The Simulation Framework for Automated Trading Algorithms on Capital Markets

Doina BANCIU[1], Anca-Gabriela PETRESCU[2]*, Ionica ONCIOIU[3], Marius PETRESCU[1], Florentina-Raluca BÎLCAN[2], Adrian-Ionuț GHIBANU[2]

[1] Academy of Romanian Scientists, 3 Ilfov Street, Bucharest, 050044, Romania
doina.banciu@aosr.ro, profdrmpetrescu@yahoo.com

[2] Valahia University of Targoviste, 2 Carol I Blvd., Targoviste, 130024, Romania
anki.p2007@gmail.com (**Corresponding author*), bilcan.florentina.raluca@gmail.com, ighibanu@yahoo.com

[3] Titu Maiorescu University, 189 Calea Vacaresti Street, Bucharest, 040051, Romania
ionica.oncioiu@prof.utm.ro

Abstract: Recent research on automated trading algorithms has focused on evaluating their effectiveness on various financial markets and comparing their performance with that of other trading methods. This paper proposes an innovative framework for the simulation of trading algorithms with the purpose of supporting the development of automated decision systems for traders using agent-based modeling and reinforcement learning methods in a market context. The obtained results demonstrate that a trading agent can learn and optimize the trading strategies, even if performance variations were observed in relation to environmental changes. The success of the trading agent depends on the commitment of trading partners and the implementation of risk management in correlation with the market norms. The proposed model provides a valuable platform for further studies on learning behavior in trading based on the decisions of a real trader.

Keywords: Artificial neural networks, Multi-agent systems, Transaction databases, Reinforcement learning methods.

1. Introduction

The fast growth of information technology and artificial intelligence (AI) has increased the adoption of machine learning algorithms in the finance sector, which has led to the development of automated trading systems capable of making on-the-spot decisions by analyzing big data (Gil et al., 2012; Sun et al., 2018; Strader et al., 2020). They help improve the effectiveness of trade in addition to making quick responses to market dynamism hence risk mitigation and profits maximization (Haykin, 2009).

In this context, agent-based modeling (ABM) and reinforcement learning (RL) stand out as some of the potential techniques for building such systems because they make it possible to simulate trading behavior and strategize optimization by studying complex and dynamic conditions in the marketplace (Ban, 2024; Zhu & Kwong, 2010; Zhao & Zhu, 2015). Although there is an extensive literature examining the use of machine learning algorithms and simulation models on global financial markets, research targeting emerging markets such as Romania is still limited.

This article intends to explore the potential of using ABM and reinforcement learning (RL) to create a simulation framework that facilitates the development of automated trading algorithms on the Romanian capital markets. It addresses a significant gap in the specialty literature by adapting advanced methodologies to the unique characteristics of the Romanian market and providing an innovative perspective on the practical implementation of these techniques locally.

The proposed model is based on the dynamic interaction of agents with the market environment and allows not only the simulation of transactions, but also the evaluation of the performance of these agents according to market conditions. Thus, the algorithms will learn to recognize price patterns and trading volumes, thereby optimizing the moment a certain product enters or exist the market to maximize profits and minimize risks.

Additionally, the proposed model is able to incorporate real-time agent learning and adaptation, thus providing a holistic approach to optimizing decision-making in automated trading. Modeling the environment and action options creates the right framework for an agent to solve the problem, and to control the agent`s behavior effectively, an incentive mechanism is formulated and an experience-based learning method is used (Chen & Huang, 2022).

This research not only verifies and validates the existing simulation models on the Romanian capital market, but also proposes a series of improvements by using a new reward function and by integrating a wide range of risk factors

https://doi.org/10.24846/v33i4y202405

and opportunities specific to financial markets. The article underlines the efficiency and applicability of the proposed model through a simulation environment specific to the Romanian market. It clearly states that a complex simulation technology should be used for creating automated decision systems for capital markets.

This paper is structured as follows. Section 2 includes a brief presentation of the literature. Section 3 presents the proposed model and methodology. Section 4 is devoted to a detailed analysis of the results obtained from simulations, and of the advantages and limitations of the proposed approach. A discussion of the obtained simulation results is presented in Section 5. Finally, Section 6 sets forth the conclusion of the research findings and outlines future development directions regarding the potential of automated decision systems in transforming global financial markets.

2. Theoretical Framework

Artificial neural networks can be used as alternatives for autoregressive, moving average, multidimensional regression or combined models (Liagkouras, 2019; Dempster & Leemans, 2006). The modeling of time series with the help of neural networks is done by choosing the causal variables, choosing the number of nodes in the hidden level or by building the training dataset (Cernazanu, 2008; Zhao & Zhu, 2015).

The selection of the number of nodes in the hidden layer depends on identifying the optimal architecture of the network and usually is laid down heuristically, although some authors have put forward several selection algorithms (Haykin, 2009; Filip, 2012; Deng et al., 2017)

The literature indicates that in the last few decades, automated trading has achieved significant improvements due to the higher computing power of machines and the availability of real-time financial data (Weiss, 2000; Liagkouras, 2019). These trading algorithms are trades executed by computer programs following rules based on which trade orders are generated (Nakashima et al., 2005). They have been widely applied on financial markets because they can execute trades at a speed and volume beyond the reach of human traders (Haykin, 2009).

Dodd & Gilbert (2016) further investigated the use of neural networks and other machine learning techniques for creating predictive models for financial markets. They argue that this approach has a significant advantage in detecting complicated non-linear relationships between financial variables due to the ineffectiveness of standard methods at capturing such structures.

Similarly, Akopov (2024) stresses that while algorithmic trading strategies do bring about large benefits concerning execution speed and correctness, they carry substantial risks coming from market volatility or potential programming errors.

Some previous studies have indicated the potential of RL in automated trading (Ban, 2024; Deng et al., 2017). For the purpose of determining the quality of RL algorithms, it is possible to evaluate either the speed at which a policy is learned or the quantity of rewards that are accumulated throughout the learning process (Dempster & Leemans, 2006).

Zhao & Zhu (2015) showed that their algorithm could learn to be highly profitable with the correct strategy selection for different market conditions. In the continuation of these researches, Sewak (2019) introduced Deep Q-Networks (DQN) with RL by fusing neural networks into the framework of traditional approaches optimally based on their strengths. Following the completion of this improvement, this strategy has been successfully used in a variety of areas, including automated trading, as well as in the training of agents to make optimal decisions in environments that are constantly changing and dynamic.

On the other hand, the DQN represents a groundbreaking approach that successfully combined deep learning and reinforcement learning, enabling agents to learn from raw, unprocessed data, such as stock prices that are directly closing (Papageorgiou, Gkaimanis & Tjortjis, 2024; Lin et al., 2022). This approach guided them in formulating profitable strategies through propagation steps, known as iterations in reinforcement learning, before arriving at the final deployment actions that the agents learned when interacting with the environment.

In another study, Filip (2012) applied ABM to the study of how speculative bubbles and market crashes form. His paper provided empirical evidence that ABM can make a contribution to better informing how the collective behavior of agents can lead to emergent phenomena, such as speculative bubbles. This issue is of extreme importance for the development of automated trading systems because it provides the possibility of simulating different market situations and the behavior of trading agents in these situations.

3. Data and Methodology

The proposed model is structured as a modular architecture, which allows flexibility and scalability in the simulation and learning of trading behaviors. The overall architecture consists of the following main components:

• *market environment*: represents the simulated capital market, in which agents carry out transactions. Chen & Huang (2022) take into account the environment as including market circumstances, economic variables, and risk factors affecting agent behaviour.

• *trading agencies*: represent autonomous entities that can make decisions to buy and sell financial assets. Dicks et al. (2024) argue that the agents are designed to learn and modify trading methods based on market feedback.

• *learning engine*: using reinforcement learning (RL), the learning engine allows agents to optimize their trading strategies. From the point of view of Shavandi & Khedmati (2022), the reward function that is connected with this engine is responsible for determining whether or not each transaction was successful and for guiding the learning process.

• *the simulation interface*: ensures the interaction between the agents and the market environment, allowing the simulation of transactions, the collection of data and the analysis of the results (Dicks et al., 2024).

Agents collect market data, such as asset prices and trading volume, and use it to make informed decisions. The agents` perception of the market is influenced by the use of algorithms in data processing such as technical analysis and fundamental analysis. Agents, in this case, are traders who engage the market and other agents in trade based on learned strategies.

In comparison with other dedicated trading strategies, the proposed ABM-RL model outperforms buy-and-hold and moving average strategies by virtue of its ability to learn from the market environment and continuously adjust its trading strategy. By continuously learning based on market feedback, this algorithm can not only optimize returns in favorable periods, but also minimize the risks in adverse market conditions, thereby avoiding significant losses that buy-and-hold strategies cannot avoid. While moving average-based strategies only respond to a limited number of factors (price movements relative to the average), this model also considers the complex interactions between agents and other aspects of market dynamics, which makes it more robust and adaptable in the face of uncertainty.

In terms of financial performance, the ABM-RL model has significant advantages. Indicators such as the Sharpe ratio and maximum drawdown indicate that risk management is more effective, while buy-and-hold strategies and moving averages are susceptible to market volatility and long-term losses. In addition, the proposed model can flexibly integrate multiple types of market data, not just historical prices, allowing it to trade more actively and efficiently.

For this study the period between 2013 and 2023 was selected for OMV Petrom (SNP) from the Bucharest Stock Exchange data portal where this information is made publicly available (Bucharest Stock Exchange, 2024).

Historical data on SNP share price, trading volumes, international oil price developments, the company's published annual and quarterly financial reports, as well as major energy price-influencing events (government regulations, global oil price fluctuations) are collected to train and validate algorithms. Machine learning algorithms simulate various trading scenarios based on historical data (Georgescu, 2011; Strader et al., 2020).

During these simulations, different strategies for buying and selling SNP shares are tested, both during periods of increase and during periods of decrease in oil prices. Based on simulated market conditions, algorithms will learn to recognize price patterns and trading volumes, thus optimizing market entry and exit timing to maximize profits and minimize risks.

After the simulations indicate a robust trading strategy was reached, the algorithms are implemented in the real environment to automatically execute trades based on the price movements of SNP shares. For example, if the algorithm detects a buying opportunity when the price of oil is rising, it can initiate a purchase of SNP shares. The evolution of the weighted average price according to volume (vwap) in the analyzed period is presented in Figure 1.

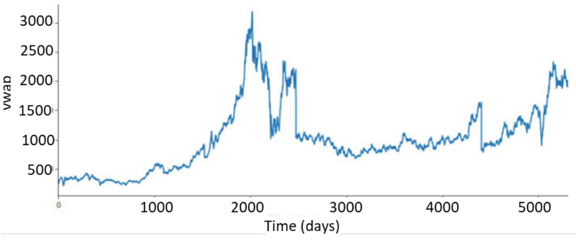


Figure 1. The evolution of the weighted average price according to volume (SNP) (DataStream, authors' calculations)

Building the network first involves building the training examples. This is done starting from equation (1), depending on the type of network used:

$$\begin{array}{ll} x_0 = \left[x_0^1, x_0^2, \ldots, x_0^I, 0, \ldots, 0\right] & \hat{y}_1 = f\left(x_0\right) \\ x_2 = \left[x_1^1, x_1^2, \ldots, x_1^I, s_1^1, s_1^2, \ldots, s_1^H\right] & \hat{y}_2 = f\left(x_2\right) \\ \vdots & \vdots \\ x_t = \left[x_{t-1}^1, x_{t-1}^2, \ldots, x_{t-1}^I, s_{t-1}^1, s_{t-1}^2, \ldots, s_{t-1}^H\right] & \hat{y}_{t+1} = f\left(x_t\right) \end{array} \tag{1}$$

where s_t (status level) is calculated according to equation (2), with the specification that $S_0 = 0$:

$$\begin{gathered} s_t^h = u_h\left(t-1\right) = \\ F\left(\sum_{i-1}^{I} \pi_{ih} x_i + \sum_{s-1}^{I} \pi_{sh} s_{t-1}^s\right) \circ h = \overline{1,H} \end{gathered} \tag{2}$$

Equation (3) is derived from equation (1) and represents the way of building training examples for the case of time series modeling:

$$\begin{aligned} & x_1 = \left[y_1, y_2, \ldots, y_I\right] \\ & \quad \hat{y}_{I+1} = f\left(y_1, y_2, \ldots, y_I\right) \\ & x_2 = \left[y_2, y_3, \ldots, y_{I+1}, s_1^1, \ldots, s_1^H\right] \\ & \quad \hat{y}_{I+C+2} = f\left(y_2, y_3, \ldots, y_{I+1}, s_1^1, \ldots, s_1^H\right) \\ & \vdots \\ & x_{t+1} = \left[y_{t-I+1}, y_{t-I+2}, \ldots, y_t, s_{t-1}^1, \ldots, s_{-1}^H\right] \\ & \quad \hat{y}_{t+1} = f\left(y_{t-I+1}, y_{t-I+2}, \ldots, y_t, s_{t-1}^1, \ldots, s_{-1}^H\right) \end{aligned} \tag{3}$$

where s_t is calculated according to equation (2), with the specification that $S_0 = 0$.

According to the specialized literature, a RL model includes a set of algorithms for solving the model of a decision problem (Ban, 2024). To determine the final reward function for the potential trader, which influences the agent's actual behavior, the partial aspects of a reward are converted into a convex combination, and each is given a weighting coefficient ω. This results in the following reward function:

$$r_{t+1} = \frac{\omega_1 r^1\left(m_t\right) + \omega_2 r^2\left(Trd, a\right)_t + \omega_3 r^3\left(E_{t+1}\right)}{\sum_{i=1}^{3} \omega_i} \tag{4}$$

for $\omega 1 \geq 0$ with $r_t \in [0,1]$ and $t \geq 0$. To investigate what effects the consideration of certain parts of the reward function has on the learning behavior of the agent, the weighting factors ω1, ω2 and ω3 are used as parameters in different configurations.

For this it is necessary to determine the total probability of the occurrence of the event that a trader is a real customer. The probability that a trader is a real customer is therefore defined as $P(Trd0) = \psi$. For the occurrence of the event that a trader is a non-client, the counter probability is $P(Trd1) = 1 - \psi$. Nevertheless, the following probabilities of occurrence can be calculated for state changes:

$$P\left(Trd0 \middle| a0\right) = \frac{\left(1-\varphi\right)\psi}{\chi\left(1-\psi\right) + \left(1-\varphi\right)\psi} \tag{5}$$

$$P\left(Trd0 \middle| a0\right) = \frac{\psi\left(1-\psi\right)}{\left(1-\varphi\right)\psi + \chi\left(1-\psi\right)} \tag{6}$$

$$P\left(Trd0 \middle| a1\right) = \frac{\varphi\psi}{\varphi\psi + \left(1-\chi\right)\left(1-\psi\right)} \tag{7}$$

$$P\left(Trd1 \middle| a1\right) = \frac{\left(1-\chi\right)\left(1-\psi\right)}{\varphi\psi + \left(1-\chi\right)\left(1-\psi\right)} \tag{8}$$

4. Results

The integration of ABM and RL represents an important step in the development of automated decision systems for capital markets. ABM provides a framework for simulating complex interactions between agents in a market environment, while RL allows agents to learn and optimize their trading strategies based on these interactions.

Following the application of the methodology described, a series of relevant results were obtained in the context of simulating the trading behavior of autonomous agents based on the ABM model and reinforcement learning (RL) for SNP.

The simulations were performed on a dataset from the analyzed timespan including periods of high and low volatility, thus obtaining data on the performance of agents in various scenarios, which emphasizes their ability to learn and adapt. This means, in this case, that the simulation of Mauer et al. (2017) uncertainty modeling is carried out by modifying the initial distribution of vector components as shown in Figure 2.

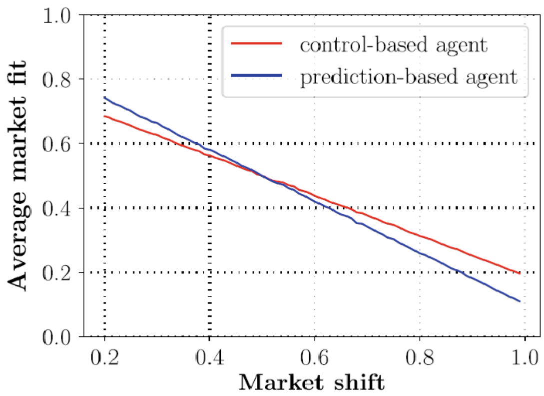


Figure 2. Simulation representation for using the model based on control and prediction

Control-based agents achieve a better market fit in comparison with prediction-based agents when market change occurs as a result of stock (SNP) price increases or trading volumes. The number of prediction-based and control-based agents is set to 20 for each category, and the number of simulation runs is 80.

Next, by applying equation (4) the reward function r_t is obtained for the weighting factor ω_3, which was initialized with the values 1 and 4 to illustrate the associated change of the reward function r_t (see Figure 3).

Similarly, ω_3 parameters can be used to control which aspects of the agent`s effective behavior influences the agent's learning experience. In the case of SNP, the results obtained when using two parameter configurations are illustrated in Figure 4. On top of that, the average rewards per transaction are compared for an efficient agent and a randomly acting agent over a period of 6,000 transactions, using the parameter configurations included in Figure 4.

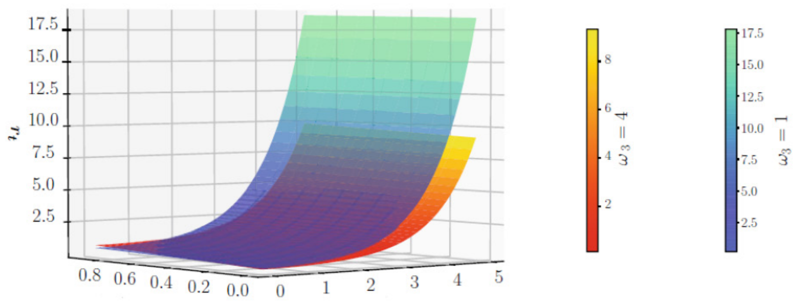
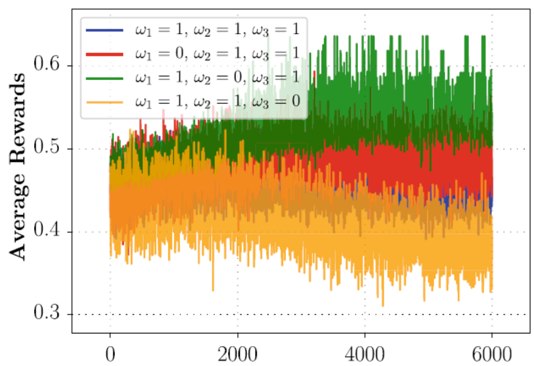
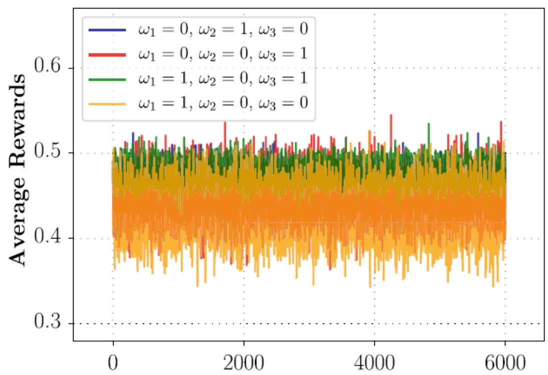


Figure 3. The graph of the reward function with the variation of the weighting factor



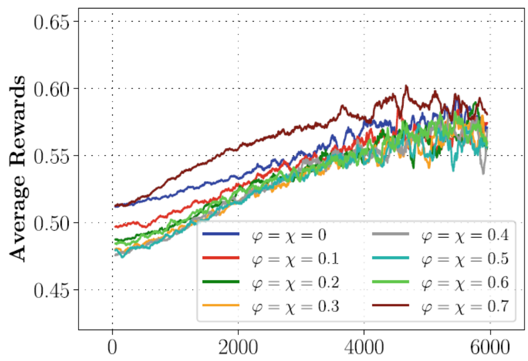
(a) The average rewards received per transaction by an efficient agent



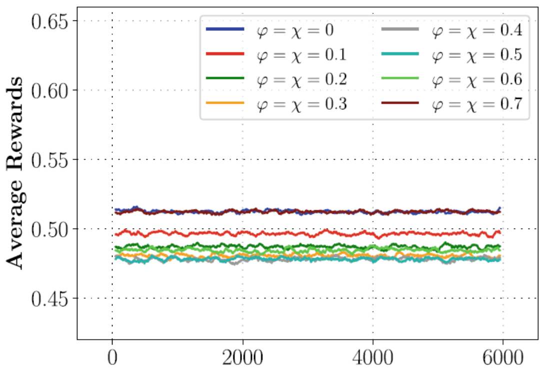
(b) The average rewards received per transaction by an agent acting randomly

Figure 4. Comparison of agent learning behavior with a different weighting of the convex combination terms of the reward function

If the average trading volumes in equations (5) to (8) are taken into account, the agents' learning behavior can be illustrated as it is shown in Figure 5.



(a) The transactions made by an efficient agent



(b) The transactions made by an agent acting randomly

Figure 5. The learning behavior of agents taking into account the change in trading volumes

5. Discussion

The integration of agent-based modelling (ABM) and reinforcement learning (RL) in the development of automated decision-making systems for capital markets is a major contribution that provides a sophisticated framework for simulating the trading behavior of autonomous agents. ABM is able to simulate in detail the complex interactions between market agents, while RL enables them to learn and optimize trading strategies based on these interactions. The simulation results show the effectiveness of this approach, especially in the context of volatile markets.

The simulations were conducted on a dataset covering periods of high and low volatility, allowing the evaluation of the agents` ability to learn and adapt to different market conditions. The obtained results show that control-based agents are able to adapt better to the market than prediction-based agents when there is a market shift, such as an increase in SNP stock prices or variations in trading volumes. These results highlight the fact that agents that adjust their behavior based on market changes (and rely on immediate feedback) tend to perform better than those that strictly rely on long-term predictions, especially in unpredictable market environments.

The analysis of the reward function using the weight factor ω_i shows how the employed weight parameters affect an agent's decision and its performance. The changes in the weight values show clear changes in the agent's behavior and highlight the model's flexibility in adjusting the learning parameters to better adapt to the market environment (as shown in Figure 3). This is an important component of RL as it allows the agent to prioritize different success factors, such as profit maximization or risk minimization, depending on market conditions.

The comparison of the performance of the efficient agent with that of the random agent revealed a significant difference with regard to the average reward per trade. The agent configured with the specified weight parameters achieved a superior performance, confirming that properly implemented machine learning algorithms have a clear advantage in automated trading. This is evident in Figure 4, which shows that the learning behavior of the agent was significantly improved by using an appropriate combination of convex terms in the reward function.

Therefore, the obtained results clearly demonstrate the advantages of the proposed approach in the development of automated trading algorithms. The approach based on ABM and RL not only features a superior adaptability and learning capabilities in dynamic financial markets, but it also improves the flexibility of adjusting model parameters to cope with complex market scenarios. This flexibility, along with an improved performance relative to random strategies, highlights the great potential of the proposed model in optimizing trading decisions and reducing the risks associated with market volatility.

6. Conclusion

This paper looked into how to create a simulation framework for agent-based models (ABMs) and reinforcement learning (RL) algorithms that can be used for automated trading algorithms on capital markets. The SNP-related results demonstrate that the employed methods are effective in creating trading agents capable of learning and optimizing strategies in response to changing market conditions.

The use of ABM made it possible to capture emerging phenomena and complex capital market dynamics, thereby obtaining detailed and realistic insights into market behavior by contrast to traditional static models. These findings directly contribute to expanding the existing literature on the applicability of ABM and RL on emerging financial markets.

One of the main advantages of this model lies in the ability of the agents to learn from experience and dynamically adapt to market changes. This ability enables the agents to quickly respond to fluctuations in the market environment, thereby maximizing their long-term performance.

This study also highlights the importance of properly tuning algorithm parameters and input data quality to achieve an optimal trading agent performance. This is consistent with the existing literature, which shows that the performance of the employed algorithms depends on these adjustments (Georgescu, 2011; Gil et al., 2012; Papageorgiou, Gkaimanis & Tjortjis, 2024). However, this study makes a significant contribution by applying these techniques in a specific context (e.g. the Romanian capital market) and validating them in a real environment.

Another important contribution of this study lies in the ability of the employed algorithms to better predict the market's reaction to global oil price fluctuations and optimize trades to maximize returns. Moreover, by managing risks and minimizing losses, these automated strategies provide investors with a higher level of security while improving the efficiency of the Romanian capital market. These results are consistent with the existing literature, but also provide new opportunities for expanding the applicability of these techniques to other markets and sectors.

This study paves the way for a number of future research directions. First, it would be useful to extend the model so as to include more market variables and simulate more complex scenarios by introducing the concept of agent-based modeling to map emerging systems. And finally, new reinforcement learning algorithms can be developed that are more computationally efficient and can quickly learn from limited data.

Second, the integration of more advanced machine learning techniques, such as convolutional neural networks or recursive neural networks could further improve the ability of agents to learn and adapt to dynamic environments. Furthermore, extending and applying this model to other companies from the Bucharest Stock Exchange, could contribute to the digital transformation of trading processes and enhance the competitiveness of the Romanian capital market.

Finally, another unexplored potential as regards the current research is the integration of alternative data into the agent learning process. In addition to historical prices and volumes, data such as market sentiment based on financial news, social media, or analyst reports can improve the ability of agents to predict market movements and make more informed decisions. Integrating this unstructured data sources through natural language processing (NLP) techniques can represent an important step in optimizing ABM-RL models and provide agents with a competitive advantage in identifying market trends before they are discovered through traditional methods.

REFERENCES

Akopov, A. S. (2024) A Clustering-Based Hybrid Particle Swarm Optimization Algorithm for Solving a Multisectoral Agent-Based Model. *Studies in Informatics and Control*. 33(2), 83-95. doi: 10.24846/v33i2y202408.

Ban, L. (2024) Optimization of Energy Management Algorithm for Hybrid Power Systems Based on Deep Reinforcement Learning. *Studies in Informatics and Control*. 33(2), 15-25. doi: 10.24846/v33i2y202402.

Bucharest Stock Exchange. (2024) *Bucharest Stock Exchange - Homepage.* http://www.bvb.ro [Accessed 22nd February 2024].

Cernazanu, C. (2008) Training Neural Networks Using Input Data Characteristics. *Advances in Electrical and Computer Engineering.* 8(2), 65-70. doi: 10.4316/AECE.2008.02012.

Chen, H.-Y. & Huang, S.-H. (2022) Generating a trading strategy in the financial market from sensitive expert data based on the privacy-preserving generative adversarial imitation network. *Neurocomputing*. 500, 616-631. doi: 10.1016/j.neucom.2022.05.039.

Dempster, M. A. H. & Leemans, V. (2006). An automated FX trading system using adaptive reinforcement learning. *Expert Systems with Applications*. 30(3), 543-552. doi: 10.1016/j.eswa.2005.10.012.

Deng, Y., Bao, F., Kong, Y., Ren, Z. & Dai, Q. (2017) Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems*. 28(3), 653-664. doi: 10.1109/TNNLS.2016.2522401.

Dicks, M., Paskaramoorthy, A. & Gebbie, T. (2024) A simple learning agent interacting with an agent-based market model. *Physica A: Statistical Mechanics and its Applications*. 633, 129363. doi: 10.1016/j.physa.2023.129363.

Dodd, O. & Gilbert, A. (2016) The Impact of Cross-Listing on the Home Market's Information Environment and Stock Price Efficiency. *Financial Review*. 51(3), 299–328. doi: 10.1111/fire.12110.

Filip, F. G. (2012) A Decision-Making Perspective for Designing and Building Information Systems. *International Journal of Computers, Communications & Control*. 7(2), 264-272. doi: 10.15837/ijccc.2012.2.1408.

Georgescu, V. (2011) An Econometric Insight into Predicting Bucharest Stock Exchange Mean-Return and Volatility–Return Processes. *Economic Computation and Economic Cybernetics Studies and Research*. 3(3), 25-42.

Gil, D., Girela, J. L., De Juan, J., Gomez-Torres, M. J. & Johnsson, M. (2012) Predicting seminal quality with artificial intelligence methods. *Expert Systems with Applications*. 39(16), 12564-12573. doi:. 10.1016/j.eswa.2012.05.028.

Haykin, S. (2009) *Neural Networks and Learning Machines* 3rd edn. Prentice Hall Publishing.

Liagkouras, K. (2019) A New Three-Dimensional Encoding Multiobjective Evolutionary Algorithm with Application to the Portfolio Optimization Problem. *Knowledge-Based Systems*. 163, 186-203. doi: 10.1016/j.knosys.2018.08.025.

Lin, Y.-C., Chen, C.-T., Sang, C.-Y., Huang, S.-H. (2022) Multiagent-based deep reinforcement learning for risk-shifting portfolio management. *Applied Soft Computing*. 123, 108894. doi: 10.1016/j.asoc.2022.108894.

Mauer, R., Wuebker, R., Schlüter, J. & Brettel, M. (2017) Prediction and control: An agentbased simulation of search processes in the entrepreneurial problem space. *Strategic Entrepreneurship Journal*. 12(2), 237-260. doi: 10.1002/sej.1271.

Nakashima, T., Ariyama, T., Kitano, H. & Ishibushi, H. (2005) A Fuzzy Rule-Based Trading Agent: Analysis and Knowledge Extraction. In: Halgamuge, S. K. & Wang, L. (eds.) *Computational Intelligence for Modelling and Prediction.* Berlin, Germany, Springer, pp. 265-277.

Papageorgiou, G., Gkaimanis, D. & Tjortjis, C. (2024) Enhancing Stock Market Forecasts with Double Deep Q-Network in Volatile Stock Market Environments. *Electronics*. 13(9), 1629. doi: 10.3390/electronics13091629.

Sewak, M. (2019) *Deep Q Network (DQN), Double DQN, and Dueling DQN - A Step Towards General Artificial Intelligence*. In: Sewak, M. *Deep Reinforcement Learning - Frontiers of Artificial Intelligence*. Singapore, Springer, pp. 95-108.

Shavandi, A. & Khedmati, M. (2022) A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Systems with Applications*. 208, 118124. doi: 10.1016/j.eswa.2022.118124.

Strader, T. J., Rozycki, J .J., Root, T. H. & Huang, Y.-H. J. (2020) Machine Learning Stock Market Prediction Studies: Review and Research Directions. *Journal of International Technology and Information Management*. 28(4), 63-83. doi: 10.58729/1941-6679.1435.

Sun, Y., Zhang, M., Chen, S. & Shi, X (2018) A Financial Embedded Vector Model and Its Applications to Time Series Forecasting. *International Journal of Computers, Communications & Control*. 13(5), 881-894. doi: 10.15837/ijccc.2018.5.3286.

Wang, H., Li, S., Tian, Y. & Aitouche, A. (2016) Intelligent Proportional Differential Neural Network Control for Unknown Nonlinear System. *Studies in Informatics and Control*. 25(4), 445-452. doi: 10.24846/v25i4y201605.

Weiss, G. (2000) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, Massachusetts, USA, MIT Press.

Zhao, D. & Zhu, Y. (2015) MEC - A Near-Optimal Online Reinforcement Learning Algorithm for Continuous Deterministic Systems. *IEEE Transactions on Neural Networks and Learning Systems*. 26(2), 346-356. doi: 10.1109/TNNLS.2014.2371046.

Zhu, G. & Kwong, S. (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*. 217(7), 3166-3173. doi: 10.1016/j.amc.2010.08.049