Int. J. Nonlinear Anal. Appl. In Press, 1–15
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2025.37619.5455

Design and Implementation of an Automated Trading System Based on Reinforcement Learning for Financial Markets

Erfan Abedini[a], Hadi Mohammadi Mohammadi[b,*], Mohammad Hassan Saadatian Farivar[c]

[a] *Department of Financial Management, Faculty of Management and Economics, SR.C., Islamic Azad University, Tehran, Iran*

[b] *Department of Economics, Faculty of Economics and Accounting, CT.C., Islamic Azad University, Tehran, Iran*

[c] *Department of Financial Management, Accounting and Financial Engineering, Faculty of Management and Economics, SR.C., Islamic Azad University, Tehran, Iran*

(Communicated by Majid Eshaghi Gordji)

Abstract

In recent years, the rapid growth of technology and data-driven financial markets has led to an increased interest in using artificial intelligence, and especially reinforcement learning, in designing automated trading systems. In this study, an automated trading system based on reinforcement learning for the stock market is designed and implemented. The main objective of this study is to investigate the efficiency of reinforcement learning models including Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), and Deep Q-Network (DQN) compared to traditional investment methods. To train the models, real stock market data and technical indicators are used, and two different reward functions are proposed to evaluate the system performance. The performance of the models is analyzed with criteria such as total return and Sharpe ratio. Experimental results show that reinforcement learning models can provide higher returns than traditional methods in certain conditions, and choosing the appropriate reward function plays a significant role in improving trading strategies. This research, while emphasizing the importance of carefully designing the reward function and selecting appropriate features, offers new perspectives for the development of intelligent systems in financial markets.

Keywords: Reinforcement learning, artificial intelligence, financial markets, automated trading system, reward function
2022 MSC: O20, O10

1 Introduction

Financial markets are fundamental structures in any economy that serve as organized spaces for trading various financial assets, including stocks, bonds, commodities, and other assets and securities. This investment environment facilitates the efficient allocation of resources by directing capital from individuals or institutions with excess capital to those in need of investment. Financial arenas not only contribute to economic growth by providing access to capital for businesses and governments, but also help generate wealth for individual investors and provide opportunities

*Corresponding author
Email addresses: erfan.abedini@srbiau.ac.ir (Erfan Abedini), hm_mohamady@iauctb.ac.ir (Hadi Mohammadi Mohammadi), mohammadsaadatianf@gmail.com (Mohammad Hassan Saadatian Farivar)

Received: May 2025 *Accepted:* August 2025

to diversify investment portfolios and increase wealth [9]. Financial markets are generally divided into two main categories, primary markets and secondary markets. In primary markets, financial assets are created and sold for the first time, so that investors directly help finance companies or institutions. This market plays an important role in providing initial capital for projects and newly established companies. On the other hand, secondary markets are places where financial assets created in the primary market are traded. These markets allow for high liquidity of assets and investors can profit from price changes by buying and selling existing assets. In addition, financial markets are divided into categories such as stock market, bond market, commodity market and foreign exchange market based on the type of asset traded. Each of these markets plays a unique role in the economy. This diversity in the structure and functioning of financial markets reflects their high importance in directing capital, managing risk and supporting economic growth. Each type of market, with its unique characteristics and role, contributes to the development and sustainability of the global economy and allows investors to adjust their investment strategies to suit their goals and needs [1]. The importance of the capital market in research and studies related to automated trading systems is also very prominent. Dynamic financial environments such as the stock market, due to their variable and complex characteristics, provide an ideal opportunity to benefit from new approaches to design and implement optimal trading strategies [19]. In the meantime, artificial intelligence-based trading methods can play an important role in optimizing the trading process and increasing investor returns by analyzing the market in real time and processing complex patterns. From this perspective, emerging algorithms, using real-time information and analyzing historical data, help improve the quality of decision-making in the capital market and other financial markets [8].

In the past few years, artificial intelligence has become one of the biggest technological revolutions in human life, affecting almost all aspects of our lives. From everyday applications such as personalized recommendations on social networks and smart assistants, to more complex applications in the medical, engineering, and financial industries, artificial intelligence is increasingly entering our lives [12]. The rapid growth of artificial intelligence algorithms and their ability to process huge data has led to the development of models that are capable of analyzing complex patterns and making optimal decisions in a fraction of the time on many topics [14]. They allow traders to act more confidently and achieve better results in risky and volatile environments. In addition, artificial intelligence allows traders to react quickly in volatile market conditions and manage opportunities and risks optimally. These developments not only reduce risk and increase efficiency, but also provide a new path for exploiting financial data in intelligent decision-making [16]. Machine learning-based algorithms, especially reinforcement learning, have a high ability to identify complex patterns and adapt to dynamic environments [15]. However, little research has been conducted on the use of reinforcement learning in the financial field, and this field has a research gap. Therefore, this research aims to design and implement an automated trading system based on reinforcement learning algorithms, which can be an effective step towards filling this gap. The results of this research can lead to the introduction of new algorithms and methods in the development and implementation of automated trading. In recent decades, financial market analysis has always been based on two main approaches: fundamental and technical analysis. Analysts have used these methods to try to identify the intrinsic value of assets or predict price trends, but the increasing complexity of the market and the huge volume of financial data have transformed the analysis environment. In the fundamental analysis method, the intrinsic value of assets is determined based on the examination of financial and economic variables. By collecting data such as income, profit margins, debt structure and cash flow, as well as macroeconomic indicators such as GDP growth rates and interest rates, analysts use models such as discounted cash flow to calculate the present value of the asset and, if there is a significant difference with the market price, take action to buy or sell [10]. However, collecting and continuously updating this volume of data is difficult, especially for small companies or emerging markets, and the nonlinear and complex relationships between economic variables often require simplifying assumptions that can reduce the accuracy of valuation [18, 17]. Furthermore, the fundamental analysis process is time-consuming and not efficient enough to respond to rapid and instantaneous market changes. On the other hand, technical analysis, focusing on historical price and volume data, is based on the assumption that all the necessary information is reflected in price and volume charts and that past trends can provide clues to future market behavior. Tools such as trend lines, support and resistance levels, classic patterns (head and shoulders, flags, etc.), and technical indicators such as RSI and MACD are used in this method to determine optimal entry and exit points [11]. However, increased market volatility and the impact of unpredictable news and events have made historical patterns sometimes misleading and false signals can lead to significant losses [18]. Common challenges of these two traditional approaches include a high dependence on analyst interpretation and experience, delays in accessing up-to-date data, and an inability to respond to market changes in real time; issues that are exacerbated in algorithmic markets and high-frequency trading [17]. In contrast, automated trading systems based on artificial intelligence and machine learning algorithms—including neural networks, support vector machines, and genetic algorithms—are capable of analyzing large volumes of market data in real time and extracting complex patterns without human intervention [6]. Reinforcement learning methods, in particular, have enabled the design of dynamic and adaptive trading strategies by defining the market environment and training an agent to optimize cumulative returns. Although little research has been conducted on the application of reinforcement learning in financial markets, the present study aims to fill the existing research gap and introduce new algorithms and methods to the scientific community by designing and implementing an automated trading system based on this method.

2 Research background

In recent decades, financial markets have become a popular research topic due to their high complexity and dynamism. With the advancement of technology and the emergence of new artificial intelligence tools, the use of automated trading systems based on reinforcement learning in financial markets has received increasing attention. These systems are able to improve the efficiency and profitability of transactions by analyzing extensive data and making decisions in real time [13]. A scientific paper [7] discusses the application of deep reinforcement learning in capital market transactions. It uses one of the most famous deep reinforcement learning algorithms called DQN, the output of which is three actions: buy, sell, and hold. The selection of the action space is one of the most important parts in reinforcement learning modeling, which affects the final performance of the system. In this research, in addition to those three actions, a deep neural network is also used to predict the trading volume, which complements one of the selected actions. In another scientific study, [20] more than three actions were considered as the output of reinforcement learning algorithms. In addition to the DDPG, PPO, and A2C algorithms, the combined effect of these three algorithms was also used as a separate algorithm to determine the trading policy. Although the PPO model performed best among the algorithms in terms of percentage return, the combined algorithm influenced by the three algorithms recorded a better performance in terms of Sharpe ratio. [21] In a paper titled Deep Reinforcement Learning for Trading, they used DRL algorithms to design trading strategies for continuous futures contracts that consider both discrete and continuous action spaces. Their approach included scaling the volatility in the reward functions, which allows for adjusting trading positions based on market fluctuations. By testing on a diverse range of futures contracts, their algorithms outperformed classical time series momentum strategies even with heavy transaction costs. Their experiments demonstrated the ability of the proposed algorithms to follow large market trends without frequent position changes, and to adapt to periods of consolidation. [5] In a paper titled Financial Trading as a Game: A Deep Reinforcement Learning Approach, they employed the Markov Decision Process (MDP) model and the Deep Recurrent Q-Network (DRQN) algorithm with modifications to increase its suitability for financial markets. They significantly utilized small replay memory and introduced action reinforcement techniques to improve the exploration efficiency. In addition, they sampled longer sequences for training the recurrent neural network, which significantly reduced the training time. Validation of this approach in the foreign exchange market showed strong experimental performance compared to traditional exploration methods. In recent years, researchers have investigated deep reinforcement learning techniques to solve the challenges of algorithmic trading in a scientific manner. For example, a study used a fuzzy deep recurrent neural network structure to design a trading system without the need for technical indicators and using fuzzy learning to reduce uncertainty in time series [4]. Another study also investigated the application of the DQN algorithm to trading in foreign exchange markets [20]. Another paper [2] investigated the application of deep reinforcement learning (DRL) in high-frequency trading. The authors propose a DRL framework that includes a deep Q-network to optimize trading strategies in a high-frequency trading environment. This study demonstrates the effectiveness of DRL in high-frequency trading and provides better performance compared to traditional trading strategies. The authors highlight the potential of DRL to adapt to changing market conditions and optimize trading decisions in real time.

3 Research methodology

3.1 Dataset

In this study, historical data related to Apple Inc. stocks with the symbol AAPL have been selected as the main dataset for training and evaluating the reinforcement learning model. The selection of this stock is fully justified from both a scientific and practical perspective for several reasons, including its high trading volume, high liquidity, appropriate price fluctuations, and the company's important position in the technology industry. Apple's stock is one of the most traded assets in international financial markets, and features such as price dynamics and susceptibility to news and macroeconomic factors have made it an ideal option for evaluating the performance of reinforcement learning algorithms. In addition, analyzing this stock can make the research results more reliable for generalization to other stocks active in the technology industry.

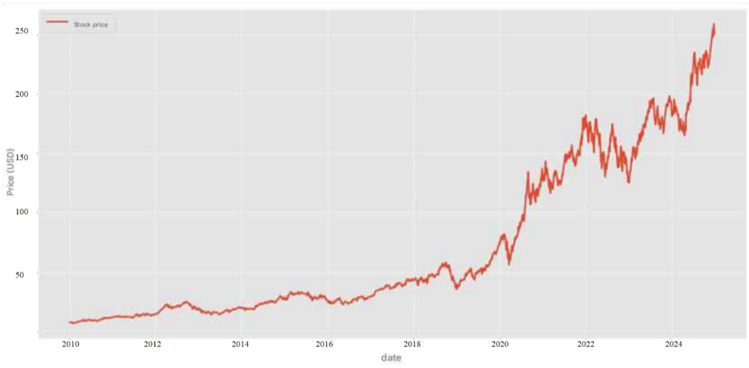

Figure 1: Apple stock price chart

The Yahoo Finance API and the open-source yfinance library were used to collect the relevant data. Developed in the Python programming language, this library provides easy, free, and standardized access to historical stock data, financial indicators, and other market information. The data received includes opening, high, low, and closing (OHLC) prices, trading volume, and fundamental data. The choice of this tool, due to its high compatibility with other scientific Python libraries such as Pandas, NumPy, and Matplotlib, facilitated the process of data analysis, visualization, and preparation of data for model training. The raw data obtained goes through various stages of processing and cleaning before being used in the reinforcement learning model. In the first step, the data is examined and corrected for missing data, time inconsistencies, and possible errors. Then, in order to normalize the data and reduce the effect of different scales of variables, methods such as MinMaxScaler or standardizing the data within a certain range have been used to optimize the data for machine learning and reinforcement algorithms. In addition, to extract technical features and technical indicators, the TA-Lib library has been used, which allows the calculation of indicators such as moving averages, relative strength index (RSI), oscillators and other technical analysis tools. These indicators have been added as inputs to reinforcement learning models to increase the accuracy of predictions.

3.2 Data preprocessing

To effectively use raw data in reinforcement learning models, it is necessary for this data to undergo careful preprocessing steps before entering the training process. The first step in this process is data cleaning; in this step, missing data is identified or removed, and if possible, it is replaced with appropriate values. Incorrect or inconsistent data is also corrected to ensure the completeness and temporal consistency of the dataset. The next step is to normalize or standardize the data scale. The goal of this is to prevent the model from biasing towards features with larger numerical scales and to ensure that the variables are aligned during learning. In this study, the MinMaxScaling method was used to standardize the numerical data. This method maps the data to a specific interval, which specifically prevents problems such as algorithm divergence during training and leads to greater stability of the model. Another important step in preprocessing is feature engineering. For example, the values of the original unstandardized data are as follows:

Table 1: Key values in the dataset

Date	Open	High	Low	Close	Volume
2010-01-04	7.622500	7.660714	7.585000	7.643214	493729600
2010-01-04	7.622500	7.660714	7.585000	7.643214	493729600
2010-01-04	7.622500	7.660714	7.585000	7.643214	493729600

After scaling, the data is spaced as follows:

Table 2: Scaled values in the dataset

Date	Open	High	Low	Close	Volume
2010-01-04	0.014273	0.013735	0.014556	0.014476	0.253588
2010-01-04	0.014273	0.013735	0.014556	0.014476	0.253588
2010-01-04	0.014273	0.013735	0.014556	0.014476	0.253588

Also, in this stage, new and important features are created using the initial data. Given the financial nature of the problem, these features mainly include technical analysis indicators that provide useful information about the trend, momentum, and price behavior of the stock. Indicators such as moving averages, RSI, MACD, and various oscillators are extracted at this stage and added to the dataset so that the model can make more accurate decisions by better understanding the market structure. At the end of the preprocessing process, the final dataset, including historical price data (open, high, low, close) along with trading volume and extracted technical indicators, is stored in a structured database. This storage structure allows for fast and secure access to the data during the training, testing, and evaluation stages of the model. Through this process, the raw data has been transformed into a standardized and optimized set for use in reinforcement learning models that are highly accurate, consistent, and efficient.

3.3 Technical indicators

Technical indicators are tools that help traders make optimal decisions by providing information about overbought and oversold conditions, possible reversal points, and the general direction of price movement. In this study, indicators such as the Relative Strength Index (RSI), Exponential Moving Average (EMA), Williams' Index (W%R), and Money Flow Index (MFI) have been used to cover various aspects of technical analysis and extract more accurate signals.

3.4 Settings and training strategies for reinforcement learning models

In designing and implementing a reinforcement learning-based trading system, choosing the right settings for the data and the training process plays a fundamental role in the final performance of the model. This section examines the structure of the data, how to segment it, use a sliding window, and determine the number of training cycles. One of the key steps in training reinforcement learning models is data separation. In this study, Apple stock price data, including features such as opening and closing prices, highs and lows, trading volumes, and technical indicators, were pre-processed and divided into two parts: training (80%) and testing (20%). The training data included 3019 samples from January 4, 2010 to December 29, 2021, and the testing data included 755 samples from December 30, 2021 to December 31, 2024. This separation was performed in a time series to prevent information leakage and to measure the performance of the model in real market conditions. Data trend analysis and autocorrelation were also performed to increase the accuracy of the model's generalizability.

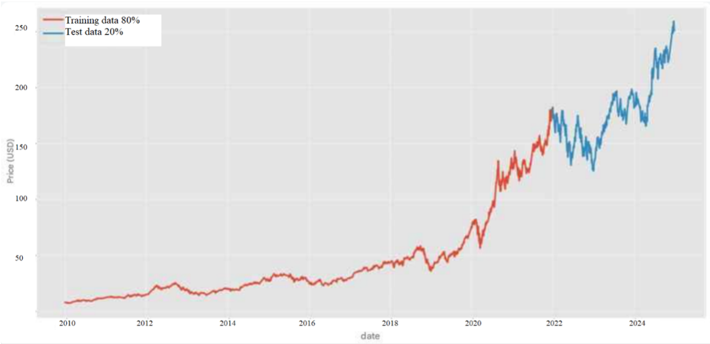

Figure 2: Training and testing data partitioning

On the other hand, using the sliding window technique allows the model to use the latest data in trading decisions. In this study, two sizes of 5 and 20 periods are considered for the sliding window. The short-term window of 5 periods allows the model to react to rapid changes and perform well in volatile market conditions. In contrast, the 20-period window, by considering medium-term trends, prevents hasty reactions and increases the stability of decision-making. The simultaneous use of these two windows improves the flexibility and generalizability of the model in the face of various market conditions. Also, choosing the appropriate number of training periods is one of the effective factors in training reinforcement learning models. In this study, two values of 1000 and 2000 have been selected for the training periods. A smaller value reduces the training time and prevents overfitting, while a larger value allows the model to learn market patterns more deeply and develop more accurate strategies. These two values were chosen with the aim of balancing the learning speed and the quality of the model output.

The combination of these three components, including the temporal structure of the data, the use of a sliding window, and the number of training cycles, provides a comprehensive framework for effective training of the PPO, A2C, and DQN models and provides a suitable platform for evaluating the performance of these models in real market conditions.

3.5 Structure of the trading robot

In this research, the reinforcement learning system is considered as the core of the trading robot design, a system that consists of various components, each of which plays a vital role in achieving optimal decisions for operating in the financial markets. These components are designed in such a way that they can simulate market behavior and enable accurate and effective trading decisions.

One of the most important components of this system is the learning environment, which represents the real financial market. This environment includes historical data such as open, close, high, low prices, trading volume, and technical indicators, all collected from reliable sources. The environment also reflects key market characteristics such as volatility, uptrends, and downtrends, and real trading conditions. The agent, or trading robot, learns how to modify and optimize its decisions in response to changes and different market conditions by continuously interacting with this environment.

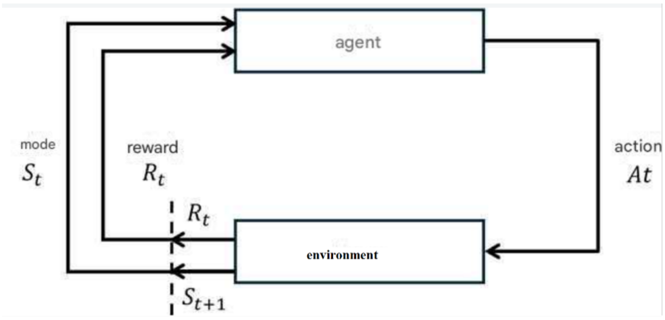

Figure 3: Interaction between agent and environment in reinforcement learning model

The learning agent is a central part of the learning and decision-making process. By receiving inputs from the environment and using a reinforcement learning policy, this agent takes actions such as buying, selling, or holding stocks. The main goal of the agent is to find the best possible strategy to maximize profit and reduce risk. In this research, the agent is trained using advanced reinforcement learning algorithms such as PPO, A2C, and DQN. A state or mode is a set of information that describes the current market conditions and the agent's position. This information includes market prices, technical indicators, and the status of the agent's investment portfolio. Properly designing the state provides the agent with a comprehensive and accurate view of the market and allows for better analysis of price trends and future predictions. The actions that the agent takes at each stage include buying, selling, or holding assets. These actions are selected based on the current market situation and the learning policy, and their goal is to improve the investment portfolio composition to achieve the highest possible return. The accurate and timely selection of these actions has a direct impact on the final performance of the system. Reward plays a key role in guiding the learning process and is considered as a criterion for measuring the quality of the agent's decisions. In this research, reward is defined based on the returns from transactions as well as the costs incurred. Reward design is done in such a way as to reinforce profitable behaviors and reduce behaviors that lead to losses or increased risk. Finally, policy is the framework based on which the agent decides what action to take in each situation. This policy can be defined as deterministic or probabilistic. In this research, advanced algorithms such as PPO and A2C are used to optimize the agent's decision-making policy to establish a proper balance between exploration (learning from new situations) and exploitation (using past experiences). This coherent structure has provided a solid foundation for designing a trading robot that has the ability to make optimal decisions in complex financial markets.

3.6 Reinforcement learning algorithms used

In this research, three different reinforcement learning algorithms have been used to design and implement a trading robot in financial markets: PPO, A2C, and DQN. Each of these algorithms has its own characteristics and advantages, which will be discussed below. These algorithms were selected because of their ability to handle dynamic conditions in financial markets and adapt to complex data.

- PPO: is a policy-based algorithm that performs direct policy optimization and is suitable for maintaining a balance between exploration and exploitation. It has been selected in this research because of its ability to handle high market fluctuations.

- A2C: is an actor-critic model that accelerates learning by using the concept of advantage and helps analyze changing market conditions. This algorithm is suitable for faster improvement of trading strategies.
- DQN: It is a value-based algorithm that uses deep neural networks to approximate Q values. This algorithm performs better in environments where the data is relatively more stable and predictable.

The use of these three algorithms is due to their different characteristics, which can identify the strengths and weaknesses of each and provide the best approach for designing a trading robot in financial markets.

3.7 Evaluation criteria

In order to systematically analyze the performance of trading robots based on reinforcement learning, it is essential to use accurate quantitative criteria. These criteria not only allow us to measure the effectiveness of the proposed models, but also play a key role in answering the research questions and confirming or rejecting the research hypotheses. In this study, five main criteria including total period return, model trading return, Sharpe ratio, maximum drawdown and trading success rate are considered to evaluate the models. The investment return is divided into two parts: the total period return and the return from the purchase and sale of the model. The total period return measures the changes in the value of the asset from the beginning to the end of the test period and provides a criterion for comparing the model's performance with the overall market trend. The return from the model's transactions indicates the profitability of the trading strategy and the algorithm's ability to take advantage of market opportunities. The Sharpe ratio is used as one of the common indicators to evaluate the performance of investment strategies. This ratio measures the excess return over the risk-free rate per unit of risk and provides a criterion for measuring the efficiency of returns over return fluctuations. The maximum capital loss, as an indicator of downside risk, determines the greatest decrease in the portfolio value during the trading period and provides valuable information about the stability of the model in volatile market conditions. In addition, the transaction success rate, which shows the percentage of profitable transactions compared to the total transactions, is used to evaluate the accuracy of the model in identifying profitable opportunities. The simultaneous use of these metrics allows for multidimensional evaluation and comprehensive comparison of the performance of different algorithms and creates a valid framework for qualitative and quantitative analysis of reinforcement learning-based trading robots.

4 Results

4.1 Performance of the A2C model

The results obtained from the implementation of the A2C model under different conditions indicat the variable performance of this algorithm in comparison with the market return. The market return during the period under review was 40.5%. The A2C model has been able to generate a positive return in some conditions, while in other conditions it has performed worse than the market. **In its best case**, the A2C model, with a sliding window of 20

Table 3: Results of the A2C Model

Number of training courses	1000		2000	
Sliding window	5	20	5	20
Market return	% 40.5	% 40.5	% 40.5	% 40.5
Model return	% -11.9	% -28.9	% 5.1	% 8.7
Number of trades	368	229	330	178
Sharp ratio	-0.33	-1.05	0.01	-0.03
Sortino ratio	-0.05	-1.39	0.01	-0.05
Maximum loss	%-38.1	%-38.4	% -24.7	%-22.1

and 2,000 training episodes, achieved a return of 7.8%. In the worst case, with the same window but 1,000 training episodes, it recorded a return of -28.9%. These results indicate that the model's performance is highly dependent on the number of training episodes. The number of trades varied between 178 and 368, and even in the scenario with the highest return, both the Sharpe and Sortino ratios were negative, indicating weak profitability accompanied by high risk. Moreover, the model faced significant losses across all configurations, with the maximum drawdown reaching -38.4%, suggesting that the A2C model does not perform well in risk control and capital preservation in volatile markets.

4.2 DQN model performance

The DQN model results demonstrate the different performance of this model under four different conditions, with training episodes of 1,000 and 2,000 and sliding windows of 5 and 20 periods. The results show significant variations in the model's performance in response to market volatility and parameter changes. The DQN model outperformed both the market and the A2C model in all configurations. Its best performance was recorded with a 38% return under the setting of 2,000 training episodes and a sliding window of 20. The positive Sharpe and Sortino ratios in all scenarios indicate a good balance between risk and return, as well as the model's high efficiency in managing market volatility. The highest Sharpe ratio was 0.39 and the highest Sortino ratio was 0.65. The maximum drawdown in the best scenario was only -15.1%, which reflects effective risk management. Additionally, the number of trades varied from 40 to 490, depending on the model configuration, which demonstrates the model's flexibility and adaptability to different market conditions.

Table 4: DQN model results

Number of training courses	1000		2000	
Sliding window	5	20	5	20
Market return	% 40.5	% 40.5	% 40.5	% 40.5
Model return	% 30.7	% 24.9	% 30.4	% 38
Number of trades	490	350	40	56
Sharp ratio	0.33	0.31	0.31	0.39
Sortino ratio	0.54	0.48	0.53	0.65
Maximum loss	% -25	%-15.1	% -37.2	% -32.9

4.3 PPO model performance

The PPO model, across all configurations with the same parameter values, has shown notable results. This analysis more closely examines the model's return, risk ratios, number of trades, and maximum drawdown, evaluating its strengths and weaknesses under different conditions. The PPO model outperformed the market and the other models (A2C and DQN) in all configurations. In the setting with 2,000 training episodes and a sliding window of 20, it achieved an impressive return of 84.3%. The average number of trades was around 360, indicating consistency and stability in decision-making. A Sharpe ratio of 0.91 and a Sortino ratio of 1.55 in the best scenario highlight the model's excellent performance in risk management and return generation. The maximum drawdown ranged from -16.4% to -28.9%, reflecting the model's balance between risk tolerance and return.

Table 5: PPO model results

Number of training courses	1000		2000	
Sliding window	5	20	5	20
Market return	% 55.7	% 77.8	% 66.5	% 84.3
Model return	% 30.7	% 24.9	% 30.4	% 38
Number of trades	361	367	360	368
Sharp ratio	0.64	0.82	0.72	0.91
Sortino ratio	1.09	1.41	1.18	1.55
Maximum loss	% -26.3	%-16.4	% -17.3	% -28.9

4.4 Simulation results: Evaluating the impact of combining technical ondicators on reinforcement learning system performance

One of the key factors in improving the performance of reinforcement learning models in financial trading is the effective use of technical indicators alongside raw price data. These indicators provide detailed information on market trends, volatility, and buy-sell pressure, and can enhance the model's decision-making process. The analysis of results indicates that combining technical indicators with raw data has a positive impact on return, risk management, and performance metrics such as the Sharpe and Sortino ratios.

In the evaluation of the PPO, A2C, and DQN models, configurations with 2,000 training episodes and a sliding window of 20 consistently delivered the best performance across all models. These settings led to higher returns, reduced drawdowns, and improved performance metrics. Therefore, the same configuration will be used as the baseline in the next phase of the research to more accurately and comparably assess the impact of reward functions and strategy optimization. Under optimal conditions, the PPO model performed exceptionally well, recording a return of 99.3%,

Table 6: Comparison of results of A2C, DQN, and PPO models

Model	**A2C**	**DQN**	**PPO**
Number of training courses	2000	2000	2000
Sliding window	20	20	20
Market return	% 40.5	% 40.5	% 40.5
Model return	% 61.4	% 77	% 99.3
Number of trades	331	58	345
Sharp ratio	0.82	0.71	1.04
Sortino ratio	1.34	1.21	1.77
Maximum loss	% -14.2	% -25	% -26.3

which is significantly higher than the market return. With 345 executed trades, the model demonstrated a high level of trading activity. Its Sharpe and Sortino ratios (1.04 and 1.77, respectively) indicate a well-balanced trade-off between risk and return. However, the maximum drawdown of -26.3% reflects the model's sensitivity to sharp market fluctuations during certain periods.

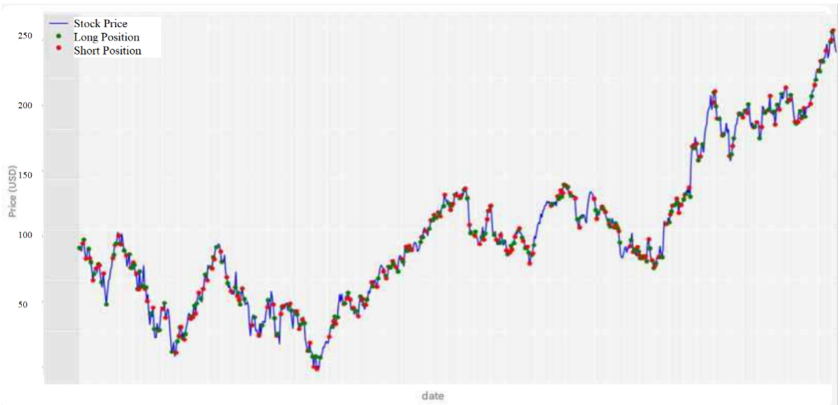

Figure 4: PPO Model Performance with Added Technical Indicators

In this experiment, the DQN model recorded a lower return of 77% and adopted a more conservative approach with only 58 trades. Its Sharpe ratio (0.71) and Sortino ratio (1.21) were also lower than those of the PPO model, indicating a weaker balance between return and risk. Although its maximum drawdown of -25% was slightly less than that of PPO, the overall performance of DQN in terms of return and trading efficiency is considered inferior.

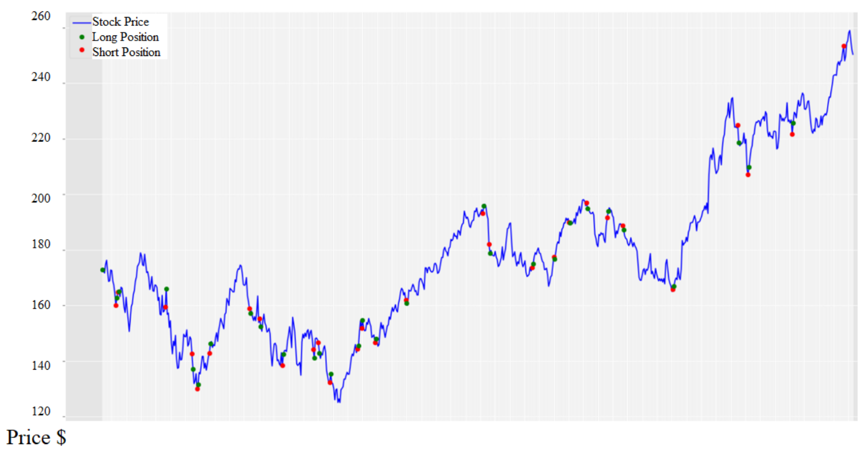

Figure 5: DQN Model Performance with Added Technical Indicators

In this experiment, the A2C model recorded a return of 61.4%, which is lower than that of the PPO and DQN models, but demonstrated high trading activity with 331 trades. Its Sharpe ratio (0.82) and Sortino ratio (1.34) indicate a moderate balance between return and risk. In terms of risk management, A2C achieved the best performance among the three models with a maximum drawdown of -14.2%, showing greater resilience to severe market downturns.

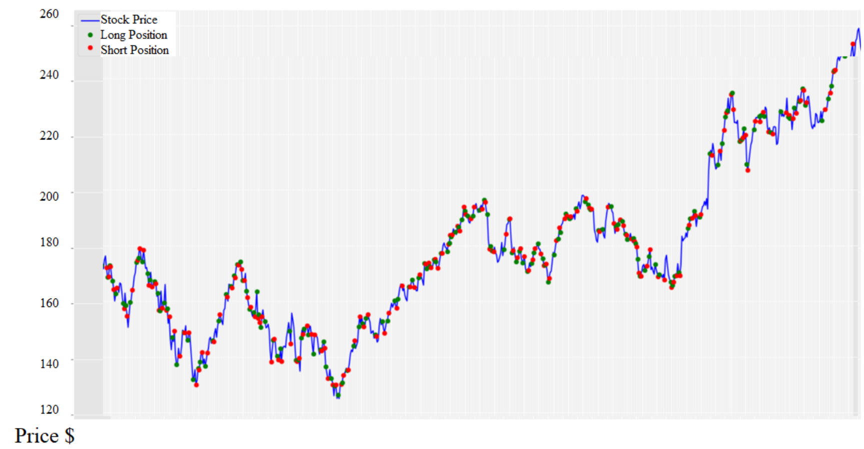

Figure 6: A2C Model Performance with Added Technical Indicators

4.5 The impact of reward function design on enhancing trading strategies in reinforcement learning

This section examines the effect of reward function design on the performance of reinforcement learning models in financial trading. Two reward functions with different approaches have been analyzed: one focusing on the Sharpe ratio to optimize the balance between return and risk, and the other aiming to minimize maximum drawdown.

The use of a Sharpe ratio-based reward function, which evaluates excess return relative to the risk-free rate against portfolio volatility, encourages the model to lean toward strategies that not only yield higher returns but also bear lower risk. In this approach, the reward is calculated based on a combination of instantaneous return and Sharpe ratio, helping the model make more precise decisions in changing market conditions.

This analysis was conducted using the best model configurations to accurately evaluate the true impact of the reward functions on key metrics such as return, risk, number of trades, and the Sharpe and Sortino ratios.

Table 7: Comparison of A2C, DQN, and PPO Models Results

Model	**A2C**	**DQN**	**PPO**
Number of training courses	2000	2000	2000
Sliding window	20	20	20
Market return	% 40.5	% 40.5	% 40.5
Model return	% 75.1	% 74.1	% 72.4
Number of trades	367	398	357
Sharp ratio	0.87	0.86	0.76
Sortino ratio	1.50	1.46	1.24
Maximum loss	% -18.9	% -21.5	% -32.9

The results indicate that all three models performed better than the market return (5.40%). Among them, the A2C model ranked first with a return of 1.75%, followed by DQN with 1.74% and PPO with 4.72%. These results highlight the positive impact of incorporating risk-based metrics, such as the Sharpe ratio, in optimizing trading strategies.

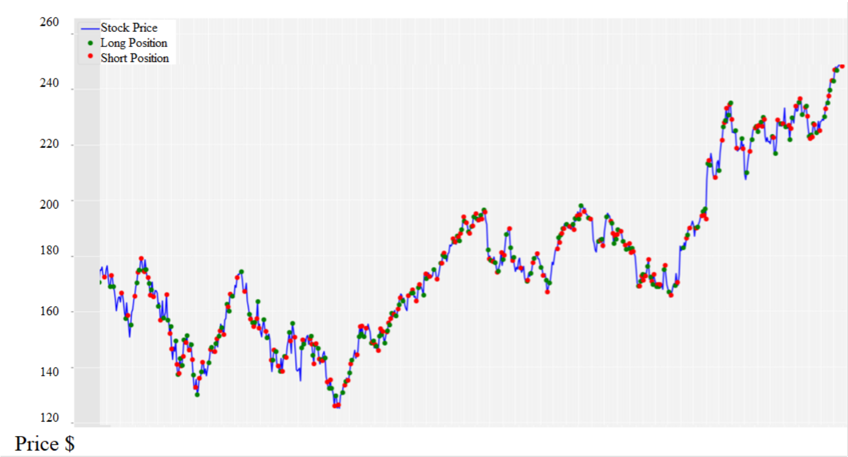

Figure 7: Performance of the A2C model using the reward function adjusted based on risk and return

The DQN model, with 398 trades recorded, exhibited the highest trading activity, whereas the A2C and PPO models executed 367 and 357 trades, respectively. This difference may stem from the higher sensitivity of DQN to market fluctuations and its greater risk tolerance. However, a larger number of trades does not necessarily indicate better performance and may lead to increased transaction costs.

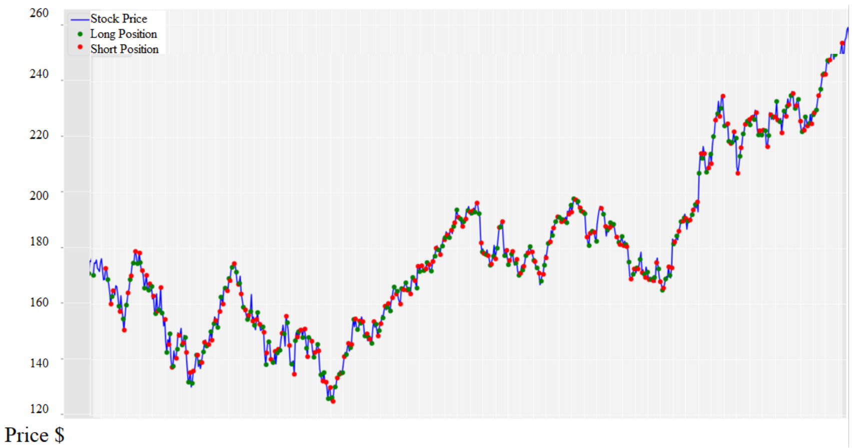

Figure 8: Performance of the DQN model using the reward function adjusted based on risk and return

For a more precise evaluation of the models' performance, the Sharpe ratio and Sortino ratio were used. The A2C model, with a Sharpe ratio of 0.87 and a Sortino ratio of 1.50, delivered the best performance in terms of risk-adjusted returns. These results indicate that A2C not only generated higher returns relative to overall volatility but also effectively managed downside volatility. The DQN and PPO models, with lower values in these indices, ranked next.

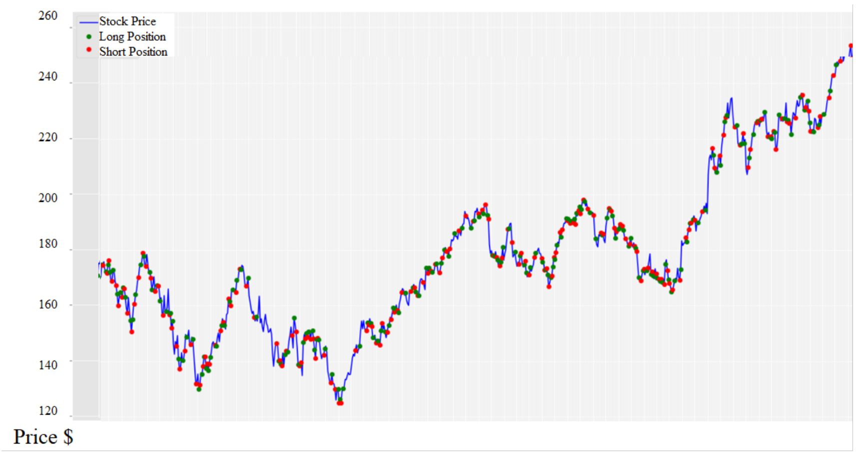

Figure 9: Performance of the PPO model using the reward function adjusted based on risk and return

In the evaluation of maximum drawdown, the PPO model recorded the largest drop at -32.9%, indicating the highest risk. In contrast, the A2C model demonstrated better loss control with the smallest drawdown of -18.9%, while DQN, with -21.5%, ranked in between. These results suggest that A2C, in addition to delivering favorable returns, was more successful in mitigating severe losses.

In the next stage, a reward function based on maximum drawdown control was designed for risk management and to prevent sharp declines in portfolio value within reinforcement learning trading systems. This function helps the model adopt strategies that avoid large drawdowns and reduce undesirable volatility. In this approach, historical trading data within a specified time window (252 trading days) is used to assess drawdown. Then, the instantaneous reward is calculated as the change in portfolio value. Subsequently, the drawdown relative to the previous peak portfolio value is computed, and if it exceeds the allowable limit, the model is penalized. This penalty discourages the model from entering high-risk trades that could result in significant capital losses. Finally, the final reward is calculated by subtracting the penalty from the instantaneous reward. This method helps models simultaneously focus on maximizing returns and managing risk, thereby creating more stable strategies with reduced volatility. In many trading systems, the maximum drawdown metric is considered a key indicator for performance evaluation.

Table 8: Comparison of A2C, DQN, and PPO Models Results

Model	**A2C**	**DQN**	**PPO**
Number of training courses	2000	2000	2000
Sliding window	20	20	20
Market return	% 40.5	% 40.5	% 40.5
Model return	% 77.4	% 70.9	% 83.6
Number of trades	18	172	394
Sharp ratio	0.70	0.66	0.93
Sortino ratio	1.19	1.11	1.66
Maximum loss	% -28	% -29.9	% -13.5

The results show that all three models outperformed the market return, indicating the success of the reward function in improving trading outcomes. The PPO model ranked first with a return of 83.6%, while A2C and DQN recorded returns of 77.4% and 70.9%, respectively. These findings suggest that implementing maximum drawdown control not only did not reduce returns but, in some cases, even led to a significant increase.

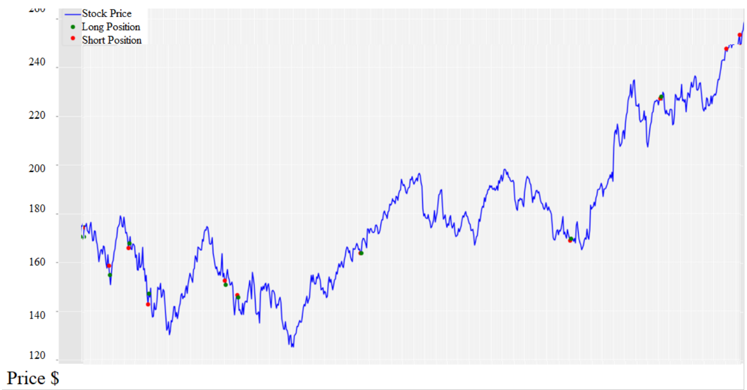

Figure 10: Performance of the A2C model using the reward function based on maximum drawdown control

The number of trades executed in this experiment reveals significant differences in the trading behavior of the models. The PPO model, with 394 trades, exhibited the highest activity in response to market fluctuations, while DQN, with 172 trades, adopted a more cautious approach. Meanwhile, the A2C model executed only 18 trades, indicating a highly selective and conservative strategy aimed at reducing maximum drawdown.

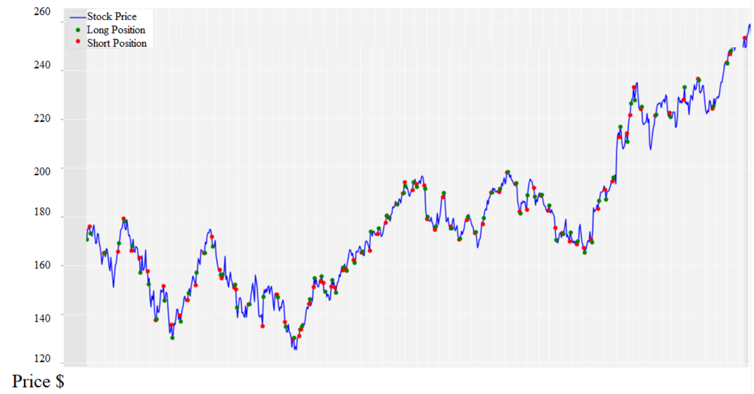

Figure 11: Performance of the DQN model using the reward function based on maximum drawdown control

The analysis of the Sharpe and Sortino ratios shows that models with higher returns also performed better on these risk-adjusted metrics. The PPO model, with a Sharpe ratio of 0.93 and a Sortino ratio of 1.66, ranks first, indicating its ability to generate excess returns relative to overall volatility and its stronger focus on positive volatility. These results demonstrate the model's high efficiency in risk management and in capitalizing on profitable market opportunities.

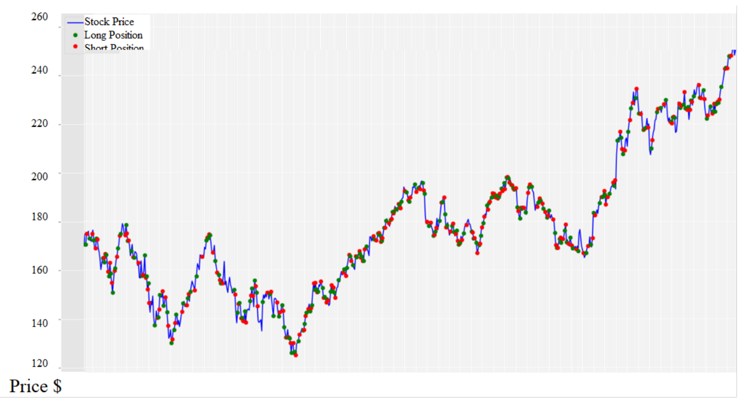

Figure 12: Performance of the PPO model using the reward function based on maximum drawdown control

The A2C model ranked second with a Sharpe ratio of 0.70 and a Sortino ratio of 1.19, indicating its decent returns along with higher volatility compared to the PPO model. The DQN model showed the weakest performance, with a Sharpe ratio of 0.66 and a Sortino ratio of 1.11, likely due to its lower number of trades and reduced portfolio volatility. An important point in this section is the maximum drawdown metric, which is directly influenced by the design of the reward function. The PPO model achieved the best risk control, recording the smallest drawdown at -13.5%. In contrast, A2C and DQN experienced the largest losses, with drawdowns of -28% and -29.9%, respectively. This difference demonstrates that the PPO model was able to maintain higher returns while reducing risk despite its high trading activity, whereas the other models, despite fewer trades, faced more severe capital losses.

5 Conclusion and recommendations

In today's financial world, markets pose many challenges for investors due to rapid fluctuations and numerous uncertainties. Therefore, automated trading systems capable of quickly adapting to market changes are highly important. These systems, by leveraging machine learning algorithms, have significantly improved trading performance and achieved higher returns compared to traditional methods. Reinforcement learning, through the interaction of an intelligent agent with the environment and learning from rewards and penalties, designs models that make optimal decisions in complex and dynamic conditions. This method is particularly useful for trading systems because it offers features such as self-learning, adaptability to market changes, and the ability to manage uncertainty. The present study focused on the design and implementation of an automated trading system based on reinforcement learning, utilizing various reinforcement learning algorithms, including PPO, DQN, and A2C, to enhance trading performance. In this study, the performance of these algorithms was evaluated from different perspectives. In the first stage, the models were run with a basic reward function without using technical indicators. The results showed that some reinforcement learning models could achieve higher returns than passive market strategies. The PPO model delivered the best performance in terms of return and risk-return ratios, while the DQN model achieved positive returns but, due to its more conservative approach, recorded lower returns than the PPO model. The A2C model, however, showed weaker performance and recorded negative returns.

In the second stage, technical indicators were added to the input data to provide the learning agents with more comprehensive information on price trends and market behavior. These changes led to reduced volatility and increased stability in the models' returns. Notably, the A2C model, after adding technical indicators, achieved better risk-return ratios and its performance improved significantly. These changes highlight the positive impact of incorporating technical data into the model's learning process and enhancing trading decisions. In the third stage, the models were optimized using advanced reward functions. Two types of reward functions were designed: one to optimize returns concerning risk, and the other to control drawdown. These functions helped the models not only strive to maximize returns but also manage risks more effectively. The PPO model, using these reward functions, continued to deliver the best performance in terms of return and risk management. The DQN model, with its more conservative approach, was able to significantly reduce drawdown, and finally, the A2C model proved more successful in controlling severe portfolio fluctuations. The findings of this study indicate that the performance of reinforcement learning algorithms is influenced by the precise design of reward functions and data inputs. In particular, the PPO model, with its ability to maximize returns and deliver favorable risk ratios, showed superior overall performance. On the other hand, the DQN and A2C models, with their distinct approaches, are suitable options for scenarios with lower risk tolerance. These analyses also demonstrate that each model can be customized based on the investor's risk tolerance level, and by adjusting parameters and reward functions, trading strategies aligned with different investors' goals can be developed. For example, investors seeking high returns can use models focused more on maximizing profits, while those with lower risk tolerance can choose models that excel at controlling volatility and reducing drawdown. These results also provide a basis for developing hybrid strategies that combine the advantages of each model to meet the diverse needs of investors in various market conditions. In this way, the flexibility of reinforcement learning models to adapt to different objectives can improve trading performance and facilitate future research into intelligent trading tools. Finally, this study emphasizes the importance of model selection and precise parameter tuning to develop optimal and sustainable trading strategies in financial markets, which can significantly enhance investor performance and reduce market risks.

Future research recommendations include expanding input data, developing reinforcement learning models for the Iranian market, and optimizing trading strategies for different types of securities. Additionally, adapting reinforcement learning models for better risk management and improving investment efficiency has also been highlighted. These recommendations can help improve the accuracy and efficiency of intelligent trading systems and pave the way for the development of advanced trading strategies.

References

[1] A. Aziz, *How to Day Trade for a Living*, Bear Bull Traders, Vancouver, Canada, 2016.

[2] A. Briola and et al., *Deep reinforcement learning for active high frequency trading*, arXiv preprint arXiv: 2101.07107 (2021).

[3] J. Carapuço, R. Neves and N. Horta, *Reinforcement learning applied to Forex trading*, Applied Soft Computing **73** (2018), no. 6, 783-794.

[4] Y. Deng and et al., *Deep direct reinforcement learning for financial signal representation and trading*, IEEE transactions on neural networks and learning systems **28** (2016), no. 3, 653-664.

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

[6] C. F. Huang and H. C. Li, *An Evolutionary Method for Financial Forecasting in Microscopic High-Speed Trading Environment*, Computational Intelligence and Neuroscience **10** (2017), no. 1, 9580815.

[7] G. Jeong and H. Y. Kim, *Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning*, Expert Systems with Applications **117** (2019), no. 8, 125-138.

[8] A. Kaur, N. Gupta, *Revolutionizing stock market: Role of AI and its impact on stockholders*, **10** (2025), no. 2, 845-848.

[9] I. K. Kavonius and V. M. Tormalehto, *Is the Financial Market Driving Income Distribution?*, An analysis of the linkage between income and wealth in Europe, 2022.

[10] A. Kumanireng, *Application of Fundamental Analysis in Stock Valuation in the Capital Market and Investment Decisions by Price Methods Earnings Ratio (PER)*, Available at SSRN, 2023.

[11] S. K. Inani and et al., *Navigating the technical analysis in stock markets: Insights from bibliometric and topic modeling approaches*, Investment Management and Financial Inno.vations **21** (2024), no. 1, 275-288.

[12] R. Islam, D. Chakraborty, A. Kumar Roy, M. M. Rahman and M. M. Rahman Enam, *Effectiveness of AI-based bachine learning algorithms in predicting global market movements*, Journal of Engineering Research and Reports **26** (2024), no. 8, 343-354.

[13] B. Itri and et al., *Deep reinforcement learning strategy in automated trading systems*, in 2023 3rd International Conference on Inno.vative Research in Applied Science, Engineering and Techno.logy (IRASET) (2023), IEEE.

[14] E. Molina, P. Cortes, I. Molina and F. Sobrino, *The Impact of Artificial Intelligent Tools on Decision Making Behavioral and Neural Dynamics*, Research Square, 2024, https://doi.org/10.21203/rs.3.rs-4651166/v1.

[15] S. Otabek and J. Choi, *Multi-level deep Q-networks for Bitcoin trading strategies* Scientific Reports **14** (2024), no. 1, 771-790.

[16] J. D. D. Roger, *AI-driven financial modeling techniques: Transforming investment strategies*, The Journal of Applied Business and Economics **26** (2024), no. 4, 63-74.

[17] R. G. Sloan, *Fundamental analysis redux*, The Accounting Review **94** (2019), no. 2, 363-377.

[18] P. E. Tsinaslanidis and A. D. Zapranis, *Technical analysis for algorithmic pattern recognition*, Springer publication, 2016.

[19] P. Yang and X. Hou, *Research on dynamic characteristics of stock market based on big data analysis*, Discrete Dynamics in Nature and Society **1** (2022), no. 2, 8758976.

[20] H. Yang and et al., *Deep reinforcement learning for automated stock trading: An ensemble strategy*, in Proceedings of the first ACM international conference on AI in finance, 2020.

[21] Z. Zhang, S. Zohren and S. Roberts, *Deep reinforcement learning for trading*, arXiv preprint arXiv: 1911.10107 (2019).