Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 3, No. 1, Winter & Spring 2023
Research paper

A novel financial trading system based on reinforcement learning and technical analysis applied on the Tehran securities exchange market

Zahra Pourahmadi[1], **Dariush Farid**[2], **Hamid Reza Mirzaei**[3]

[1] Faculty of Management, Economics and Accounting, Yazd University, Yazd, Iran
z.pourahmadi@stu.yazd.ac.ir

[2] Faculty of Management, Economics and Accounting, Yazd University, Yazd, Iran
fareed@yazd.ac.ir

[3] Faculty of Management, Economics and Accounting, Yazd University, Yazd, Iran
hmirzaei@yazd.ac.ir

Abstract:
Stock trading is a significant decision-making problem in asset management.This study introduces a financial trading system (FTS) that leverages artificial intelligence (AI) techniques to automate buy and sell orders specifically in Iran's stock market. Due to limited availability of labelled data in financial markets, the FTS utilizes reinforcement learning (RL), a subset of AI, for training. The model incorporates technical analysis and a constrained policy to enhance decisionmaking capabilities. The proposed algorithm is applied to the Tehran Securities Exchange, evaluating its efficiency across 45 periods using three different stock market indices. Performance comparisons are made against common strategies such as buy and hold, randomly selected actions, and maintaining the initial stock portfolio, with and without transaction costs. The results indicate that the FTS outperforms these methods, exhibiting excellent performance metrics including Sharp ratio, PP, PF, and MDD. Consequently, the findings suggest that the FTS serves as a valuable asset management tool in the Iranian financial market.

Keywords: Algorithmic trading, Investment portfolio, Machine learning, Reinforcement learning, Stock exchange.

1 Introduction

Artificial intelligence (AI) and machine learning (ML) have made significant advancements in various fields, especially in complex and expansive problems. Financial markets have also seen a growing trend in the use of trading algorithms to make investment decisions. AI and ML are attractive to finance researchers because of their more appropriate and homogeneous capabilities than statistical

[2]Corresponding author
Received: 04/06/2023 Accepted: 03/08/2023
https://doi.org/10.22054/JMMF.2023.74166.1088

methods [14]. Machine learning is the science of designing machines to operate based on given data and experiences, without being entirely programmed. It includes sub-disciplines such as supervised, semi-supervised, and reinforcement learning (RL) [3]. RL does not require predicting stock price trends before determining investment strategies, making it advantageous over other ML sub-sections. It trains decision-making power using past information and selects optimal actions based on the probability of (action, state). RL also calculates and considers the possible future rewards of current decisions [5], [4].

This study aims to develop an algorithmic trading model based on RL to provide a trading strategy that considers transaction fees. The innovation in this research combines technical analysis data and primary data prices in training the model and uses a restricted reinforced policy instead of pure reinforced learning. An RL model maximizes rewards received over time, so the agent understands the environment and takes optimal actions to achieve its goals. The agent is rewarded or punished based on the state and effect of the completed action on the environment [1]. By developing an algorithmic trading model based on RL, this study contributes to the advancement of the financial market's automated trading system.

Upon reviewing the literature on the subject, it becomes apparent that the development of trading algorithms using reinforcement learning, a sub-field of machine learning, has become a popular research topic among computer science and financial researchers seeking to improve the modelling and implementation of these algorithms. These upgrades typically involve changes to various aspects of the algorithm implementation, such as the input data, definition of reward and penalty functions, value function, and policy, to obtain a combination that can provide reasonable and profitable trading suggestions in most cases. In this paper, we propose an innovative approach to improving the performance of the trading algorithm by changing the input data and applying restrictions on the policy, using a controlled form of reinforcement learning. Given the lack of studies in the Tehran stock market, we present an efficient trading algorithm based on reinforcement learning specifically for financial markets in Iran.

The following section provides a brief overview of the theoretical foundations and research background. In section 3, we describe the components of the proposed model, including the environment states, current position, policies, and reward functions. We then discuss the research methodology and objectives in determining the reward functions. Section 4 presents the results of a numerical problem to demonstrate the effectiveness of our proposed model compared to other strategies, and the final section briefly summarizes our findings.

2 Literature review

One hotly debated topic in the financial field is algorithmic trading, which includes methods based on prior knowledge and machine learning [6]. Examples of prior

knowledge methods are trend following [7], mean-reversing [8], and delta-neutral trading strategies [9], [10]. In recent years, machine learning has gained more attention from researchers in various financial subjects, including option pricing optimization [11], [12], multi-stage portfolio investment [14], [13], and risk management [4]. Machine learning aims to learn trading strategies in algorithmic trading and to find profitable strategies using historical price data. It can be divided into the supervised and reinforcement learning approach.

The supervised learning method attempts to predict the next time point's stock price or price trend, but it requires other prior knowledge to decide on trading actions [15]. As a result, it leads the model to have two error propagations: the error of price prediction and the error caused by the decision-making process. Moreover, the supervised method cannot consider environmental factors such as transaction cost and the characteristic of delayed return, resulting in algorithmic trading [16]. Examples of supervised learning-based algorithmic trading are [21], [18], [20], [17], [19], [6].

In contrast, the reinforcement learning method is more suitable for problems defined by long-term goals and delayed returns than other machine learning methods [22]. Existing methods for algorithmic trading based on reinforcement learning can be divided into policy-based and value-based methods [23]. Examples of algorithmic observation using RL for policy can be found in [25], [27], [24]. In policy-based methods, the problem of the environment is continuous and non-discretized, and the state is not allowed, which limits its ability to guarantee convergence [28]. Value-based methods, on the other hand, estimate the value function (state, action) repeatedly and then select an action based on the value [29], [30], [31].

The above studies used simple artificial neural networks for their policy network, but the combination of deep artificial neural networks and reinforcement learning has created a new type of reinforcement learning called deep reinforcement learning (DRL), which can be found in studies such as [32], [33]. Actor-Critic is a combination of policy and value-based methods that were developed to take advantage of all the benefits of both methods while removing their limitations. Models based on actor critics are presented in [35], [34]. In the Actor-Critic model, both the function of (policy, action) and the function of (state, value) are optimized. These two parts participate in a game and both become better in their role over time, resulting in the overall structure learning to solve the problem more efficiently than the two separate methods. The operator takes the state as input and provides the best action as output based on what it has learned so far. Other algorithms derived from the actor-critic method include PPO and A2C.

In this study, the same original and pure algorithm is selected for model development, which helps to focus on items and methods that increase the general efficiency of the model. The details of our proposed scheme are presented in the next subsection

3 Proposed algorithm

In this research, we have used Python programming to develop and present a stock trading algorithm. Our algorithm aims to achieve the best possible performance by maintaining simplicity and considering real market conditions, such as transaction costs. To accomplish this, we have creatively defined the states, rewards, and goal functions. The components and execution process of our algorithm are explained below, and the schematic view of the RL algorithm is presented in Fig. 1.

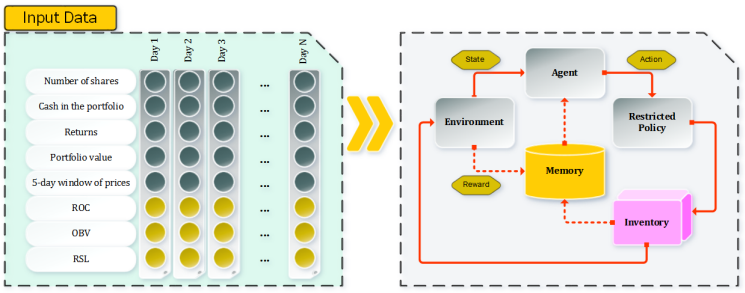

Figure 1: The flowchart of the proposed RL model

The problem of proposing a trading position can be considered a Markov decision process, in which the agent interacts with the environment. At any point in time, the agent uses states to perceive the environment and selects possible actions. The agent is then rewarded for taking that action, and the system transitions to a new state. This process is captured in the trajectory τ, as shown in Eq. (1).

$$\tau = [S0, A0, R1, S1, A1, R2, S2, A2, R3] \tag{1}$$

At any time t, the objective is to maximize the future expected return, which is equivalent to the present value of returns discounted by a factor of gamma, as shown in Eq. (2).

$$G_t = \prod_{k=t+1}^{T} \gamma^{k-t-1} R_K \tag{2}$$

3.1 Environment states

The state of the environment is shaped by information about the investment environment, which is presented as features. The appropriate definition of these characteristics and states is crucial for the actor to understand the environment and learn from it, which, in turn, affects the overall performance of the model. The following states are used in this algorithm:

- state [0] = Number of shares

- state [1] = Cash in the portfolio
- state [2] = Returns
- state [3] = Portfolio value
- state [4] = Five-day window of prices
- state [5] = ROC
- state [6] = OBV
- state [7] = RSI

States 0 and 1 represent the share of stocks and cash in the investment portfolio at each stage. An initial portfolio is set, consisting of a specific number of shares and an amount of cash, which is used by default to start the algorithm. This portfolio changes over time based on the stock price changes and the proposed strategy of the model. State 2 calculates the return from price changes using the following equation:

$$Return = ln(\frac{ClosePrice_{t+1}}{ClosePrice_t}) \tag{3}$$

State [3] represents the value of the investment portfolio, which is calculated at any given time according to Eq. (4). The transaction cost rate (TCR) affects the agent's understanding of the environment. Another assumption in the proposed trading system is that all assets will be converted into cash at the end of the selected period.

$$portfoliovalue_t = state\,[0] \times ClosePrice_t \times (1 - tc) + state[1] \tag{4}$$

To look at recent price changes, we use a 5-day price window as an environmental specification called State 4. In most past research, the state of the environment was defined as only the price at the time of buying and selling, but with this change, we aim to avoid making decisions based on abnormal daily fluctuations and increase the stability of the model.

We include three well-known technical analysis indicators as characteristics of the environment in the model:

(i) Relative Strength Index (RSI)

(ii) Price Rate of Change Indicator (ROC)

(iii) On-Balance-Volume (OBV)

By doing so, we aim to take advantage of technical analysis in developing our RL-based model. Two factors have influenced our choice of these indicators: a) their use in similar studies such as [36], [37], and b) the information that each of these indicators can provide to the operator. Although both RSI and ROC indicators are classified as momentum indicators, they provide different types of information.

The ROC indicator calculates the rate of return between two closing prices, with the initial price typically chosen as the price 14 days before, using Eq. (5) [38]. This indicator attempts to provide the delayed return of past actions as an input to the model. Instead of comparing only the two prices at the beginning and end of the period, the RSI focuses on changes. If gains significantly exceed losses during the period, the stock is overbought, and if losses significantly exceed profit, then the stock is oversold. Generally, changes in the price trend in the range of 30% to 70% are expected, as shown in Eq. (6) [39].

The OBV indicator is a volume-based index. As shown in Eq. (7), Three rules are implemented when calculating the OBV, [40].

$$ROC = \left(\frac{ClosingPrice_p - ClosingPrice_{p-n}}{ClosingPrice_{p-n}}\right) \times 100 \tag{5}$$

$$RSI_{stepone} = 100 - \left[\frac{100}{1 + \frac{Averagegain}{Averageloss}}\right] \tag{6}$$

$$OBV = OBV_{prev} + \begin{cases} volume \quad if close > close_{prev} \\ 0 \quad if close = close_{prev} \\ -volume \quad if close < close_{prev} \end{cases} \tag{7}$$

It's important to note that in reinforcement learning, the model is not given any analysis or interpretation of the indicators mentioned by the user. Only raw data and indicators are provided to the algorithm.

3.2 Actions/model output

In this subsection, we introduce the three actions that the operator can choose in the model:

- Action 0 = Buying the shares with the cash available in the portfolio
- Action 1 = Selling all shares in the portfolio
- Action 2 = Do nothing

The liquidity required to buy shares and the amount received from selling shares is calculated based on the transaction cost, and the number of shares in the investment portfolio is updated accordingly. To encourage the model to buy and sell without penalties for taking action and to avoid impossible situations, a negative reward has been defined. The pseudo-code for defining the actions and updating states is presented below.

Define initial parameters: No action penalty, Negative reward
if action [0] == 2: # **Do Nothing** update the state by new prices return the state and no action penalty
if action [0] == 0: # **Buy Action** if price > cash on hand: # **Not Enough Cash** update the state by new prices return the state and negative reward else: # **Buy** calculate the number of shares that can be purchased as follows: item purchased = cash/(price∗(1+transaction_cost))) updating number of shares in portfolio: shares = shares on hand + item purchased calculating the cash spent to buy shares: cash spent = item purchased∗ price ∗ (1+transaction_cost) update the state by new prices and portfolio structure calculating changes in portfolio value as gain return the state and gain
if action [0] == 1: #**Sell Action** if shares on hand = 0 # **Not Enough share** update the state by new prices return the state and negative reward else: #**Sell** calculate shares that can be sold: item sold= shares on hand update the shares on hand: shares = 0 calculate the cash earned by selling shares: cash_earned = item sold ∗ price∗ (1-transaction_cost) update the state by new prices and portfolio structure calculating changes in portfolio value as gain return the state and gain

Algorithm Pseudo-Code for Defining the Actions

3.3 Reward function

Modern portfolio theory suggests that investors should behave in a way that maximizes the expected utility of wealth Eq. (8) during a limited period. If we assume the investor is risk-neutral, the utility function will be linear. Maximizing this function is equivalent to maximizing the return earned by the selected action Eq. (9).

$$\mathbb{E}\left[U\left(W_T\right)\right] = \mathbb{E}\left[U\left(W_0 + \sum_{t=1}^{T} \delta W_t\right)\right] \tag{8}$$

$$E\left[U\left(W_T\right)\right] = \mathbb{E}\left(\sum_{t=1}^{T} \delta W_t\right) \tag{9}$$

To simplify the model and avoid complexity in the reward function, we have decided to use the simplest form of the reward function, which is equal to the change in portfolio value Eq. (10):

$$Reward = \frac{portfoliovalue_t - portfoliovalue_{t-1}}{portfoliovalue_{t-1}} \tag{10}$$

As mentioned earlier, we aim to maximize future and delayed rewards by selecting an action. To do this, we define an action-value function of $Q\left(S_t.a_t\right)$ that indicates the value of the action (a_t) taken in the current situation (S_t). As shown in Eq. [40], the value of the actions performed so far is equal to the value of the action taken up to the previous state $Q\left(S_{t-1}.a_{t-1}\right)$ plus the reward received in the current state $R\left(S_t\right)$, plus the maximum future value earned by a decision converted to the current value using the reduction factor γ. These factors affect the value of the value based on the learning rate α. The change to the model during each step of the optimal sets of the weight search process, or the step size, is called the learning rate.

$$Q\left(S_t.a_t\right) \leftarrow Q\left(S_{t-1}.a_{t-1}\right) + \alpha \times \left[R\left(S_t\right) + \gamma \times maxQ\left(S_t.a_t\right) - Q\left(S_{t-1}.a_{t-1}\right)\right] \tag{11}$$

Assuming that the function parameters of Q are specified as θ, we aim to optimize the mean square error of real and expected Q to get the optimal state and action mode Eq. [25].

$$L\left(\theta\right) = \mathbb{E}\left[\left(Q_\theta\left(S, A\right) - Q_\theta^{'}\left(S, A\right)\right)^2\right] \tag{12}$$

Our objective will be to minimize $L\left(\theta\right)$. We use the Adam optimization method to optimize the objective function, which is one of the reduction gradient methods that calculates the learning rate according to the data, and this adapted learning rate usually shows the best convergence.

3.4 Restricted Policy

In pure RL, the strategy is determined based on the probability of each (state, action). However, in financial markets, simple RL is not efficient enough to solve such a complex problem. To improve the performance of the system, we have added some limiting boundaries to simple RL. These boundaries help the model avoid bankruptcy (resulting from buying more shares than available cash) or short selling (selling more shares than the number of shares in the portfolio). We named the RL with this policy the "limited RL strategy." The following equations define the boundaries:

- if action == 0 and open price > env.state [21]: action = 2
- if action == 1 and env.state [0] < 1: action = 2

3.5 Neural Network Structure

Neural networks are a common technique used to solve machine learning problems. In this paper, we use a deep neural network to solve the RL model learning algorithm. A neural network with at least two layers is known as a deep neural network (DNN) in its simplest form.

The layers of a neural network of the proposed algorithm are defined as follows:

- Input layer: The number of nodes in the input layer of our model includes 8 nodes, which are defined according to the number of environment states.
- Hidden layers: The structure of this model has 2 fully connected layers of size 256, which are flowed by a GRU block of size 128. Then 3 fully connected layers 128, and 64 are defined in our model structure.
- Output layer: In the structure of the proposed neural network, there are 4 nodes in the output layer, which show the first 3 outputs of the proposed action of the model and the last output of the expected value.

4 Numerical Results

In this paper, we tried to obtain an acceptable performance from the model by changing the components of the trading algorithm, such as input data, defining environmental conditions, applying changes in the model's policy, and how to define actions and reward them. In order to test the modelling method, the data from Tehran Stock Exchange indices were used and the results were discussed and analysed. In the following, we will describe the results in detail.

4.1 Data and preparation

To ensure the accuracy of the data and to avoid potential issues such as dividends, trading halts, stock splits, and other special events, we used three indexes of the Tehran stock market for the purpose of training and testing.

(i) Overall Index (3436 data points - (2008/12-2023/03))

(ii) Top 30 Index (3030 data points - (2010/08-2023/03))

(iii) Financial Index (3436 data point - (2008/12-2023/03))

To obtain the historical data related to the opening and closing prices and the volume for the entire available period of each index, we used the "TseClient" software.

We also used Python programming to calculate technical indicators such as RSI, ROC, and OBV for the data, which were then added to the input data.

4.2 Implementation of the model

We used a large dataset to train and evaluate our model across varying conditions, volatility levels, and average returns over time intervals. To achieve this, we split the dataset for each index into 500-point folds starting within 50-point steps. Each fold was divided into training (400 data points) and test (100 data points) parts.

4.3 Validity of the Proposed Model

To evaluate the performance of our proposed model, we compared it with several conventional strategies such as the buy and hold strategy, randomly selecting actions without entering the transaction, and keeping the initial portfolio. Since the model may not perform best in all trends and price fluctuations, we evaluated its performance using the following indicators:

- Sharpe ratio: The Sharpe ratio divides a portfolio's excess returns by a measure of its volatility to assess risk-adjusted performance. (Risk-free rate of return considered 0)
- Percentage of Profitable trades (PP): The percentage of profitable trades to all trades made by the system.
- Profit Factor (PF): The ratio of cumulative/gross profit to cumulative/gross loss
- Maximum Drawdown (MDD): Measures the maximum loss generated by trade.

We evaluated the model under conditions with and without transaction costs.

Case 1: No Transaction Fee

In the case of having no transaction costs in the environment, we conducted rigorous testing of the strategies outlined in the previous sections over 45 different periods for each of the three data series: Overall Index, Top 30 Index, and Financial Index. The aim was to thoroughly evaluate the performance of each strategy under various market conditions and over an extended time horizon.

The tested strategies are briefly described below:

- Buy and Hold (BAH): This classic strategy involves purchasing assets and holding them for an extended period without making any further transactions. It is considered a passive investment approach, where investors maintain their initial portfolio allocation throughout the testing period.

- No Transaction (No-T): In this approach, we refrained from entering into any transactions and simply kept the initial portfolio allocation unchanged throughout the testing period. This strategy is designed to assess the performance of a static investment approach without any adjustments.
- Random Selection (Rand): To evaluate the effectiveness of our proposed strategy in comparison to random decision-making, we randomly selected actions for each period. This serves as a baseline to gauge the impact of employing a more systematic and data-driven approach.
- Reinforcement Learning (RL): Our presented model incorporates reinforcement learning techniques to determine the optimal actions to take at each time step. By learning from historical data and considering the market dynamics, this strategy is expected to adapt and improve its decision-making over time.

Table 1 summarizes the results of these strategies across the three data series, indicating the returns obtained from each approach. It provides a comprehensive view of how each strategy performed in terms of generating returns over the testing periods.

The results show that the Reinforcement Learning (RL) strategy outperformed the other approaches, including the Buy and Hold (BAH) and Random Selection (Rand) strategies, across all three data series. This suggests that the RL strategy was successful in adapting to changing market conditions and identifying more favourable investment decisions.

While the Buy and Hold (BAH) strategy also showed positive returns, it may not be the most optimal approach, especially when market conditions fluctuate. The No Transaction (No-T) strategy resulted in returns similar to the Buy and Hold strategy, as both strategies maintained static portfolios. However, the No-T strategy lacks the adaptability and learning capability that the RL strategy offers.

Overall, the results demonstrate the potential of employing reinforcement learning techniques in financial trading systems, specifically in the Tehran Securities Exchange Market. The ability of the RL strategy to dynamically adjust its actions based on market trends and historical data showcases the value of data-driven decision-making in enhancing investment performance.

In Table 2, the average performance of each strategy has been measured according to performance measurements.

Based on the results, it can be stated that the presented algorithm has achieved better efficiency in all performance indicators for all three investigated data series. Although the average return obtained by the proposed model is close to the buy-and-hold strategy, a better risk-adjusted relative return has been obtained according to the Sharpe ratio.

For more clarification, examples of proposed RL-based strategies can be found in Figs. 2a to 2i, where buy and sell orders are displayed in red and green dots, respectively. The x denotes a decision to do nothing at that point. We chose three different trends, namely increasing, decreasing, and high fluctuations, to illustrate

Table 1: Results for Case 1

	Overall Index				**Top 30 Index**				**Financial Index**			
	BAH	No-T	Rand	RL	BAH	No-T	Rand	RL	BAH	No-T	Rand	RL
1	0.251	0.039	0.015)	0.259	0.267	0.004	0.090	0.020	0.129	0.044	0.124	0.165
2	0.329	0.052	0.209	0.336	0.407	0.006	0.210	0.269	0.193	0.068	0.166	0.243
3	0.107	0.020	(0.014)	0.130	0.249	0.004	(0.034)	0.261	0.176	0.066	0.091	0.229
4	0.031	0.006	(0.038)	0.046	0.273	0.006	(0.040)	(0.025)	0.071	0.028	0.054	0.095
5	0.026	0.005	0.044	0.047	0.438	0.010	0.078	0.230	(0.055)	(0.023)	0.006	0.011
6	0.018	0.004	0.014	0.103	0.532	0.014	0.251	0.259	(0.140)	(0.057)	(0.033)	0.035
7	0.010	0.002	0.020	0.043	0.189	0.006	0.136	0.272	(0.058)	(0.023)	(0.002)	0.088
8	(0.027)	(0.006)	(0.009)	0.051	(0.091)	(0.004)	(0.120)	(0.052)	(0.031)	(0.012)	0.005	0.033
9	0.195	0.040	0.097	0.168	(0.083)	(0.003)	(0.031)	(0.018)	(0.058)	(0.022)	(0.030)	0.015
10	0.420	0.086	0.187	0.471	(0.112)	(0.004)	(0.075)	(0.043)	(0.003)	(0.001)	(0.043)	0.003
11	0.496	0.133	0.101	0.299	(0.131)	(0.004)	(0.102)	0.014	0.654	0.240	0.442	0.447
12	0.458	0.136	0.118	0.499	(0.070)	(0.002)	(0.060)	(0.025)	0.825	0.325	0.446	0.829
13	0.536	0.189	0.248	0.357	0.020	0.001	(0.076)	0.062	0.636	0.311	0.478	0.657
14	0.304	0.116	0.014	0.070	(0.061)	(0.002)	(0.111)	0.046	0.271	0.147	0.119	0.208
15	(0.072)	(0.033)	(0.082)	0.049	(0.111)	(0.003)	(0.059)	(0.038)	(0.055)	(0.034)	(0.015)	0.034
16	(0.081)	(0.036)	(0.029)	(0.038)	0.239	0.006	0.123	0.227	(0.056)	(0.034)	0.043	0.002
17	(0.078)	(0.034)	(0.005)	0.004	0.201	0.005	(0.059)	0.227	(0.135)	(0.081)	(0.097)	(0.061)
18	(0.059)	(0.025)	(0.093)	0.066	(0.051)	(0.002)	(0.065)	(0.031)	(0.062)	(0.036)	(0.062)	(0.028)
19	(0.107)	(0.045)	(0.021)	0.039	0.040	0.001	0.040	0.071	(0.031)	(0.017)	(0.004)	0.117
20	(0.083)	(0.034)	(0.101)	0.002	0.015	0.000	0.027	0.028	0.027	0.015	0.071	0.090
21	0.039	0.015	(0.011)	0.041	(0.045)	(0.001)	(0.023)	(0.002)	0.160	0.089	0.056	0.191
22	(0.027)	(0.011)	0.005	0.075	(0.061)	(0.002)	(0.040)	(0.022)	(0.057)	(0.033)	(0.048)	0.010
23	(0.075)	(0.030)	(0.043)	(0.016)	0.094	0.003	0.017	0.098	(0.121)	(0.071)	(0.077)	(0.074)
24	0.262	0.100	0.127	0.204	0.198	0.006	0.058	0.200	0.233	0.131	0.116	0.236
25	0.248	0.094	0.137	0.214	0.275	0.009	0.085	0.118	0.210	0.117	(0.034)	0.073
26	(0.000)	(0.000)	(0.037)	0.025	0.146	0.005	0.060	0.148	(0.008)	(0.005)	0.036	0.035
27	0.028	0.012	0.032	0.030	0.258	0.011	0.125	0.261	0.005	0.003	0.010	0.025
28	0.005	0.002	0.025	0.037	0.828	0.033	0.059	0.268	(0.055)	(0.034)	(0.050)	(0.007)
29	(0.016)	(0.007)	(0.018)	0.019	0.523	0.027	0.301	0.611	(0.143)	(0.087)	(0.064)	(0.035)
30	0.010	0.005	0.007	0.014	(0.013)	(0.001)	0.126	0.154	(0.113)	(0.067)	(0.019)	0.004
31	0.069	0.030	0.044	0.051	0.268	0.020	0.083	0.300	(0.015)	(0.008)	(0.003)	(0.002)
32	0.105	0.047	0.066	0.071	0.489	0.034	0.056	0.077	(0.019)	(0.011)	(0.013)	0.011
33	0.197	0.089	0.125	0.136	0.306	0.029	0.113	0.309	0.019	0.011	(0.027)	0.021
34	0.088	0.041	0.070	0.100	0.316	0.032	0.071	0.368	(0.103)	(0.058)	(0.025)	(0.020)
35	0.108	0.054	0.031	0.120	0.581	0.070	0.271	0.467	(0.088)	(0.050)	(0.110)	(0.015)
36	0.521	0.273	0.097	0.568	0.714	0.599	0.488	0.646	0.492	0.269	0.053	0.497
37	0.022	0.013	0.047	0.156	0.825	0.238	0.145	1.031	0.297	0.178	0.162	0.310
38	0.285	0.178	0.295	0.347	(0.263)	(0.128)	(0.470)	0.045	0.347	0.223	0.071	0.432
39	0.601	0.368	0.149	0.493	(0.134)	(0.057)	(0.174)	(0.036)	0.610	0.402	0.209	0.690
40	0.354	0.254	0.109	0.361	0.320	0.125	0.192	0.552	0.339	0.256	0.041	0.345
41	0.629	0.479	(0.005)	0.366	0.281	0.107	(0.076)	0.596	0.502	0.399	(0.061)	0.526
42	0.569	0.518	0.361	0.739	0.214	0.089	0.075	0.221	0.376	0.353	0.350	0.568
43	(0.254)	(0.242)	(0.139)	(0.049)	(0.082)	(0.038)	(0.042)	(0.031)	(0.179)	(0.172)	(0.150)	(0.078)
44	(0.026)	(0.025)	0.046	(0.025)	(0.071)	(0.033)	(0.033)	(0.032)	(0.206)	(0.197)	(0.159)	(0.004)
45	(0.236)	(0.221)	(0.153)	0.031	0.154	0.068	0.134	0.179	(0.205)	(0.196)	(0.126)	(0.027)

Table 2: Performance of each Strategy Case-1

No Transaction Cost		**Mean**	**Variance**	**Sharp**	**PP**	**PF**	**MDD**
Overall	Buy And Hold	0.1373	0.05	2.66	0.69	6.39	(0.25)
	Do Nothing	0.0589	0.02	2.95	0.69	4.54	(0.24)
	Rand	0.0447	0.01	4.10	0.62	3.48	(0.15)
	RL	0.1580	0.03	4.74	0.91	56.81	(0.05)
Top 30	Buy And Hold	0.1840	0.07	2.68	0.67	7.00	(0.26)
	Do Nothing	0.0285	0.01	2.84	0.67	5.53	(0.13)
	Rand	0.0383	0.02	1.76	0.58	2.02	(0.47)
	RL	0.1841	0.05	3.56	0.73	24.46	(0.05)
Financial	Buy And Hold	0.1017	0.06	1.57	0.47	3.29	(0.21)
	Do Nothing	0.0522	0.02	2.29	0.47	2.77	(0.20)
	Rand	0.0423	0.02	2.00	0.49	2.52	(0.16)
	RL	0.1538	0.05	2.98	0.76	20.59	(0.08)

how the model decides in each data series.

Case 2: With Transaction Fee

In Case 2, we conducted a similar study to the previous case study, with the only difference being the introduction of a transaction fee of 0.1% per transaction amount. This transaction fee is a crucial factor to consider, as it reflects the real-world costs associated with executing trades in financial markets. The inclusion of transaction costs adds a layer of complexity to the evaluation of the strategies, as it directly impacts the overall profitability of each approach.

To assess the performance of the strategies under the new environment condition, we once again tested the 3 selected indices - Overall Index, Top 30 Index, and Financial Index - over various periods. The results of the proposed strategies in the presence of transaction costs are presented in Table 3. As seen in Table 3, the impact of the transaction fee is evident in the performance of the strategies. The Reinforcement Learning (RL) strategy, which outperformed the other approaches in Case 1, is still expected to demonstrate its adaptability and learning capabilities. However, the introduction of transaction costs may affect its overall profitability, given that each trade now incurs a small fee.

The Buy and Hold (BAH) strategy, which performed reasonably well in Case 1, may also be affected by transaction costs. The impact of the fees on the BAH strategy's returns depends on the trading frequency and the holding period of the assets.

The No Transaction (No-T) strategy, by design, does not incur any transaction fees, as it maintains the initial portfolio allocation without making any changes. Consequently, the returns of this strategy are not directly affected by transaction costs. However, the strategy's performance may still be impacted by market move-

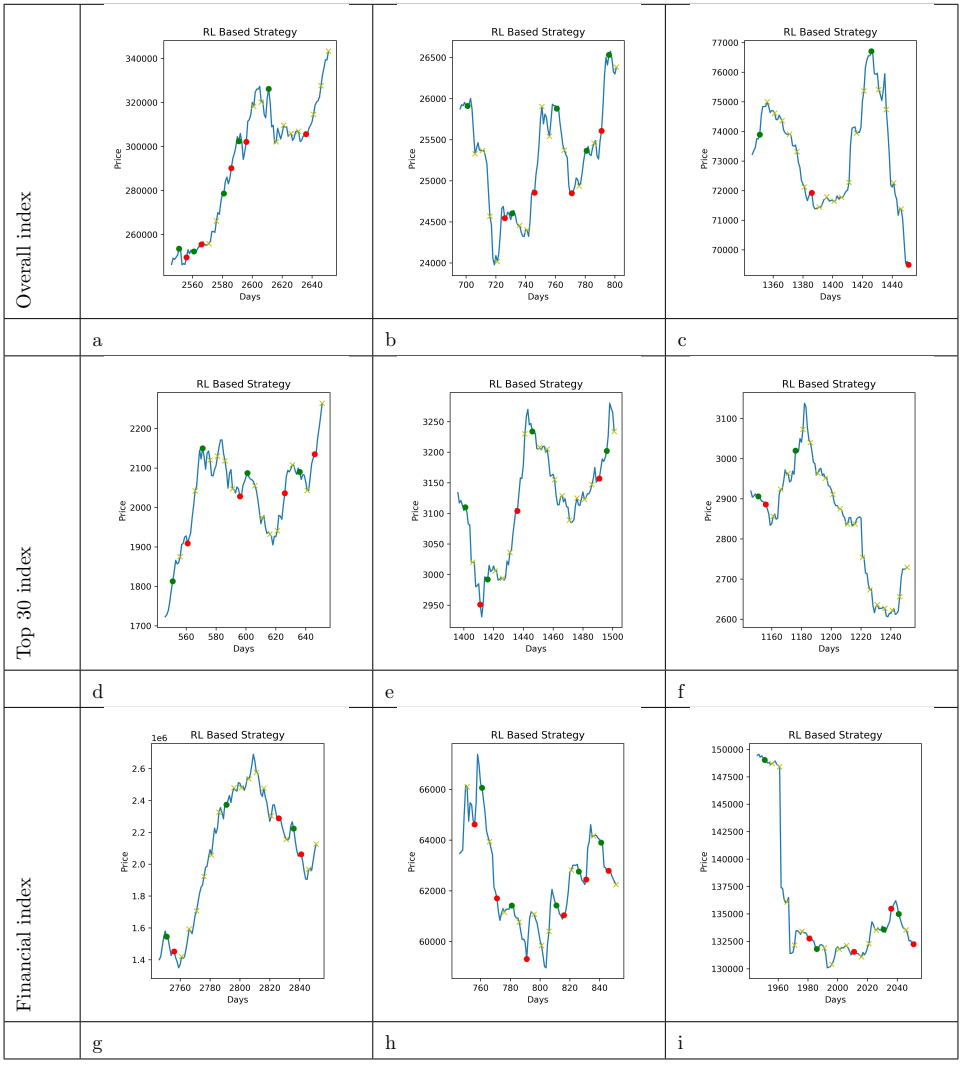

FIGURE 2. Examples of Purposed RL-Based Strategies

ments and the chosen portfolio allocation.

The Random Selection (Rand) strategy, as a baseline approach, is also subject to transaction costs for each randomly selected action. The returns of this strategy are likely to vary widely across different periods due to its inherent randomness.

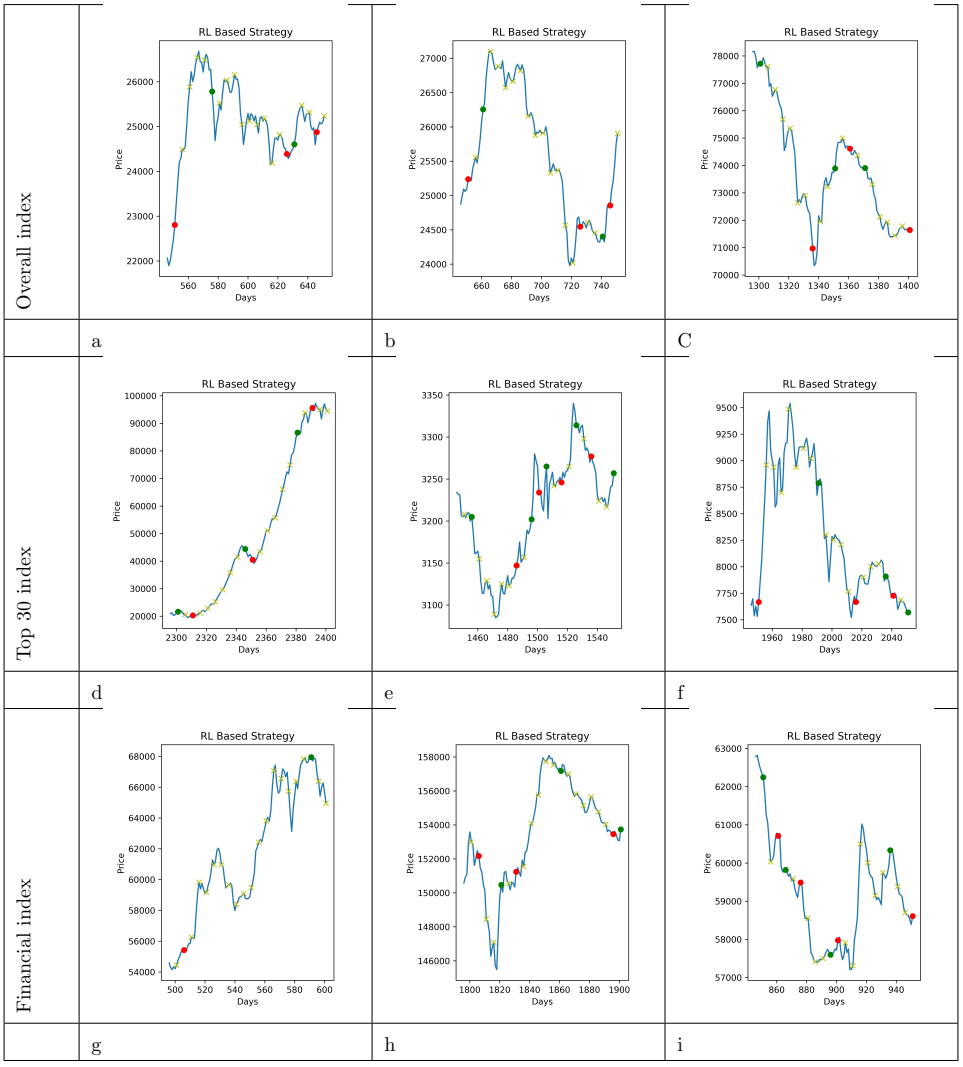

FIGURE 3. Examples of Purposed RL-Based Strategies Considering TC

5 Conclusion

Considering the transaction cost, a reduction in returns obtained compared to the previous case would be normal. However, as in the previous case, the training of

Table 3: Results for Case 2

	Overall Index				**Top 30 Index**				**Financial Index**			
	BAH	No-T	Rand	RL	BAH	No-T	Rand	RL	BAH	No-T	Rand	RL
1	0.249	0.038	0.208	0.171	0.264	0.004	0.008	0.275	0.127	0.044	(0.032)	0.139
2	0.327	0.052	0.140	0.150	0.404	0.006	0.099	0.438	0.191	0.068	0.020	0.232
3	0.105	0.020	0.080	0.146	0.246	0.004	0.103	0.328	0.174	0.065	0.088	0.224
4	0.029	0.006	0.019	0.058	0.270	0.006	0.130	0.304	0.070	0.028	(0.016)	0.092
5	0.025	0.005	0.010	0.069	0.435	0.010	0.335	0.528	(0.057)	(0.023)	(0.044)	(0.050)
6	0.016	0.004	(0.021)	0.034	0.529	0.014	0.197	0.279	(0.141)	(0.058)	(0.123)	(0.135)
7	0.008	0.002	0.026	0.059	0.187	0.006	0.002	0.042	(0.060)	(0.024)	(0.081)	(0.032)
8	(0.029)	(0.006)	0.015	0.011	(0.093)	(0.004)	(0.040)	0.008	(0.033)	(0.012)	(0.062)	0.043
9	0.193	0.040	0.111	0.199	(0.085)	(0.003)	(0.068)	0.006	(0.060)	(0.023)	(0.028)	(0.013)
10	0.418	0.086	0.160	0.326	(0.113)	(0.004)	(0.054)	0.028	(0.005)	(0.002)	(0.039)	0.000
11	0.494	0.132	0.127	0.347	(0.133)	(0.004)	(0.082)	(0.041)	0.652	0.239	0.286	0.458
12	0.455	0.136	0.233	0.289	(0.072)	(0.002)	0.056	(0.041)	0.822	0.324	(0.003)	0.284
13	0.534	0.189	0.179	0.311	0.018	0.001	0.070	0.151	0.634	0.310	0.048	0.574
14	0.301	0.115	0.073	0.124	(0.063)	(0.002)	0.016	0.034	0.269	0.146	0.152	0.292
15	(0.074)	(0.033)	(0.062)	(0.070)	(0.112)	(0.003)	(0.046)	(0.057)	(0.056)	(0.034)	(0.002)	0.025
16	(0.083)	(0.037)	(0.044)	(0.057)	0.236	0.006	0.160	0.237	(0.058)	(0.034)	(0.129)	(0.035)
17	(0.080)	(0.035)	(0.072)	0.026	0.199	0.005	0.170	0.218	(0.136)	(0.081)	(0.108)	(0.056)
18	(0.061)	(0.026)	(0.041)	(0.006)	(0.053)	(0.002)	0.000	0.060	(0.063)	(0.037)	(0.066)	(0.045)
19	(0.109)	(0.045)	(0.091)	(0.044)	0.038	0.001	0.026	0.049	(0.032)	(0.018)	0.092	0.069
20	(0.085)	(0.035)	(0.103)	(0.022)	0.013	0.000	(0.005)	0.034	0.025	0.015	0.099	0.100
21	0.037	0.015	(0.057)	0.116	(0.047)	(0.001)	(0.072)	(0.017)	0.158	0.088	0.134	0.220
22	(0.029)	(0.011)	(0.011)	(0.012)	(0.063)	(0.002)	(0.006)	(0.043)	(0.058)	(0.033)	(0.103)	0.074
23	(0.076)	(0.030)	(0.010)	(0.005)	0.092	0.003	0.024	0.098	(0.122)	(0.072)	(0.071)	(0.085)
24	0.260	0.100	0.111	0.101	0.195	0.006	0.094	0.108	0.232	0.131	0.049	0.308
25	0.246	0.094	0.125	0.251	0.273	0.009	0.062	0.077	0.208	0.117	0.177	0.230
26	(0.002)	(0.001)	0.029	0.025	0.143	0.005	0.002	0.089	(0.010)	(0.006)	(0.058)	(0.005)
27	0.026	0.012	(0.049)	0.039	0.256	0.011	0.014	0.293	0.003	0.002	(0.026)	0.021
28	0.004	0.002	(0.011)	0.018	0.825	0.033	0.246	0.545	(0.056)	(0.034)	(0.023)	(0.041)
29	(0.018)	(0.008)	(0.005)	(0.012)	0.520	0.027	0.136	0.345	(0.144)	(0.087)	(0.048)	(0.010)
30	0.009	0.004	(0.007)	0.034	(0.015)	(0.001)	0.064	0.151	(0.114)	(0.068)	(0.055)	0.010
31	0.067	0.030	0.046	0.043	0.265	0.020	0.150	0.272	(0.016)	(0.009)	(0.015)	(0.008)
32	0.104	0.046	0.019	0.059	0.486	0.034	0.205	0.497	(0.021)	(0.011)	(0.019)	(0.019)
33	0.195	0.089	0.065	0.110	0.303	0.029	0.057	0.308	0.018	0.010	(0.024)	0.034
34	0.086	0.041	0.106	0.087	0.314	0.032	0.068	0.205	(0.104)	(0.058)	0.001	(0.066)
35	0.106	0.053	(0.070)	0.128	0.578	0.070	0.227	0.306	(0.089)	(0.050)	(0.005)	0.061
36	0.519	0.272	0.554	0.598	0.696	0.598	0.508	0.623	0.490	0.268	0.115	0.550
37	0.020	0.012	0.122	0.121	0.822	0.237	0.282	0.269	0.296	0.177	0.128	0.386
38	0.283	0.178	0.022	0.262	(0.264)	(0.128)	(0.204)	(0.138)	0.345	0.222	(0.012)	0.392
39	0.599	0.367	0.165	0.267	(0.136)	(0.057)	(0.230)	(0.001)	0.608	0.401	0.189	0.199
40	0.352	0.253	(0.025)	0.177	0.318	0.124	0.055	0.460	0.337	0.255	0.006	0.354
41	0.627	0.478	0.153	0.319	0.279	0.106	0.280	0.295	0.500	0.398	0.329	0.517
42	0.568	0.517	0.455	0.723	0.212	0.089	0.090	0.251	0.375	0.352	0.157	0.611
43	(0.255)	(0.242)	(0.176)	(0.001)	(0.084)	(0.038)	(0.049)	0.027	(0.180)	(0.173)	(0.143)	(0.059)
44	(0.026)	(0.025)	(0.049)	(0.014)	(0.072)	(0.033)	(0.029)	(0.049)	(0.207)	(0.198)	(0.065)	(0.013)
45	(0.237)	(0.222)	(0.164)	(0.099)	0.152	0.067	0.116	0.157	(0.206)	(0.196)	(0.140)	(0.116)

Table 4: Performance of each Strategy Case 2

With Transaction Cost		Mean	Variance	Sharp	PP	PF	MDD
Overall	buy and hold	0.1355	0.05	2.63	0.69	6.24	(0.25)
	do nothing	0.0585	0.02	2.94	0.69	4.49	(0.24)
	rand	0.0508	0.02	2.75	0.58	3.14	(0.18)
	RL	0.1213	0.03	4.52	0.76	17.01	(0.10)
Top 30	buy and hold	0.1815	0.07	2.67	0.67	6.81	(0.26)
	do nothing	0.0284	0.01	2.84	0.67	5.48	(0.13)
	rand	0.0704	0.02	3.90	0.73	4.58	(0.23)
	RL	0.1780	0.03	5.25	0.82	21.73	(0.14)
Financial	buy and hold	0.1001	0.06	1.55	0.47	3.22	(0.21)
	do nothing	0.0515	0.02	2.27	0.47	2.73	(0.20)
	rand	0.0118	0.01	1.03	0.38	1.34	(0.14)
	RL	0.1269	0.04	3.16	0.62	8.25	(0.13)

the model has been done in such a way that the proposed algorithm achieves better results in terms of performance metrics compared to competing strategies. The results can be seen in Table 4.

As we present in case 1, a few examples of purposed RL-based strategies considering transaction cost are demonstrated in Figs. 3a to 3i.

In the past decade, automatic trading systems have become prevalent in international markets, and Iran's financial markets are moving in the same direction. The development of infrastructure and laws related to this development indicates a process of change in Iran's financial markets. Therefore, the establishment of an algorithmic trading system based on artificial intelligence and its effective subset, reinforcement learning, can be a small but effective step towards localizing global knowledge.

This study focuses on items and methods that increase the general efficiency of the model. To increase the efficiency of the model, the following solutions were implemented:

(i) The use of technical analysis as an input provides additional and useful information to the model. However, the interpretation of this information is entirely the responsibility of the operator.

(ii) The actions and behaviors that the operator can choose are defined in a way that is consistent with market conditions. Heavy penalties are given for wrong behaviors to prevent the model from acting on them, and a small penalty is considered for not entering a trade to encourage the model to enter a transaction.

(iii) If the model chooses an impossible action, the model will exit the learning block, and future cycles of the block will not be evaluated. To prevent this

from happening, a constraint was added to the model policy. If a wrong action is taken that causes an exit from the cycle, a heavy penalty is assigned, and that action is changed to the action of not entering the transaction.

The findings indicate that this model is more effective than other methods in both with and without transaction cost scenarios. The proposed model was examined using 45 different periods for each data series, and the results were analyzed with four performance measurements (Sharp, PP, FP, and MDD), indicating that the proposed model is superior to all other competitive strategies. Additionally, decision diagrams presented in different periods illustrate that the algorithm can understand the turning points of the price movement and choose the appropriate action after learning.

Credit authorship contribution statement: Z. Pourahmadi: Conceptualization, Methodology, Software, Validation, Writing original draft. **D. Fareed:** Conceptualization, Supervision, Writing review & editing. **H. R. Mirzaei:** Supervision, Writing review & editing.

Disclosure statement: The authors report there are no competing interests to declare.

Data availability Statement: The data has been generated by the model itself

Bibliography

[1] NECCHI PG, *Reinforcement learning for automated trading*, Math Eng Di Milano Milano, Italy., (2016), pp. 120.

[2] MOLGA M, SMUTNICKI C, *Test functions for optimization needs*, 2007330 Httpwww Zsd Ict Pwr Wroc Plfilesdocsfunctions Pdf,. (2005), pp. 143.

[3] BISHOP CM, NASRABADI NM, *Pattern recognition and machine learning*, 4th ed. New York: springer; (2006).

[4] LI Y, *Deep Reinforcement Learning: An Overview*, arXiv:1810.06339, (2018), p. 1–85.

[5] CHAKOLE J, KURHEKAR M, *Trend following deep QLearning strategy for stock trading*, Expert Syst,. 37 (2020).

[6] NIAKI STA, HOSEINZADE S, *Cvx: Forecasting SP 500 index using artificial neural networks and design of experiments*, J Ind Eng Int,. 9 (2013) pp. 1–9.

[7] SCHWAGER JD, *A complete guide to the futures market: Technical analysis, trading systems, fundamental analysis, options, spreads, and trading principles*, John Wiley Sons., (2017).

[8] CHAN E, trading: how to build your own algorithmic trading business, John Wiley Sons., (2009).

[9] PLOTNIKOV AP, SHISHLOV RA, ARSENOV V V, *An algorithm for organizing long volatility trading based on a delta-neutral strategy*, Vestn SAMARA Univ Econ Manag., 13 (2022), pp. 156–73.

[10] THAKRAR K, *Research Report on Delta Neutral Trading Strategy*, Vidhyayana-An Int Multidiscip Peer-Reviewed E-Journal-ISSN., 6 (2020).

[11] LONGSTAFF FA, SCHWARTZ ES, *Valuing American options by simulation: a simple least-squares approach*, Rev Financ Stud., 14 (2001).

[12] Tsitsiklis JN, Van Roy B, *Regression methods for pricing complex American-style options*, IEEE Trans Neural Networks., 12 (2001), pp. 694–703.

[13] Zhang X, Shen H, Lv Z, *Deployment optimization of multi-stage investment portfolio service and hybrid intelligent algorithm under edge computing*, PLoS One., 16 (2021).

[14] Kim JH, Lee Y, Kim WC, Fabozzi FJ, *Goal-based investing based on multi-stage robust portfolio optimization*, Ann Oper Res., (2022).

[15] Chong E, Han C, Park FC, *Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies*, Expert Syst Appl., 83 (2017), pp. 187–205.

[16] Recht B, *A tour of reinforcement learning: The view from continuous control*, Annu Rev Control Robot Auton Syst., 2 (2019).

[17] Heaton JB, Polson NG, Witte JH, *Deep learning for finance: deep portfolios*, Appl Stoch Model Bus Ind., 33 (2017), pp. 3–12.

[18] Atsalakis GS, Valavanis KP, *Surveying stock market forecasting techniquesPart II: Soft computing methods*, Expert Syst Appl., 36 (2009).

[19] Martinez LC, da Hora DN, Palotti JR de M, Meira W, Pappa GL, *From an artificial neural network to a stock market day-trading system: A case study on the bmf bovespa*, Int. Jt. Conf. Neural Networks., (2009).

[20] Ding X, Zhang Y, Liu T, Duan J, *Deep learning for event-driven stock prediction*, Twenty-fourth Int. Jt. Conf., (2015).

[21] Akita R, Yoshihara A, Matsubara T, Uehara K, *Deep learning for stock prediction using numerical and textual information*, IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci., (2016), pp. 1–6.

[22] Won J, Lee JW, *Stock price prediction using reinforcement learning*, ISIE 2001., (2001), p. 690–5.

[23] Lei K, Zhang B, Li Y, Yang M, Shen Y, *Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading*, Expert Syst Appl., 140 (2020).

[24] Moody J, Wu L, Liao Y, Saffell M, *Performance functions and reinforcement learning for trading systems and portfolios*, J Forecast., 17 (1998), pp. 441–70.

[25] Guéanta O, *Reinforcement Learning Methods in Algorithmic Trading*, Mach Learn Data Sci Financ Mark A Guid to Contemp Pract., (2023).

[26] Moody J, Saffell M, *Learning to trade via direct reinforcement*, IEEE Trans Neural Networks., 12 (2001), pp. 875–889.

[27] Moody J, Saffell M, *Learning to trade via direct reinforcement*, IEEE Trans Neural Networks., 12 (2001), pp. 875–889.

[28] Sutton RS, McAllester DA, Singh SP, Mansour Y, *Policy gradient methods for reinforcement learning with function approximation*, NIPs., (1999), pp. 1057–63.

[29] Gao X, Chan L, *An algorithm for trading and portfolio management using Q-learning and sharpe ratio maximization*, Proc. Int. Conf. neural Inf. Process., Citeseer., (2000), pp. 832–837.

[30] Pendharkar PC, Cusatis P, *Trading financial indices with reinforcement learning agents*, Expert Syst Appl., 103 (2018), pp. 1–13.

[31] Wang Y, Wang D, Zhang S, Feng Y, Li S, Zhou Q, *Deep Q-trading*, Cslt Riit Tsinghua Edu Cn., 12 (2017).

[32] Deng Y, Bao F, Kong Y, Ren Z, Dai Q, Member S, *Deep direct reinforcement learning for financial signal representation and trading*, IEEE Trans Neural Networks Learn Syst., 28 (2016), pp. 653–664.

[33] Tsantekidis A, Passalis N, Toufa A-S, Saitas-Zarkias K, Chairistanidis S, Tefas A, *Price trailing for financial trading using deep reinforcement learning*, IEEE Trans Neural Networks Learn Syst., 32 (2020), pp. 2837–2846.

[34] Yuan Y, Wen W, Yang J, *Using data augmentation based reinforcement learning for daily stock trading*, Electronics., 9 (2020).

[35] Vishal M, Satija Y, Babu BS, *Trading agent for the indian stock market scenario using Actor-Critic based reinforcement learning*, IEEE Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut., (2021), pp. 1–5.

[36] KABBANI T, DUMAN E, *Deep Reinforcement Learning Approach for Trading Automation in the Stock Market*, IEEE Access., 10 (2022), pp. 93564–74.

[37] THÉATE T, ERNST D, *An application of deep reinforcement learning to algorithmic trading*, Expert Syst Appl., 173 (2021).

[38] FAWCETT T, *An introduction to ROC analysis*, Pattern Recognit Lett., 27 (2006), pp. 861–74.

[39] WILDER JW, *New concepts in technical trading systems*, Trend Research., (1978).

[40] GRANVILLE JE, *Granvilles new strategy of daily stock market timing for maximum profit*, Prentice-Hall., (1976).

How to Cite: Zahra Pourahmadi[1], Dariush Farid[2], Hamid Reza Mirzaei[3], *A novel financial trading system based on reinforcement learning and technical analysis applied on the Tehran securities exchange market*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 3, No. 1, Pages:99–118, (2023).

The Journal of Mathematics and Modeling in Finance (JMMF) is licensed under a Creative Commons Attribution NonCommercial 4.0 International License.