Type of the Paper Article ***DEPUS SPRE EVALUARE IN VEDEREA PUBLICARII***

Developing a system on intelligent multiagent with applicability in E-Commerce

Catalin Dumitrescu [1], Firstname Lastname [2] and Firstname Lastname [2,*]

[1] Affiliation 1; e-mail@e-mail.com
[2] Affiliation 2; e-mail@e-mail.com
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials)

Abstract: The benefits of automating e-commerce activities are - a properly trained system will always react much faster to changes in market trends than the human factor, and the decisions it can make are not affected by the inconsistency that characterizes most of the time human decisions. The main objective of the research within the article is to carry out the study necessary for the design, realization, and implementation in the real environment of a multi-agent transaction system that will automate the electronic commerce activity. The system's automated trading decisions the presented multi-agent is based on predictions regarding the evolution of a product's trend, generated by intelligent agents that use artificial intelligence methods in their analysis.

Keywords: e-commerce; intelligent agents; adaptation strategi

Citation: To be added by editorial staff during production.

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

1. Introduction

Currently, e-commerce is taking over an ever-increasing role within many companies, because of the possibilities for improvement of the way of interaction with partners business (customers and suppliers) and growth the efficiency of operations. In most e-commerce applications, buyers search through a catalog of products according to well-defined categories and do purchase procedures at a fixed price. Through increasing the degree of automation, trade it becomes more dynamic, personalized, and sensitive to the context [1].

From the buyer's perspective, it is desirable for it to benefit from software that it goes through all the electronic sales markets existing to find the best option of purchasing the good or service desired and to carry out the actual process of purchase, paying and arranging delivery terms. From the seller's perspective, it is preferable for it to benefit from software that to diversify its offer depending on the customer with who negotiates, the competition in the field and of the current state of the business. To obtain this degree of automation and flexibility new models are needed software for electronic commerce [2]. One option elegant and currently under development o represents the use of interactive intelligent agents. An intelligent agent is a program that has great flexibility in reaching certain objectives. The features you need to have an agent are the following: autonomy, reactivity, and proactivity [3].

During the pursuit of objectives, agents need to interact with other similar independent agents. This interaction can range from simple communications to more elaborate forms of social interaction (such as participation in online auctions, negotiations from the owner, cooperation with other agents).

In general, electronic commerce covers any form of business or administrative transaction or informational exchange that perform using any information and communication technology.

Currently, e-Commerce impacts the traditional way of collaboration and brings major changes in the lifestyle. In this article, reference is made to e-commerce from a "business-to-business" perspective in which decision support systems play an active role by empowering participants who are given the right tools for decision-making. e-Commerce produces changes in the organization of companies, but also in personal life, through its multiple implementations, from the classic transaction process to a much more simplified process of placing an order. All this forms a very dynamic and complex framework in which e-Commerce plays a central role. e-Commerce is perceived by the public today as an "online store", when, in fact, it participates in a wider sense in the progress of society.

Multiagent systems (MAS) have emerged as a new methodology of approach to the problems of large-scale organization of software systems. This methodology provides a conceptual model that helps maintain constraints; conventional software engineering tasks being impossible to accomplish. In recent years, MAS have been used in various fields of computer science and engineering and have become a versatile tool that addresses the needs of software engineering [4]. They also expand the spectrum of computer science research and have increasingly drawn attention to a wide range of fields, moving from theoretical study to practice. An agent is a software entity that actively searches for ways to complete its tasks. Intelligent agents can acquire knowledge through problem-solving processes. The study of the social behavior of agents in cognitive science is an important part of the field of intelligent agents. Software agents focus on interactions and collaborations to achieve their goals in a context that changes in an often-unpredictable manner [5]. The need to use agents comes from the complexity of the systems large software, which bring new design problems that conventional technology fails to address. For example, mobile agents have been proposed to meet the needs of the client/server model, for a client to be able to migrate to the server to perform operations that other mechanisms cannot handle efficiently. In a distributed dynamic system, agents have self-regulating capability and can simplify the architectural design of the system [6]. The design of such a system can be extremely complicated in traditional object-oriented modeling architectures.

2. Materials and Methods

2.1. Intelligent agents in business-to-consumer commerce

Business-to-consumer (B2C) commerce refers to online retail transactions with individual customers, where buyers can transact through a company's web page. To be able to analyze the tasks of an intelligent agent in this case, we will use the Customer Buyer Behavior Model (CBB) to model consumer behavior [7]. The CBB model considers that the development of B2C e-commerce takes place in seven stages: identification of needs, product brokerage (intermediation), formation of coalitions between buyers, seller brokerage, negotiation, acquisition and delivery, service evaluation. Through the lens of this model, it is assumed that the intelligent agent acts as an intermediary in five of these stages: identification of needs, product brokerage (intermediation), formation of coalitions between buyers, brokerage the seller and the negotiation [8], [9]. In the needs identification stage, the client expresses a request for a product or service, which can be stimulated by the user agent (also called information agent or notification). Such an agent needs a profile of the user, which can be obtained by observing the behavior to the user, through direct acquisition techniques or through inductive logic programming techniques. Once known the profile of the users of to the intelligent agent, it will be able to inform the user whenever the desired good or service will be available. The product brokerage stage involves an agent determining which product/service to buy to satisfy the need identified in the previous stage. The main techniques used for product brokerage are: filtering by features, collaborative filtering and constraint-based filtering [10]. Filtering by characteristics involves selecting products based on keywords associated with them (characteristics). Collaborative filters refer to sending personalized recommendations to an agent based on the similarity between various buyer preference profiles [11], [12], [13]. Constraint-based filters involve an agent specifying constraints on the desired product.

2.2 Computational intelligence techniques

Computational Intelligence techniques are increasingly used to solve problems that cannot be addressed by traditional techniques or when the available information is insufficient to create a model based on which to a solution algorithm can be developed. In this chapter, two of the techniques specific to the field were presented, techniques used in the development of the realized multi-agent system, namely artificial neural networks and support vector machines. The main quality of neural networks lies in the fact that they can store knowledge that they can later use in new situations [14]. The acquisition of knowledge in neural networks is done by storing some values at the level of synaptic weights, values that depend equally on the architecture of the network that includes them. The process of assigning values to the synaptic weights in a neural network is called the learning process or the learning process training. As we have seen, two fundamental types of learning are distinguished: supervised learning and unsupervised learning [15]. In the case of supervised learning, the network is presented with a set of learning examples, represented by input-output vector pairs, characteristic of the knowledge to be acquired. The values of the input vectors are entered into the calculations, affecting the current weights of the network and determining the appearance at the output of some values that are compared with the expected results. Depending on the deviations between the values obtained and the expected ones, the weights are adjusted, so that the differences become zero or as small as possible. The entire process is repeated until the network is considered to have been fully trained. The learning algorithm is based on the fact that if the network behaves in a certain way in known situations, it will keep its behavior in new, unknown situations. The data used for learning is called training data, and the data used to track the behavior of the network in new situations is called test data. In the case of unsupervised learning, the set of data for learning consists only of a set of input vectors which, as in the case of supervised learning, affect the weights stored in the network. The difference with supervised learning is that the weights are adjusted in this case so that in the case of close input vectors, the same output vector or output vectors with very close values are obtained. However, if the solution obtained as a result of training the network is acceptable from the point of view of the error rate, the fact that the network has reached a local minimum point or in a global minimum is no longer relevant.

Support Vector Machines (SVM) represent an effective method in designing a feedforward network with a single hidden layer of nonlinear units. As the name suggests, the design of this type of neural networks is based on extracting a subset of the training data that serves as a support vector and represents a stable feature of the data. In the specialized literature, SVMs have established themselves as the most used algorithm due to their very good generalization performances, rigorous theoretical foundations, their relatively easy implementation and the ability to provide outstanding performances in shape recognition and regression problems. Supervised learning was used in the realization of the agent system presented in the next sub-chapter, the training and test data being synthetic data.

2.3 Multi-agent automatic transaction system using computational intelligence techniques

This chapter presents in detail the architecture, implementation and operation of the multi-agent system that actually constituted the object of study of the research. The architecture of the system is presented in figure 1.

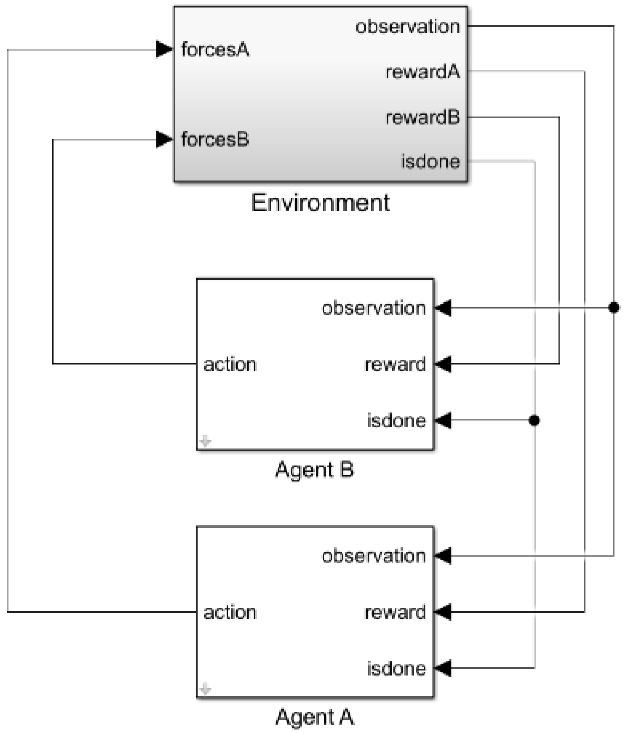


Figure 1. Architecture of the multi-agent system for e-commerce

Figure 1 shows the architecture of the realized multi-agent system. The system is connected to the e-commerce trading system connected to the Internet through a Web Gateway platform. The two-way coupling to this is done through the interface agent - Environment which transmits the received data to the merchant agent, possibly we can also introduce a supervisor type agent [16]. The seller's agent and the buyer's agent are connected with the help of a specialized database type agent. The system includes three specialized operations for each agent in trend analysis and prediction: one that uses neural networks and the second that uses support vector machines (SVM).

In the next chapter, it is presented how the predictive agents are trained and details are provided about the data used for their training [17]. In the previous section, the use of two of the best performing computational intelligence techniques in terms of predictive accuracy was exemplified: neural networks and support vector machines. Their implementation was done in the MATLAB language.

The distinctive property of an agent is that of autonomy, which it implies the ability of the agent to survive in a changing environment. An agent has the ability to sense conditions and make decisions about how to react accordingly. Adaptability requires learning capabilities that allow the agent to be able to adapt its decisions based on past experience. Moreover, an agent-oriented design should address the robustness of the system, it should be reliable when unexpected events occur. In the design of large, complex, real-life systems, an agent is an abstract entity that helps design components that address different aspects of a problem. Each agent is designed in the paradigm best suited to solve its part of the problem. A multiagent system is used to solve a complex problem which cannot be solved by a single system entity. Coordinating the behaviors of independent agents is a central part of multiagent system design. Multiagent systems are often classified according to their characteristics agents into two categories: self-interested agents and cooperative agents [18]. Self-interested agents are based on economic concepts where an agent is assumed to be a utilitarian who always tries to maximize an appropriate utility function. This assumption is widely used in microeconomics and game theory. Thus, researchers often use economic tools and game theory tools to model agencies. Self-interested agents have tend to close their private information and fail to react if no benefit is available. Cooperative agents are constructed so that they are able to engage in cooperative behavior.

3. Results

We consider an e-commerce system based on adaptive agents in which each participant has associated an intelligent agent that acts in the interest of its owner. In such a system there are several types of agents, sales agents, buyer agents, mediator agents and broker-type agents. Figure 2 shows the general model of an adaptive agent.

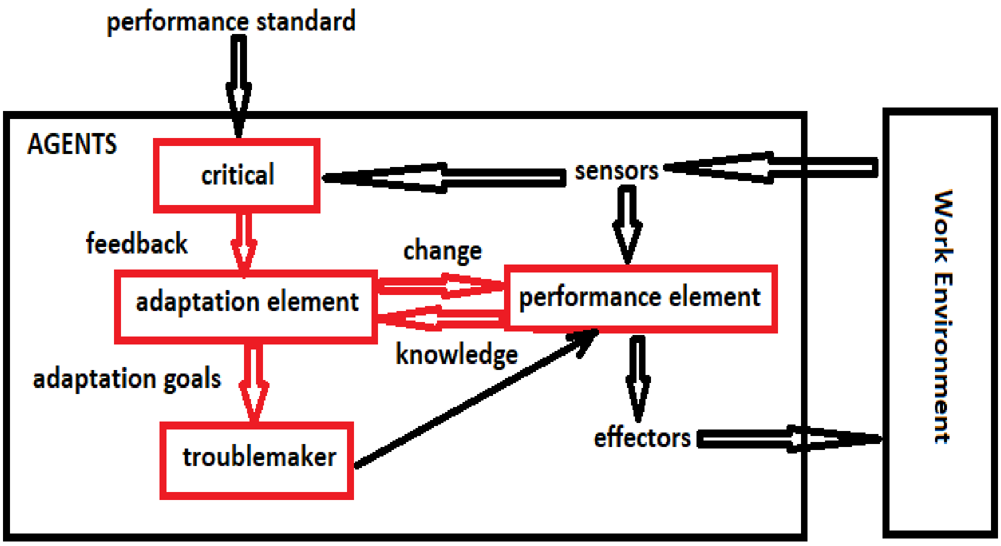


Figure 2. The general model of an adaptive agent

The global objective of the system is to facilitate the successful conclusion of as many transactions as possible. The main advantage of an electronic system based on agents is determined by the low transaction costs for both the seller and the buyer. One way to test the degree of autonomy of the agents in a dynamic business environment, such as e-commerce, is the inclusion of an adaptation capacity in the architecture of the agents. Next, we refer to the negotiation process of one or more attributes of a product and we will show two ways of adapting the agents to the negotiation process.

Negotiation is a general process in which two or more agents can be involved to determine a contract accepted by all agents if certain conditions are met. Also, the process involves continuous communication between the agents until the moment the conditions are accepted by the agents or until the deadline for concluding the negotiation is reached. Between the agents there is an exchange of proposals that refer to certain attributes of the traded products, proposals that are values of these attributes, automatically generated by the negotiation strategy of the agent making the proposal. The basic element of the multi-agent negotiation process is determined by the negotiations and the negotiation strategies associated with the intelligent agents involved in the negotiations. An agent's negotiation strategy determines how he will act within the protocol, trying to obtain the greatest possible benefit. It is based on a model of the user's economic preferences and the behavioral strategies of the parties involved. Making decision is based on the consideration of several attributes of a product. For example, the evaluation of a product by the buyer will depend on several attributes, the capacity of the product, its type and the price proposed by the seller.

The utility function (expected utility) of a user, U, applied to an action a with e status attributes is defined by relation (1):

$$U(a,e) = \sum_{i=e'}^{n} prob(ex(a,e) = e') * utility(e') \quad (1)$$

where $e' \in en\gamma(e,a)$ represents the weight of the attribute e within the global evaluation of the action a.
The multi-agent modeling is represented in the figure 3.

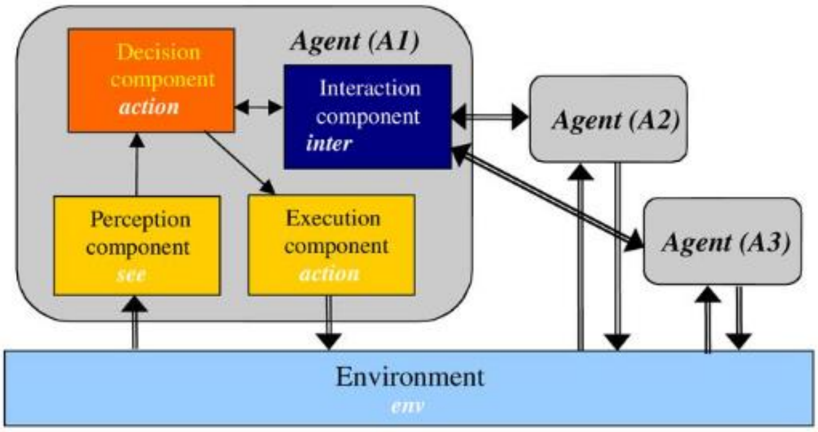


Figure 3. Multi-agent modeling.

For more agents we have: $I = \{i_1, \dots\dots, i, \dots.\}$ and results:

$$see: E \to P; env: ExA_1x \dots.A_n \to P(E); inter: P \to I; action: PxI \to A$$

$$utility(e'): Val(a_i) \to A \quad (2)$$

Where $Val(a_i)$ represents the range of values for the attribute a_i.
In the case of selling computers, the form of the function $utility(e')$ for the price is determined by equation (3):

$$utilitye'(x) = 1 - x \quad (3)$$

The value of attribute x is normalized in the interval [0, 1].
A first way of implementing the ability of agents to adapt to the negotiation process is determined by genetic algorithms. We will briefly present this method. The goal is to assign the agents the ability to learn the utility function of the agents they negotiate with whom he negotiates.
By applying the three operators (selection, crossover, and mutation), populations with better and better individuals are generated because of the search method modeled with the three operators. The role of the crossover operator is to recombine the existing genetic material in new ways, and the role of the mutation operator is to introduce new genetic material into the population through random changes.
Another way to implement the adaptability of agents is the use of neural networks. The role of artificial neural networks is to model the negotiation strategy of the other agent, in the case of bilateral negotiations. The use of this learning method allows increasing the number of completed transactions as well as making the most advantageous transactions for the agent equipped with learning capacity [20].
The architecture of the feed-forward associative neural network used to achieve the adaptation capacity of an intelligent agent to the negotiation process is presented in the figure 4.

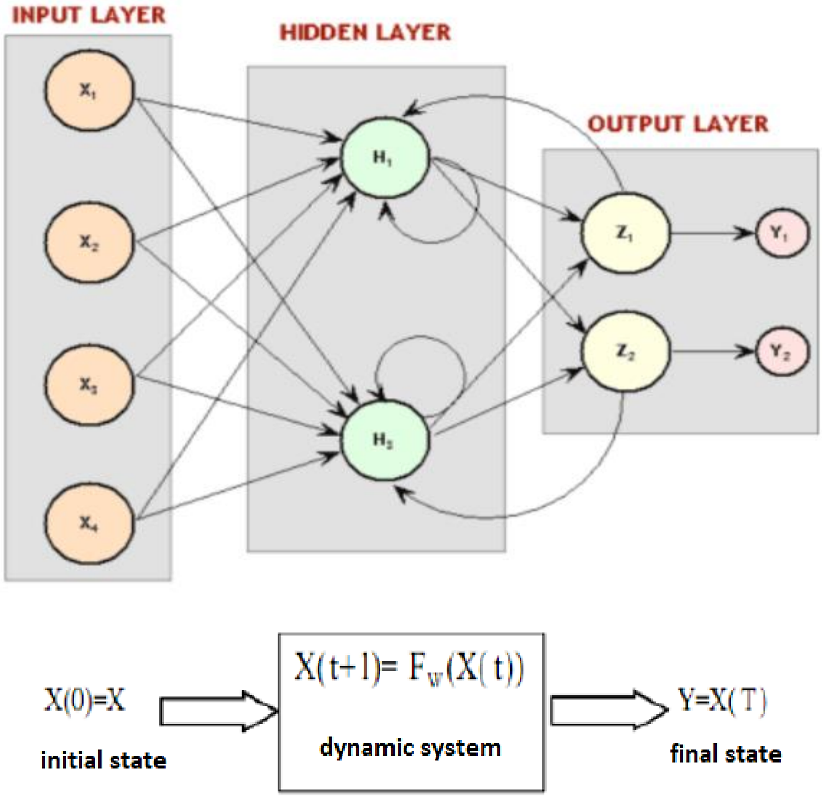


Figure 4. Architecture of the feed-forward neuronal network.

Figure 5 shows the results obtained for the decentralized random training of the neural network.

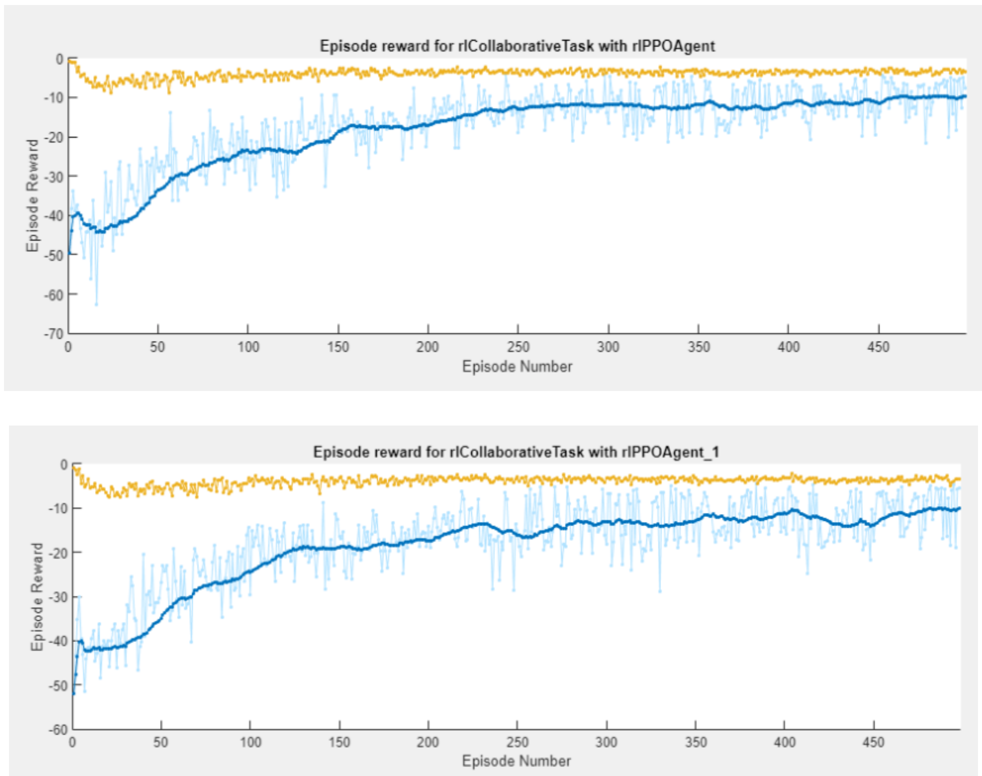
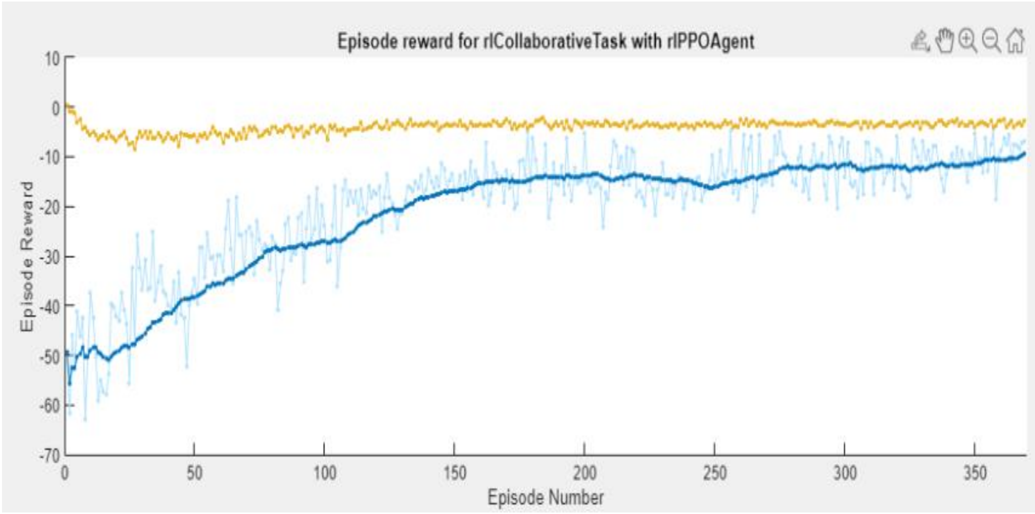


Figure 5. Results obtained for the decentralized random training of the neural network.

Figure 6 shows the results obtained for the centralized random training of the neural network.



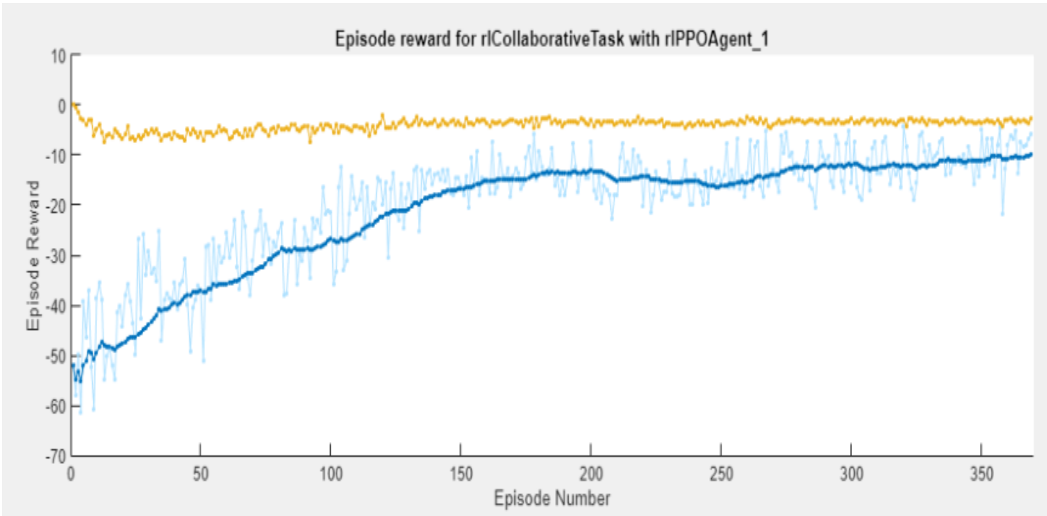


Figure 6. Results obtained for the centralized random training of the neural network.

Based on the estimated value of the offer made by the selling agent at time t+1 for the selling attribute, the buying agent can become more flexible to speed up the successful conclusion of the negotiation. Thus, the buyer's agent can adjust his negotiation strategy to the estimated negotiation strategy of the seller's agent with the help of the neural network [21]. The following figure shows the training results for the two agents to work in collaboration.

The field of observation defines what information the agents must provide at each moment of time. Agents use the field of observation to decide the next action, based on the policy learned and reinforced by the rewards received from previous actions [22].

For a sell transaction, the constraint is to have zero ending inventory, assuming successful liquidation of all stocks. For a buy transaction, the constraint is to have a complete ending inventory, assuming the successful purchase of all shares [23]. Under this ending inventory constraint, the implementation gap is calculated as follows:

- for a sell transaction: Implementation Shorfall=Arrival Price× Traded Volume−Executed Price× Traded Volume

- for a buy transaction: Implementation Shortfall=Executed Price× Traded Volume−Arrival Price× Traded Volume

The arrival price is the first bid or ask price observed at the beginning of the trading horizon. If the implementation gap is positive, it means that the average executed price was weaker than the arrival price, while a negative implementation gap implies that the average executed price was better than the arrival price. The cumulative implementation shortfall from the beginning of the trading horizon to the previous step reflects the trading cost incurred by the agent in the past and serves as the third observation variable.
The last observation variable is the divergence of the current price from the arrival price. This variable reflects the current state of the market and is measured as the difference between the arrival price and the average price of the first two levels of the current limit order (LOB) book.

- for a sell transaction: Price Divergence=Average Bid Prices of First 2 Levels of Current LOB−Arrival Price
- for a buy transaction: Price Divergence=Arrival Price−Average Ask Prices of First 2 Levels of Current LOB

A positive price divergence implies current trading conditions more favorable than the arrival price, and a negative divergence less favorable condition.

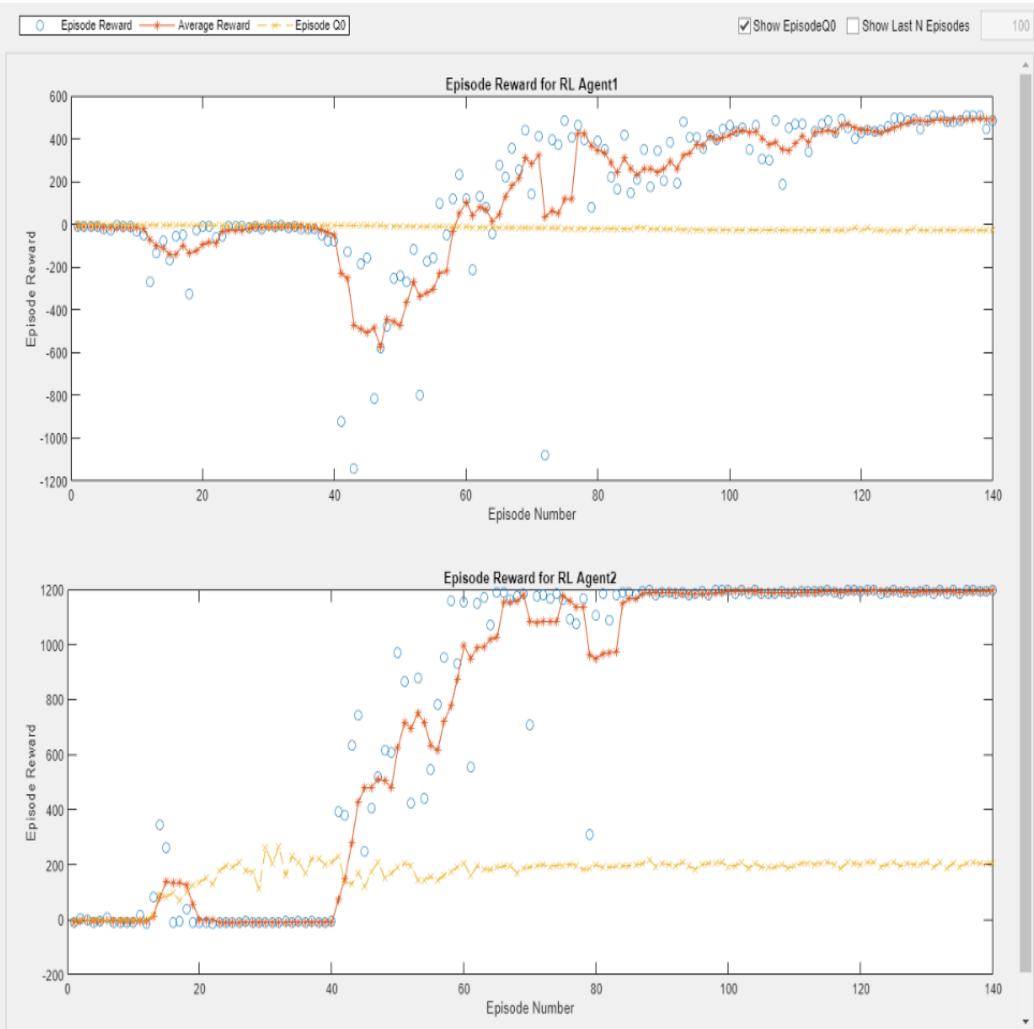


Figure 7. Training results for the two agents to work in collaboration.

Like the buyer agent, the seller agent can use the ANN-FF to estimate the proposals made by the buyer agent. Since the training set of the neural network is extracted online, there is a time interval at the beginning of the negotiation in which the learning ability is not applied. The obtained experimental results revealed an improvement in the number of successfully concluded transactions in the process of negotiating a single attribute of a process (for example, the price). The experiments carried out included the trading of second-hand products, calculators, mobile phones, in an agent-based electronic commerce system simulated using the MATLAB platform.

4. Conclusions

The advent of artificial intelligence (AI) and predictive analytics has opened exciting possibilities for e-commerce sites to better understand their customers and optimize their operations. These technologies can be used to better understand customer behavior and preferences, anticipate customer needs, and generate insights to drive more effective marketing and sales strategies [24]. One of the most common applications of AI and Predictive Analytics in e-commerce is the use of predictive models to identify customer segments and personalize product recommendations. Using customer data such as previous purchases, browsing history and engagement levels, these models can identify distinct customer segments and provide personalized product suggestions that are tailored to each individual customer [25]. This can help increase customer engagement and loyalty as well as increase sales. Another area where AI and Predictive Analytics can be leveraged is in price optimization. By analyzing customer data such as past purchases, market trends, and competitor pricing, predictive models can be used to determine optimal prices for products and services. This can help e-commerce sites maximize profits while offering competitive prices that attract customers.

The results obtained in the implementation of the multi-agent automatic trading system presented in the paper, demonstrate unequivocally that agent-based architectures can be the basis for the creation of reliable applications in an extremely dynamic field such as electronic commerce for financial transactions. Intelligent agents can acquire knowledge through problem-solving processes. The study of the social behavior of agents in science cognitive is an important part of the field of intelligent agents. Software agents focus on interactions and collaborations to achieve their goals in a context that changes in an often-unpredictable manner. The characteristics that suggest the use of agents in complex systems are primarily adaptability, autonomy, and collaboration. Autonomy is a distinctive property of a agent and presupposes the agent's ability to survive in a changing environment. An agent can perceive environmental factors and make decisions about how to react accordingly. Adaptability presupposes the existence of the learning capacities necessary for the agent to adapt decisions according to experience. The collaboration between agents means that they can be designed to address the various aspects of solving a problem, each agent being designed in the most suitable paradigm to solve its part of the overall tasks of the system. Coordinating the behaviors of independent agents is also a central part of multi-agent system design. Agents can efficiently process local data and communicate with other agents when necessary and when the tasks they face are beyond their domain of knowledge. Multi-agent systems are used in a wide range of applications, such as e-commerce, e-learning, communication, data mining, simulation, robotics, transport systems and grid computing.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, X.X. and Y.Y.; methodology, X.X.; software, X.X.; validation, X.X., Y.Y. and Z.Z.; formal analysis, X.X.; investigation, X.X.; resources, X.X.; data curation, X.X.; writing—original draft preparation, X.X.; writing—review and editing, X.X.; visualization, X.X.; supervision, X.X.; project administration, X.X.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript." Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding

Data Availability Statement: Not applicable.

Acknowledgments: This work is supported by The Intermediate Organization for the Promotion of the Information Society (OIPSI), by the National Plan of R&D, Project NEUROGUV, cod: POC/875/2/2, Contract No. 142429/01.06.2022.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, Y.; Li, N. Legal regulation on false publicity of live streaming e-commerce. *Intellect. Prop.* **2021**, *5*, 68–82.
2. Zong, K.; Yuan, Y.; Montenegro-Marin, C.E.; Kadry, S.N. Or-based intelligent decision support system for e-commerce. *J. Theor. Appl. Electron. Commer. Res.* **2021**, *16*, 1150–1164.
3. Khemakhem, F.; Ellouzi, H.; Ltifi, H.; Ayed, M.B. Agent-based intelligent decision support systems: A systematic review. *IEEE Trans. Cogn. Dev. Syst.* **2020**, *14*, 20–34.
4. Shi, P.; Yan, B. A survey on intelligent control for multiagent systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 161–175.
5. Idczak, A.; Korzeniewski, J. New algorithm for determining the number of features for the effective sentiment-classification of text documents. *Wiadomości Stat.* **2023**, *68*, 40–57. [
6. Rizk, Y.; Awad, M.; Tunstel, E.W. Decision making in multiagent systems: A survey. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *10*, 514–529.
7. Suzor, N.; Van Geelen, T.; Myers West, S. Evaluating the legitimacy of platform governance: A review of research and a shared research agenda. *Int. Commun. Gaz.* **2018**, *80*, 385–400.
8. Bertot, J.C.; Jaeger, P.T.; Munson, S.; Glaisyer, T. Social media technology and government transparency. *Computer* **2010**, *43*, 53–59.
9. Mattson, G.A. The promise of citizen coproduction: Some persistent issues. *Public Product. Rev.* **1986**, *10*, 51–56.
10. Zajko, M. Telecom responsibilization: Internet governance, surveillance, and new roles for intermediaries. *Can. J. Commun.* **2016**, *41*, 75.
11. Gorwa, R. What is platform governance? Information. *Commun. Soc.* **2019**, *22*, 854–871.
12. Hagiu, A.; Wright, J. Multi-sided platforms. *Int. J. Ind. Organ.* **2015**, *43*, 162–174.
13. Grewal, R.; Chakravarty, A.; Saini, A. Governance mechanisms in business-to-business electronic markets. *J. Mark.* **2010**, *74*, 45–62.
14. Ojo, A.; Mellouli, S. Deploying governance networks for societal challenges. *Gov. Inf. Q.* **2018**, *35*, S106–S112.
15. Sànchez-Marrè, M. *Intelligent Decision Support Systems*; Springer International Publishing: Cham, Switzerland, 2022; pp. 77–116.
16. Shi, P.; Yan, B. A survey on intelligent control for multiagent systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 161–175.
17. Qasem, M.H.; Hudaib, A.; Obeid, N.; Almaiah, M.A.; Almomani, O.; Al-Khasawneh, A. Multi-agent systems for distributed data mining techniques: An overview. *Big Data Intell. Smart Appl.* **2022**, *994*, 57–92.
18. Chaimontree, S.; Atkinson, K.; Coenen, F. A Multi-agent Based Approach to Clustering: Harnessing the Power of Agents. In Proceedings of the ADMI, Clemson, SC, USA, 14–16 April 2011; Springer: Berlin/Heidelberg, Germany, 2011.
19. Yan, B.; Shi, P.; Lim, C.C.; Shi, Z. Optimal robust formation control for heterogeneous multi-agent systems based on reinforcement learning. *Int. J. Robust Nonlinear Control.* **2022**, *32*, 2683–2704.
20. Haley, J.; Tucker, J.; Nesper, J.; Daniel, B.; Fish, T. Multi-agent collaboration environment simulation. In *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications*; SPIE: Bellingham, WA, USA, 2023; Volume 12529, pp. 197–202.
21. Albashiri, K.A.; Coenen, F.; Leng, P. EMADS: An extendible multi-agent data miner. *Knowl.-Based Syst.* **2009**, *22*, 523–528.

22. Tozicka, J.; Rovatsos, M.; Pechoucek, M. A framework for agent-based distributed machine learning and data mining. In Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, Honolulu, HI, USA, 14–18 May 2007; ACM: New York, NY, USA, 2007.
23. Cao, L.; Weiss, G.; Philip, S.Y. A brief introduction to agent mining. *Auton. Agents Multi-Agent Syst.* **2012**, *25*, 419–424.
24. Rao, V.S. Multi agent-based distributed data mining: An overview. *Int. J. Rev. Comput.* **2009**, *3*, 83–92.
25. Rabbani, I.M.; Aslam, M.; Enríquez, A.M.M. SAFRank: Multi-agent based approach for internet services selection. *Int. Arab J. Inf. Technol.* **2022**, *19*, 298–306