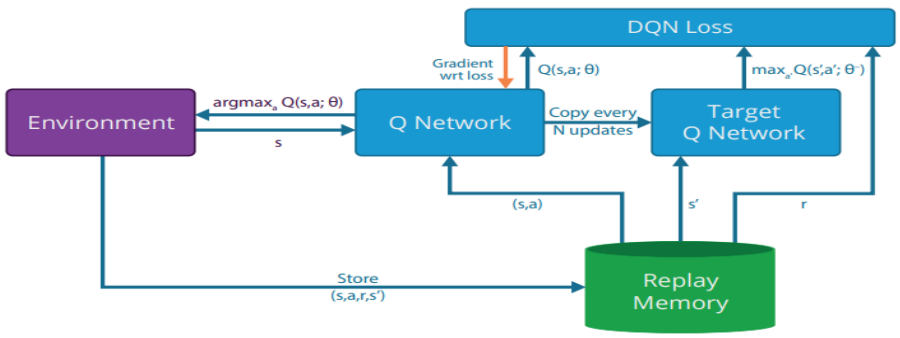


**Figure 4. DQN architecture**

Algorithm 2 briefly implements the deep Q learning process using experience replay memory.

| Algorithm 2. Deep Q-Learning with Experience Replay |
|---|
| Initialize replay memory D to capacity N |
| Initialize action-value function Q with random weights $\theta$ |
| Initialize target action-value function $\hat{Q}$ with weights $\theta' = \theta$ |
| for an episode from 1 to M, do |
| Initialize sequence $s_1$ and preprocessed sequence $\phi_1 = \phi(s_1)$ |
| for *t* from 1 to T, do |
| With probability $\epsilon$, select a random action $a_t$ |
| Otherwise select $a_t = argmax_a Q(\phi(s_t), a; \theta)$ |
| Execute action $a_t$ and observe reward $r_t$ and state $s_{t+1}$ |
| Set $s_{t+1} = s_t, a_t$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$ |
| Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in D |
| Sample random mini-batch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$from D |
| Set $Y_j = \{r_j, \ for\ terminal\ \phi_{j+1}\ r_j + \gamma max_{a'} Q(\phi(s_t, a'; \theta'), \ for\ non-terminal\ \phi_{j+1}$ |
| Perform a gradient descent step on $(Y_j - Q(\phi_j, a_j; \theta))^2$ with respect |
| to the network parameters $\theta$ |
| Every C steps reset $\hat{Q} = Q$ |
| end for |
| end for |

Table 1 summarizes recent RL and DRL stock market trading and portfolio optimization methods.

**Table 1. DRL methods for stock market trading and portfolio optimization**

| Reference | Algorithm | Financial Market | Remarks |
|---|---|---|---|
| Li, Y. (2019) | DQN+A3C | US/China Stocks and Futures | Stacked denoising autoencoders (SDAEs) and long short-term memory (LSTM) used as function approximators. |
| Zhang, Z. (2020). | DQN+A2C+Policy Gradient | Futures of commodity, equity index, fixed income, and forex | Time series momentum and technical indicators used to form state representations |
| Wu, X. (2020) | GDQN+ GDPG | US, UK, and Chines Stocks | Gated Recurrent Unit (GRU) used to extract informative financial features |
| Tsantekidis, A. (2020) | DDQN+ PPO | Forex | LSTM used as a time series feature extractor |
| Ponomarev, E. (2019) | A3C | Russia Stock Market | Recurrent layers used without dropout function in the testing |
| Briola, A. (2021) | PPO | Intel Corporation stock | High-frequency Limit Order Book data used to trade one unit of asset |

## The Agent and the Environment

Automate portfolio management means the agent periodically divides the capital between different assets. A trading robot should periodically make decisions to optimize the stock portfolio. For this purpose, the more practical information the agent has about the trading environment and the factors affecting it, the richer state (as input vector fed into DQN to generate portfolio vector) includes the estimator network better can predict the action and leads to the more optimal the robot's decisions for allocating stocks to the portfolio. Historical OHLCV (Open-High-Low-Close-Volume) stock data, Technical Indicators (TI), Fundamental Information (FI), News data, and Expert Defined Features (EDF) are example data that can be given to the agent as the state of the environment.

The state of the environment has a significant impact on portfolio performance. This paper uses different environments to investigate the impact of different features on portfolio profitability. Many of the existing methods for portfolio optimization are essentially the expansion of diversification methods for assets in the investment. Significant drawdowns and early entry into the share still need to be improved in portfolio construction. The idea here is that having a portfolio based on net money flow is less risky than only allocating a portfolio based on historical data. In this work, we are moving towards

addressing these issues by defining a novel risk indicator based on the intelligent behavior of smart money to maximize the performance of a DRL-based trading system. The following section introduces two types of Iranian stock market traders. We then design DRL-agent and multi-layer forecasting models with MNF risk indicators to optimize the stock portfolio. According to our latest knowledge, no study has been published in the field of reinforcement learning in the trading environment based on buying and selling of real/legal market traders.

## Research Methodology

This paper adds a novel perspective to portfolio management strategies. We calculate the net money flow indicator from the transaction data of the actual buyers of the Iranian capital market, which includes the number of buyers and the purchase volume of each share, and we use it to enter and exit the share and manage the portfolio. In addition to this indicator, we use Google Trends, which shows the trader's desire and interest in entering the capital market, to measure portfolio performance.

In this section, we first detail the principal of algorithmic portfolio management using DRL. Afterward, Google Trend features are introduced to enrich the environmental state and data augmentation. Finally, we define Money Net Flow Index as a new portfolio risk indicator.

### Algorithmic Portfolio Management

The automatic allocation of M risky assets in the portfolio to reach maximum profit and minimum risk is called algorithmic portfolio management. In the agent trading instructions, the trading strategy determines the share, volume, time, and price of the transaction.

In order to use DRL algorithms for portfolio management, we must precisely define the state space, action space, and reward function.

State space $S$ includes all the agent's observations of the environment at time t. It involves the cash balance, the number of shares in the portfolio, the price of each asset, the values of technical indicators, the ratios and basic information of the asset, and the values of the risk indices.

$$\boldsymbol{s_t} = [\boldsymbol{C_t}, \boldsymbol{c_t^1}, \dots \boldsymbol{c_t^M}, \boldsymbol{n_t^1}, \dots, \boldsymbol{n_t^M}, \boldsymbol{u_t^1}, \dots \boldsymbol{u_t^N}], \boldsymbol{t} = 1, \dots, \mathrm{T} \quad (4)$$

Where $C_t$ denotes the amount of cash in time t, $c_t^i, and\ n_t^i$ respectively describe the price and number of assets in time t, and $_{it}$ is technical indicators and expert-defined features.

Action space $A$ is authorized actions that our DRL agent can do, precisely the weight of buying or selling each share to reconstruct the portfolio.

$a_t$ = {sell action, $w(t) < 0$ buy action, $w(t) > 0$ no action, $w(t) = 0$ (5)

**Reward Function r**. We use portfolio returns as rewards.

$$r_t(s_t, a_t, s_{t+1}) = \frac{v(t)}{v(t-1)} - 1 \quad (6)$$

$v(t), v(t-1) \in R$ denote the portfolio value Eq. (7) at timeslot t and t-1, respectively.

$$v(t) = C_t + \sum \quad c_t^i * n_t^i \quad (7)$$

**Google Trends**

One of the features of Google is Google Trends, which shows the popularity of a keyword search in the Google search engine. The word Trend means people's search interests over time. Therefore, when we talk about the trend, we mean that we do not care about statistics and current information and are interested in the change process. According to this explanation, through this Google service, we can observe and check the search process of different words in the Google search engine.

Well, this is an excellent opportunity for stock market investors. Visit the Google Trends service and check the word "stock market." Therefore, you can recognize the trend of the public's attention to the stock market and, by nature, the trend of entering and leaving liquidity in the market.

Go to the Google Trends service at 'trends.google.com' and search for the word "stock market" in the place indicated at the top of the site. A graph will be displayed for you showing this word's search process in Google.

In fact, in this analysis, the long-term trend of the search rate of the word stock market is essential.


**Figure 5. 'Shebandar' Google trend**

Figure 5 shows how the word ‘Shebandar' (the Persian Symbol name of Bandar Abbas Oil Refinie) has trended in Google since 2019. Are the fluctuations of this chart related to the upward and downward trends of the stock market? Look at Figure 6 to answer this question.


**Figure 6. 'Shebandar' (PNBA) Price**

Indeed, this correspondence is not accidental, and the reason for it is pretty apparent when people search the word "stock market" more often on Google; it means that they are paying attention to it, and this attention is a sign of liquidity entering the market.

## Money Net Flow

In this paper, we develop the Money Net Flow market timing index. To determine the best time for buying and selling, we need to recognize the net flow of smart money. Smart money is capital entered into the market by investors, prominent financial specialists, investment companies, and market experts to create significant price trends in the share. The major investors of the market, with their high understanding of the market's needs as well as access to confidential information and more than other people, determine the direction of the stock movement with their power. Smart money is big enough to have sophisticated changes in the price trend of a stock and create significant market impacts. With the entry of smart money into the market, other parts of it are also affected, causing prices to increase or decrease. When the principal and prominent market investors recognize that they need to make a particular investment in a share according to the existing conditions, they start buying the desired share in high volumes and create a solid and positive wave on that share.

The positive net flow of money means the entry of money into a share, and the capital market, in general, can cause the growth of that share and the capital market. In general, the negative net flow of money from the share or the capital market can cause the decline of that share and the stock market.

Therefore, it is essential to identify smart money in the stock market and observe the net money flow in each share and the entire market.

## Detecting the entry and exit of smart money

- One of the most important factors by which the existence of smart money can be recognized is the volume of transactions. Considering the volume of transactions, the entry or exit of smart money to a stock or industry can be recognized. The volume of transactions is the number of transactions carried out in a specific period (daily, weekly, monthly), which can be seen in the bar chart of a share. By monitoring the volume of transactions and the price, traders can identify the direction of market movement.

Some traders believe that the price trend lags behind the trading volume trend. This means that the upward and downward trends can be seen in the volume of transactions before the price reversal trends are determined. According to one of the common Wall Street stock market principles, trading volume causes price changes. Trading volume is relatively heavy in booming and relatively light in stagnant markets. However, it cannot be said that only the volume of transactions accurately determines this issue.

Detecting the entry of smart money requires sufficient knowledge and experience. Based on experience, if the trading volume of a share increases suddenly when there is no demand for it, it is one of the signs of the arrival of smart money.

In contrast to the sharp reduction and heavy selling of a share at once, smart money can be considered withdrawn from it. Therefore, buying and selling a significant share in a short time and at once can be one of the signs of the presence of smart money in the share.

- Another sign of identifying the presence of smart money is comparing the number of buyers or sellers concerning the volume of transactions. In this way, if the number of buyers is low, but the volume is very high, it indicates that large investors are buying shares. On the other hand, the presence of few sellers but with a high volume of sales orders can indicate the exit of smart money.
- The value of daily transactions is also another factor to identify this issue. If the average value of daily transactions in the market is within a range and doubles to three times in a short time, it can be concluded that this high liquidity is the presence of smart money.

Of course, these factors can be very misleading. For example, if one day most of the symbols in the buying queue are locked, the value and volume of transactions will drop drastically, while no money has been taken out of the market.

Therefore, the medium-term trend of the value and volume of transactions is essential to us, not just seasonal fluctuations. If the volume and value of daily transactions were upward, it could be concluded that the net flow is positive, and if the volume and value of daily transactions were downward, it could be said that the net flow to the stock market is negative.

**Real traders and Legal traders**

In the financial market, we have two types of traders: real (individual) and legal (institutional). Ordinary people who conduct transactions individually and are not under the supervision of any special organization are called real traders, and the representatives of companies and organizations who conduct large transactions and the profit from these transactions will be for the company is called legal traders. As a result, the entry and exit of money in the stock market also point to the same issue. In fact, the entry and exit of this money are related to the liquidity of real traders.

Usually, the movements of legal traders, i.e., companies and large groups, are not aimed at gaining profit and include issues such as securing liquidity or joining the board of directors. Coffee C. (1991) and Hartzell T. (2003) show the tendencies of institutional investors in stock governance and securing liquidity. For this reason, the movements of legal traders are not very important in determining the market trend.

However, real traders put their capital in the market intending to make a profit, and if they buy a share, they want to profit from it. So, the actual movements in the market are significant. Checking the power of buyers and sellers is primarily focused on the real ones.

Many traders believe that the future of shares can be known by relying on actual sales statistics. It must be said that this idea needs to be corrected and may lose capital. In some cases, although real traders have had good sales, their shares will still increase in price.

On the other hand, even though the shares of real traders have had a good purchase, they are involved in a price drop. Therefore, the net flow of the purchase and sale of real traders should be estimated to obtain accurate statistics of a share's current and future status. Therefore, paying attention to the entry and exit of money in the stock market prevents users from going astray.

With the definitions of the ratio of buyer power to seller power in the financial field, we have the following:

- **Purchase per capita (PPC):** The amount of liquidity that enters a share in one day by factual or legal shareholders, which can indicate the rising share in the near future. If this average amount is calculated for real traders, it is called actual purchase per capita. If it is calculated for legal traders, it is called legal purchase per capita.

Calculating the purchase per capita is also simple, and it is obtained by dividing the purchase amount of that share by the number of buyers. If we consider the actual buyers, we reach the actual purchase per capita, and if we consider the legal ones, we reach the legal purchase per capita. Its calculation method is similar to the GPA method.

$$PPC = \frac{vol_{buy}}{N} \tag{8}$$

- **Sales per capita:** (SPC) Sales per capita are the same as purchase per capita, but one share is calculated for daily sales. This means that the per

capita sale of a share is equal to the average sales of that share in one day. This per capita is also obtained by dividing the real sales value of a particular stock by the number of its sellers. Regarding per capita sales, we can also calculate factual and legal per capita purchases separately.

$$SPC = \frac{vol_{sell}}{N} \tag{9}$$

PPCr and SPCr can be calculated in two ways: Volume or Riyal, volume per capita real purchase/sell indicates the average volume purchased/sold from that share in one day. To calculate it, we divide the actual purchase volume (in terms of the number of shares) by the number of buyers. Riyal per capita real purchase/sold represents the amount of money every real trader has spent on average for that share. To calculate it, we will follow the same method of calculating the real volume purchase/sell per capita, but we will multiply the obtained number by the final price.

- Volume Real Purchases (VPPCr) and Sales Per Capita (VSPCr):

$$VPPC_r = \frac{vol_{buy_r}}{N_r} \tag{10}$$

$$VSPC_r = \frac{vol_{sell_r}}{N_r} \tag{11}$$

Riyal Real Purchase (RPPCr) and Sales Per Capita (RSPCr):

$$RPPC_r = \frac{Price * vol_{buy_r}}{N_r} \tag{12}$$

$$RSPC_r = \frac{Price * vol_{sell_r}}{N_r} \tag{13}$$

- Average True Range (ATR): ATR is a market volatility indicator. The first step in calculating ATR is to find a series of true range values for an asset. The price range of an asset for a given trading day is its high minus its low. Meanwhile, the true range is more encompassing and is defined as:

$$TR = Max[(High - Low), Abs(High - Close_p), Abs(Low - Close_p)] \tag{14}$$

Where high and low the maximum and minimum are prices of the current day, and $Close_p$ is the previous day's price. The ATR is a moving average of n days of the true ranges.

$$ATR = \frac{1}{n}\sum_{i=1}^{n} \blacksquare TR_i \tag{15}$$

- **Up day:** A Day is an up day when its typical price – the average of high, low, and close – is higher than yesterday's price plus 1/10 ATR.

$$\underline{Price} = \frac{High + Low + Close}{3}$$
$$\underline{Price}^{+} = \{True, \quad \underline{Price} > \underline{Price}_p + \frac{ATR}{10} \quad False, \quad \underline{Price} \not> \underline{Price}_p + \frac{ATR}{10} \tag{16}$$

- **Down day:** A Day is a down day when its typical price – the average of high, low, and close – is lower than yesterday's price minus 1/10 ATR.

$$\underline{\text{Price}}^{-} = \{\text{True}, \quad \underline{\text{Price}} < \underline{\text{Price}}_{\text{p}} - \frac{\text{ATR}}{10} \quad \text{False}, \quad \underline{\text{Price}} \not< \underline{\text{Price}}_{\text{p}} - \frac{\text{ATR}}{10} \tag{17}$$

- **Positive money net flow:** We add the difference of per capita purchases and sales together on up days to get the positive money net flow.

$$MNF^{+}_{buy_r} = \sum_{\blacksquare}^{\blacksquare} \blacksquare RPPC_{buy_r}{}^{+}$$
$$MNF^{+}_{Sell_r} = \sum_{\blacksquare}^{\blacksquare} \blacksquare RSPC_{Sell_r}{}^{+} \tag{18}$$

- **Negative money net flow:** We add per capita purchases(sales) on down days to get the negative money net flow.

$$MNF^{-}_{buy_r} = \sum_{\blacksquare}^{\blacksquare} \blacksquare RPPC_{buy_r}{}^{-} \tag{19}$$

$$MNF^{-}_{Sell_r} = \sum \blacksquare RSPC_{Sell_r}{}^{-}$$

- **Money Inflow:** Money inflow refers to the positive money net flow of buyers minus the positive money net flow of sellers. If it is positive, we have money coming into the share, and the probability of a stock price rise is higher than a fall.

$$\overrightarrow{MNF} = MNF^{+}_{buy_r} - MNF^{+}_{Sell_r} \quad (20)$$

- **Money Outflow:** Money outflow refers to the negative money net flow of buyers minus the negative money net flow of sellers. If it is negative, we have money withdrawn from the share, and the probability of a stock price fall is higher than a rise.

$$MNF^{\leftarrow} = MNF^{-}_{buy_r} - MNF^{-}_{Sell_r} \quad (21)$$

- **MNF index:** The money net flow index is the moving average difference between a stock's money inflows and outflows within a given period.

$$MNF(Period) = \frac{\sum^{Period} (\overrightarrow{MNF} - MNF^{\leftarrow})}{Period} \quad (22)$$

This motivation and formulation are empirically validated in this paper and showed that by using the MNF index as a risk indicator, the drawdown could be controlled, and the profit performance of the strategy would be guaranteed.

## Results

All the experiments are carried out on a computer having 16 GB RAM with CPU Intel Core i7-10750H and GPU Nvidia GeForce GTX 1080 8 GB dedicated memory, 80 GB of virtual memory has been used to optimize the parameters. Data collection consists of three parts. Historical price data, historical data of factual and legal traders, and historical data of Google Trends. Following subsections describe details of framework implementation.

### DRL Algorithms

The proposed method to optimize portfolio returns is trained using state of the arts DRL algorithms: SAC, A2C shown in Figure 3.

As we explained, just as a human trader needs to analyze different information before making a trade, our trading agent also observes many different features to learn better in an interactive environment. The state space describes an agent's perception of the market. In this paper, we examine different environments with different state spaces, which are briefly described in the following:

- **OHLCV (OHLCV):** We have only used OHLCV historical data in this environment.
- **OHLCV augmented with Google Trend (GT)**: We use the Google Trend data of each stock to enrich the features of the stock.
- **OHLCV and Real and Legal sales data in raw form (RL):** In this environment, we use the factual and legal data of the asset board, such as the number of buyers and sellers of each and the volume of each purchase and sale without pre-processing.
- **OHLCV and Technical Indicators (TI):** MACD, RSI, CCI, SMA, DX, and Bollinger Bands technical indicators have created this environment.
- **OHLCV and Expert defined Features (EF):** The percentage of price changes with different lags has been used to build this environment.
- **OHLCV and MNF index (MNF):** The MNF index has been used as a risk indicator to build this environment.

## Datasets

We use historical OHLCV, purchase and sale data of factual and legal traders, and Google trend data on 30 large companies' stocks in the Tehran Stock Exchange to evaluate strategies. The English names and symbols of these companies are listed in Table 2.

**Table 2. 30 large TSE companies**

| Symbol | Name | Symbol | Name |
|---|---|---|---|
| MKBT | Iran Tele. Co. | BFJR | Fajr E.Persia Gulf |
| PARS | PARS Petrochemical | PASN | S*Parsian Oil&Gas |
| PTAP | Tamin Petro. | PJMZ | Jam Petr. |
| KSHJ | S*IRI Marine Co. | IKCO | Iran Khodro |
| MAP | MAPNA | PNTB | Tabriz.Oil.Refine |
| PNBA | BA Oil Refinie | PRDZ | Pardis Petr. |
| PNES | S*Isf. Oil Ref. Co. | PTEH | Palayesh Tehran |
| TAMN | Social Sec Inv | PKLJ | Khalij Fars |
| FKHZ | Khouz. Steel | MSMI | S*I. N. C. Ind. |
| FOLD | S*Mobarakeh Steel | CHML | Chadormalu |
| GOLG | Gol-E-Gohar. | MON | Mobin Petr. |
| NORI | NORI Petrochemical | MDKO | Khavarmianeh Ins. |
| HMRZ | Iran Mobil Tele | BMLT | S*Mellat Bank |
| MADN | S*Metals & Min. | SAND | Pension Fund |
| GDIR | Ghadir Inv. | BPAS | S*Pasargad Bank |

Each experiment's dataset time range is historical daily price and rea/legal and Google Trend data from March 1, 2020, to August 22, 2022. Data from March 1, 2020, to December 1, 2021 (339 Days) is used as the training set. The data from December 2, 2021, to March 1, 2022 (57 Days) is used as the validation set. The remaining data from March 2, 2022, to August 22, 2022 (105 Days) is used as the test set. We train our agent on the training data, then select the model hyperparameters by evaluating criteria such as Sharpe ratio on validation data, and finally backtest the selected model on the test data.

## Evaluation metrics

We use the following metrics to evaluate the portfolio allocation strategies' performance.

- **Annualized Return:** The geometric average amount of money an investment strategy earns each year over a given period.
- **Annualized Volatility:** The annual standard deviation of portfolio returns indicates the strength of the factor.
- **Cumulative return:** reflects the overall effect of the trading strategy in a certain period
- **Sharpe ratio** [Sharpe (1998)]: return earned per unit of volatility, a widely used measure of an investment's performance.

- **Maximum Drawdown:** The maximum percentage loss during the trading period.
- **Profit and Loss (P&L):** The algorithm's profit or loss in the desired period.

## Hyperparameter optimizations

Reinforcement learning algorithms are very sensitive to hyperparameter values, and one of the most time-consuming reinforcement learning processes is the optimization phase of hyperparameters. Figure 7 shows the process of determining the optimal number of episodes for training the reinforcement agent in the training phase. The algorithm convergence is seen in episode 5000, but in episode 200, the agent's reward output fluctuates significantly.

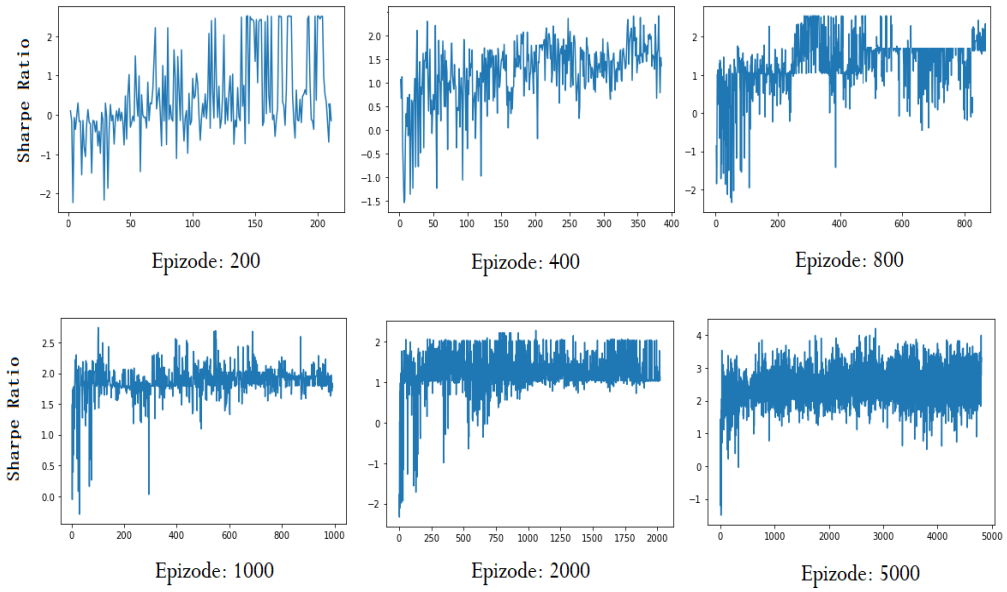


**Figure 7. The effect of the number of episodes on the convergence of the algorithm**

Batch size, learning rate, experience memory size, discount rate value, and deep learning network parameters are other parameters that need to be optimized.

## Explanation Analysis

In all experiments, the initial amount of cash balance is 1000,000. Figure 8 shows the change in portfolio value based on the number of episodes during

agent training. As it is clear from picture (a) to picture (f), with the increase in the number of episodes, the agent stores more experiences and reaches the maximum portfolio value. In a way, in episode 443, it has reached the maximum assets of nearly 2,000,000, and as the number of episodes has increased in the training process, it has reached the maximum portfolio value of more assets at the end of the time step.

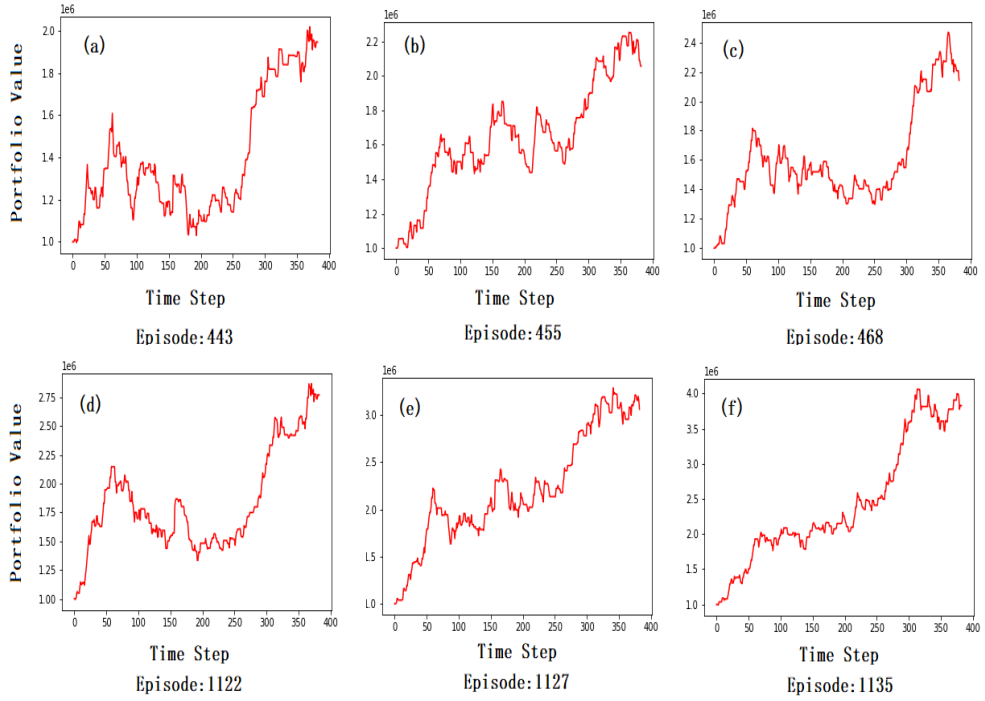


**Figure 8. Training Account Value Progress (SAC + MNF) in various episodes**

As shown in Figure 7, the portfolio's value only sometimes increases in all episodes. As mentioned in Algorithm 2, in some cases (with a probability of $\epsilon$), agents focus on improving their knowledge about each action instead of getting more rewards. By selecting random actions, they can get long-term benefits. So, agents work on gathering more information to make the best overall decision. We repeated these experiments many times, and this trend was observed in all of them, and for example, we took only six episodes in Figure 8.

## Environments

Figure 9 illustrates the cumulative return of the SAC algorithm with the proposed MNF index as the risk indicator, SAC with turbulence as risk

aversion, and the benchmark index of 30 large shares (LCI30) as a Buy and Hold (B&H) strategy. SAC behaves like the LCI30 without the smart money flow index, but the efficiency increases when the money flow index is used as market timing. The proposed model using money flow has reached the maximum efficiency of 68%, while the maximum efficiency of the DRL algorithm without the money flow sensor is 30%.


**Figure 9. Cumulative Return of sac and sac+MNF and B&H Baseline (LCI30)**

The different evaluation criteria of the strategy based on the MNF indicator of entry and exit of smart money and the reinforcement learning strategy that formed the portfolio based on the OHLCV data and turbulence as risk aversion are given in Table 3. The strategy based on the smart indicator has a maximum loss of 10.61, which is less than the price strategy of 3.17, and the maximum drawdown duration of 38 days has been reduced to 8 days, while the annual profit from building a portfolio based on the smart money index is 115.91%, which is 3.88 times better than the 30% profit of the OHLCV -based strategy.

**Table 3. summarizing the performance of the trading activity by sac and sac+MNF**

| Performance Indicator | SAC + MNF | SAC |
|---|---|---|
| Profit & Loss (P&L) | **537363** | 126284 |
| Annualized Return | **115.91** | 30.26% |
| Annualized Volatility | 38.50% | **30.39%** |
| Sharpe Ratio | **2.623** | 1.001 |
| Maximum Drawdown (MDD) | 10.61% | 13.78% |
| MDD Duration | **8 days** | 38 days |

## Drawdown Analysis

To more accurately show the performance of the money flow index, this index is shown along with the index of 30 significant stocks (LCI30) so that the signals of entry and exit from the market are shown simultaneously with the

upward and downward trends of the LCI30. As shown in Figure 10, selling based on the money exit indicator during the market crash keeps the portfolio safe from significant losses, and buying based on smart money entry prevents early buying.

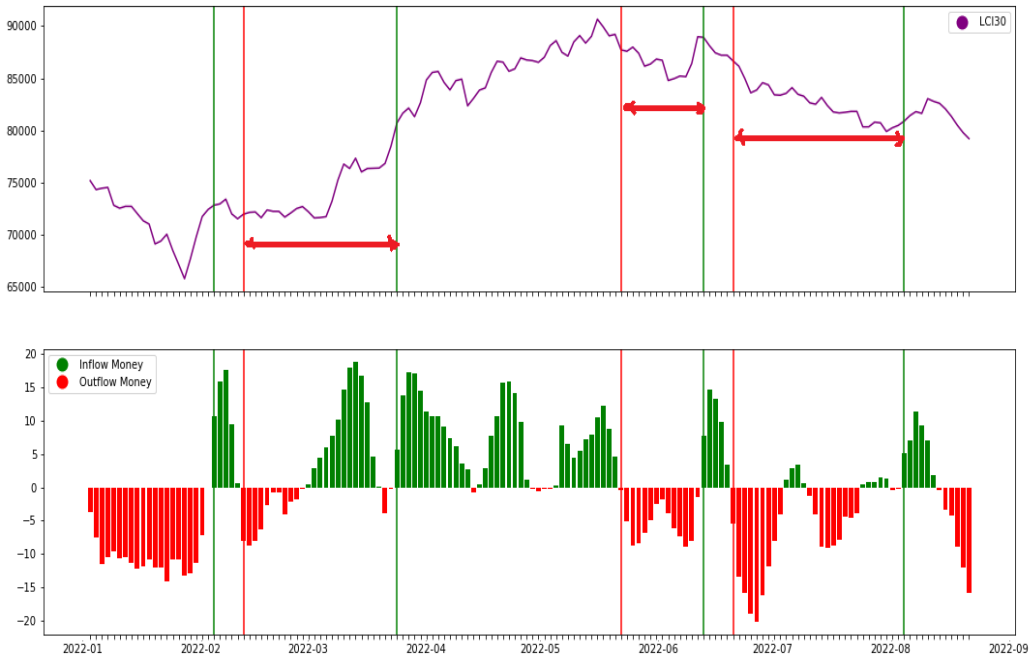


**Figure 10. Controlled DrawDowns with MNF Indicator**

## Evaluations

The analysis of different environments for the A2C DRL agent is shown in Figure 11. Two new combined environments have also been examined, including real/legal data and technical indicators (RL+TI) and real/legal data and Google Trend data (RL+GT). The cumulative daily return of the presented algorithm with the money flow index has surpassed the rest of the algorithms. The proposed algorithm has also gained more efficiency than a model with features defined by the expert. These features include the variance of daily stock return changes. Of course, further experiments can also be done for other features, which will be postponed to future works. The tabular results of different evaluation criteria are given in Table 3.

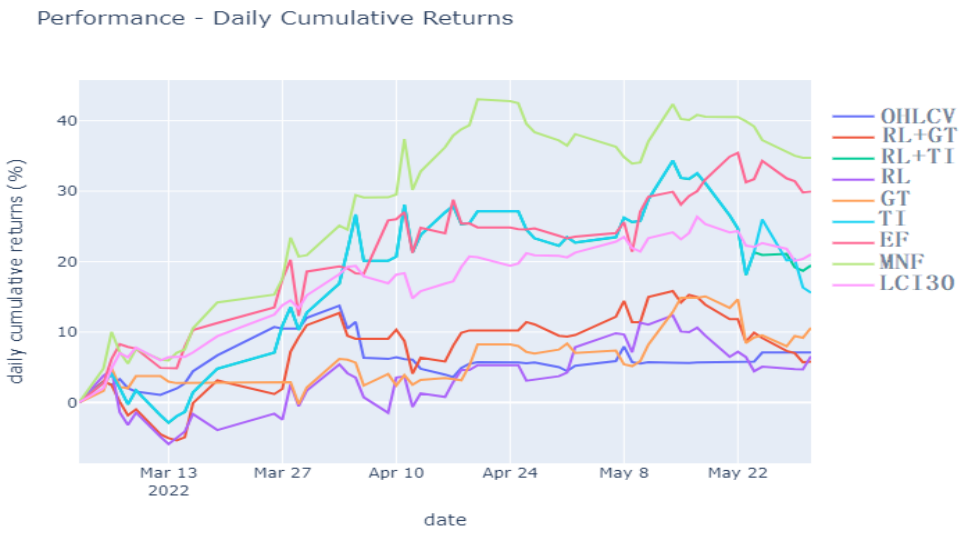


**Figure 11. Returns. A2C with different environments**

The A2C algorithm using MNF received 347,466 over 1000,000 during the trade period and obtained an annualized return of 200.18%, which outperformed all other environments. The proposed model has a minimum Maximum Draw Down duration of 1 day, which indicates the risk-free model.

**Table 3. Returns. A2C DRL algorithm with different environment**

| Performance Indicator | MNF | EF | TI | GT | RL | RL + TI | RL + GT | OHLCV |
|---|---|---|---|---|---|---|---|---|
| Profit & Loss (P&L) | 347466 | 299485 | 155703 | 106210 | 65735 | 194604 | 58181 | 70980 |
| Annualized Return | 200.18 % | 169.59 % | 85.13 % | 52.01 % | 35.29 % | 105.88 % | 29.91 % | 32.99% |
| Annualized Volatility | 33.20% | 35.78% | 41.06 % | 26.15 % | 34.77 % | 38.52% | 29.92 % | 19.26% |
| Sharpe Ratio | 4.146 | 3.423 | 1.766 | 1.839 | 0.984 | 2.237 | 0.985 | 1.671 |
| Maximum Drawdown (MDD) | 6.39% | 6.60% | 13.98 % | 6.17% | 10.07 % | 12.08% | 8.73% | 8.93% |
| MDD Duration | 1 day | 10 days | 13 days | 6 days | 5 days | 7 days | 12 days | 10 days |

## Conclusion

This paper has investigated the impact of smart money net flow on portfolio optimization based on the state of the art's deep reinforcement algorithms, which is potentially economically profitable and attractive for investment.

In more detail, we investigated deep reinforcement learning algorithms using different environments on Iranian stock market data. Experimental results in out-of-sample data using A2C and SAC show 200.18% and 115.91, respectively, an annualized return superior to all other environments. Optimization of SAC and A2C parameters was done on TSE historical data. Optimizations depend highly on the date range, bullish or bearish market, and the stocks used. Building a portfolio based on the entry and exit of smart money is less risky and more profitable than conventional turbulence. The experimental results show a 1-day Maximum Drawdown Duration, the lowest value compared to other environments. In subsequent continuations of this work, the use of various reward functions is disputable. The smart money combination of other markets, such as the dollar and stocks, can be tracked in future works.

### Declaration of Conflicting Interests

The authors declared no potential conflicts of interest concerning the research, authorship and, or publication of this article.

### Funding

The authors received no financial support for the research, authorship and, or publication of this article.

## References

Almahdi, S., & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, 87, 267-279.

Almahdi, S., & Yang, S. Y. (2019). A constrained portfolio trading system using particle swarm algorithm and recurrent reinforcement learning. *Expert Systems with Applications*, 130, 145-156.

Briola, A., Turiel, J., Marcaccioli, R., & Aste, T. (2021). Deep reinforcement learning for active high-frequency trading. arXiv preprint arXiv:2101.07107.

Coffee Jr, J. C. (1991). Liquidity versus control: The institutional investor as a corporate monitor. Colum. L. Rev., 91, 1277.

Faturohman, T., & Nugraha, T. (2022). Islimic stock portfolio optimization using deep reimforcement learning. *Journal of Islamic Monetary Economics and Finance*, 8(2), 181-200.

Filos, A. (2019). Reinforcement learning for portfolio management. arXiv preprint arXiv:1909.09571.

Francis, D. (2022). Portfolio Management for Asset Forecasting Using Recurrent Neural Network (Doctoral dissertation, Dublin Business School).

García-Galicia, M., Carsteanu, A. A., & Clempner, J. B. (2019). Continuous-time reinforcement learning approach for portfolio management with time penalization. Expert Systems with Applications, 129, 27-36.

Hartzell, J. C., & Starks, L. T. (2003). Institutional investors and executive compensation. *The Journal of Finance*, 58(6), 2351-2374.

Jiang, Z., & Liang, J. (2017, September). Cryptocurrency portfolio management with deep reinforcement learning. In 2017 Intelligent Systems Conference (IntelliSys) (905-913). IEEE.

Kang, Q., Zhou, H., & Kang, Y. (2018, October). 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 (141-145).

Kanwar, N. (2019). Deep reinforcement learning-based portfolio management. Ph.D. dissertation, The University of Texas at Arlington.

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. IEEE Access, 7, 108014-108022.

Lin, Y. C., Chen, C. T., Sang, C. Y., & Huang, S. H. (2022). Multiagent-based deep reinforcement learning for risk-shifting portfolio management. Applied Soft Computing, 123, 108894.

Markowitz, H., M.Fabozzi, F. J., & Gupta, F. (2002). The legacy of modern portfolio theory. The Journal of Investing, 11(3), 7-22.

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

Park, H., Sim, M. K., & Choi, D. G. (2020). An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications*, 158, 113573.

Ponomarev, E. S., Oseledets, I. V., & Cichocki, A. S. (2019). Using reinforcement learning in the algorithmic trading problem. *Journal of Communications Technology and Electronics*, 64(12), 1450-1457.

Salisu, A. A., Demirer, R., & Gupta, R. (2022). Financial turbulence, systemic risk, and the predictability of stock market volatility. *Global Finance Journal*, 52, 100699.

Sato, Y. (2019). Model-free reinforcement learning for financial portfolios: a brief survey. arXiv preprint arXiv:1904.04973.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *nature,* 529(7587), 484-489.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT Press.

Tsantekidis, A., Passalis, N., Toufa, A. S., Saitas-Zarkias, K., Chairistanidis, S., & Tefas, A. (2020). Price trailing for financial trading using deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7), 2837-2846.

Weng, L., Sun, X., Xia, M., Liu, J., & Xu, Y. (2020). Portfolio trading system of digital currencies: A deep reinforcement learning with multidimensional attention gating mechanism. N*eurocomputing*, 402, 171-182.

Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020). Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, 538, 142-158.

Ye, Y., Pei, H., Wang, B., Chen, P. Y., Zhu, Y., Xiao, J., & Li, B. (2020, April). Reinforcement-learning-based portfolio management with augmented asset movement prediction states. *In Proceedings of the AAAI Conference on Artificial Intelligence*.34(01), 1112-1119.

Yue, H., Liu, J., Tian, D., & Zhang, Q. (2022). A Novel Anti-Risk Method for Portfolio Trading Using Deep Reinforcement Learning. Electronics, 11(9), 1506.

Zhang, Z., Zohren, S., & Roberts, S. (2020). Deep reinforcement learning for trading. *The Journal of Financial Data Science*, 2(2), 25-40.

**Bibliographic information of this paper for citing:**

Khonsha, Samira; Agha Sarram, Mehdi & Sheikhpour, Razieh (2023). A Profitable Portfolio Allocation Strategy Based on Money Net-Flow Adjusted Deep Reinforcement Learning. *Iranian Journal of Finance*, 7(4), 59-89.

Copyright © 2023, Samira Khonsha, Mehdi Agha Sarram and Razieh Sheikhpour.