

Article

An Intelligent Decision Support System Based on Multi Agent Systems for Business Classification Problem

Mais Haj Qasem ¹, Mohammad Aljaidi ^{1,*} , Ghassan Samara ¹ , Raed Alazaidah ¹, Ayoub Alsarhan ²  and Mohammed Alshammari ³ 

¹ Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa 13110, Jordan; m.hajqasem@zu.edu.jo (M.H.Q.); gsamara@zu.edu.jo (G.S.); razaidah@zu.edu.jo (R.A.)

² Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdallah II for Information Technology, The Hashemite University, Zarqa 13116, Jordan; ayoubm@hu.edu.jo

³ Faculty of Computing and Information Technology, Northern Border University, Rafha 91431, Saudi Arabia; mohammed.alshammari@nbu.edu.sa

* Correspondence: mjaidi@zu.edu.jo

Abstract: The development of e-systems has given consumers and businesses access to a plethora of information, which has complicated the process of decision making. Document classification is one of the main decisions that any business adopts in their decision making to categorize documents into groups according to their structure. In this paper, we combined multi-agent systems (MASs), which is one of the IDSS systems, with Bayesian-based classification to filter out the specialization, collaboration, and privacy of distributed business sources to produce an efficient distributed classification system. Bayesian classification made use of MAS to eliminate distributed sources' specialization and privacy. Therefore, incorporating the probabilities of various sources is a practical and swift solution to such a problem, where this method works the same when all the data are merged into a single source. Each intelligent agent can collaborate and ask for help from other intelligent agents in classifying cases that are difficult to classify locally. The results demonstrate that our proposed technique is more accurate than those of the non-communicated classification, where the results proved the ability of the utilized productive distributed classification system.

Keywords: intelligent decision support systems (IDSS); classification; FIPA standards; multi-agent system; Naïve Bayesian



check for updates

Citation: Haj Qasem, M.; Aljaidi, M.; Samara, G.; Alazaidah, R.; Alsarhan, A.; Alshammari, M. An Intelligent Decision Support System Based on Multi Agent Systems for Business Classification Problem. *Sustainability* **2023**, *15*, 10977. <https://doi.org/10.3390/su151410977>

Academic Editors: Shadi Alzu'Bi and Ala Mughaid

Received: 3 June 2023

Revised: 4 July 2023

Accepted: 9 July 2023

Published: 13 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial intelligence (AI) is a broad group of technologies that has received a lot of attention in recent years and promises a lot of benefits for businesses in terms of additional economic value. Following an explosion of data and a significant rise in computing capability, companies have increasingly turned to AI to achieve economic value and design of methodologies based on intelligent algorithms, impacting business and society in recent years [1]. Businesses are increasingly depending on AI and related machine learning models to increase human knowledge of complex systems and automate decision making, which also necessitates continual expert contributions. Intelligent decision support systems (IDSS) are the most important subset in any business because they can help in making business model design decisions; therefore, businesses exploit AI because they must examine their business DSS to remain competitive in a rapidly changing business environment [2].

DSS is computerized software that is used to assist decisions, judgments, and courses of action in an organization or corporation. A DSS sifts through and analyzes vast amounts of data, producing comprehensive information that may be utilized to solve problems and make decisions. Therefore, the combination of AI with DSS, called IDSS, assists humans in making better decisions by providing relevant information and recommendations [3].

IDSS collects and analyzes data from different sources to create meaningful insights for analysts. Artificial intelligence decision support systems then make recommendations and communicate them to users in an understandable manner [4]. Most organizations can easily obtain enough relevant data nowadays, but knowing how to analyze it and what to do with the resultant insights has become the most difficult and time-consuming task. One of the problems that may face any business is knowing how to get the right document classification due to a shortage of data.

Document classification categorizes documents into groups according to their structure, content, or additional characteristics. Businesses may effectively handle enormous numbers of documents, gain insightful information, and automate decision-making processes by classifying the documents [5]. The ability to quickly identify and retrieve accurate, pertinent, and current information is another benefit of categorization documents. As a result, it is simpler to filter and direct documents to users, as well as find relevant information at the appropriate time.

To solve the problem that may face any business in document classification and get better results, we found that collaboration in document classification should be wisely and intelligently exploited between different organizations. Therefore, we applied one of the AI expert system technologies called multi-agent systems (MAS). MAS allows the creation of autonomous agents with specific issue-solving capabilities. These agents, which have communication and coordination abilities, can cooperate in shared surroundings [6]. Essentially, an MAS constitutes a collection of specialized agents that operate within well-defined boundaries and work towards a shared objective by interacting with one another [7]. These agents are distinguished by their flexibility and proactive behavior, as well as their ability to communicate and cooperate with other agents to achieve their goals [8].

The naive Bayesian algorithm was chosen in our proposed approach because it is one of the most basic, but effective, classification algorithms now in use and has applications in many industries. Naive Bayesian, also known as a probabilistic classifier, is used for solving classification problems and is one of the most effective and simple algorithms. It is known to outperform even highly sophisticated classification methods that work on Bayes' theorem of probability to predict the class of an unknown instance [9]. Naive Bayesian class probability is dependent on the frequency or random analysis of each class. Calculating conditional probabilities is based on the ratio between the presence of the class itself and the presence of the characteristics and classes together. In general, classification, and particularly naive Bayesian classification, perform poorly when there are few samples available to represent a given class [9]. Although there are several robust classifiers available in the literature, the naive Bayesian classifier is chosen for integration with distributed agents because of the characteristics that make it appropriate for the proposed distributed classification task.

Therefore, we proposed a productive distributed classification system that combined MAS with a Bayesian-based approach, where our proposed system consists of a set of agents and a communication protocol, each of which presents its own stored data, calculation device, and communication ability, and each of those agents has its own data and carries out its processing task independently. When one of the agents in the system must classify new data items, it can decide to take its result independently, without any collaboration, only if the probability of the result is above a specific threshold, and if the result is below this threshold, it needs to collaborate with other agents in the system and send them a request to classify that instance.

The other agents send their class classification result, probabilities, and conditional probabilities to the requested agent, which can make the right decision to take the other agent's result, or not, in feedback processing. The contacted result shows the benefits of combining the probabilities and conditional probabilities of many different sources; that is one approach that we implement in our system, where this technique functions precisely the same when all data is combined in a single source [10]. However, this

process eliminates distributed sources' specialization and privacy, which are crucial in many economic sustainability applications.

This paper is organized as follows. Section 2 reviews related work, Section 3 presents the proposed approach, Section 4 presents the result, and Section 5 gives the conclusion.

2. Literature Reviews

In this section, the recent approaches in this area will be discussed, where we discuss the research in MAS in business and the MAS in data mining techniques in general to show the importance of each of them and the encouragement of their use in proposing our approach.

In business, ref. [11] reviewed the supply chain management application that used the development of MAS to analyze various business issues. In addition, they are modeling the supply chain by examining generic frameworks.

In [12], they introduced the sustainable concept for the combination of MAS with the management of the supply chain. This study aims to socially implement sustainable supply chains by giving an example of how different plans might operate simultaneously in a supply chain.

In [13], they reviewed MAS definitions, characteristics, uses, problems, and communications. This study looks at the classification of MAS applications and challenges, as well as the definitions, characteristics, applications, problems, communications, and assessment of MAS. It also includes research citations. The information in this publication ought to be useful for MAS researchers and practitioners. Energy management, energy marketing, pricing, scheduling, reliability, network security, fault handling capacity, agent-to-agent communication, SG-electrical automobiles, SG-building energy systems, and soft grids have all been studied in relation to MAS in operating smart grids. A review, classification, and compilation of more than 100 MAS-based smart grid control articles have been completed.

In [14], they examined the bagging and boosting procedures, which enable several classifiers to work on various sets of data in classification and are a good example of such a method. By averaging the local models of the distributed sources to produce a global model, Bayesian classifiers are used in distributed environments.

In clustering, ref. [15] proposed a distributed clustering method that uses dense clustering algorithms and runs locally, called density-based distributed clustering (DBDS). The decision-making center recalculates the cluster centers based on the received centers and elements from the local cluster centers that are created at each dispersed source using a minimal amount of data elements. Similar distributed clustering strategies without element transformation were put forth in [16]. In single learning models, Java agents for meta-learning (JAM) [17] and BODHI [18] were proposed for this purpose, where each source shares its learned model with the global model to produce a single learning model for mining the input data.

Recently, various MAS-aware DDM approaches were proposed, where researchers benefit from using MAS in implementing, controlling, and organizing distributed sources. In [19], an extensible multi-agent data miner (EEMADS) was presented as MAS-based on an ensemble classification, which supplies weights for each distributed classifier or chosen ones to carry out the classification task based on understanding the learning model at each classifier. MAS-based distributed categorization with a variety of result integration techniques was offered as an abstract architecture [20,21]. Ref. [22] provides a succinct summary of various strategies.

To improve classification outcomes and enable indexers to work together, semiautomatic distributed document categorization has been proposed [23]. Mutual cooperation has been implemented within this framework, and human interference is required to assess the appropriateness of the information that has been shared. Recently, an MAS-based clustering framework that can enhance the initial cluster centers at each agent was also suggested [24]. The results of the proposed collaborative clustering showed an improvement over noncollaborative agent-based clustering.

To obtain optimal collision-free time-varying forms, a robust control technique based on reinforcement learning (RL) is proposed for uncertain, heterogeneous MAS [25]. To estimate the reference and disturbance systems for states and dynamics, they created a completely distributed adaptive observer, and simulations were used to confirm its efficacy and resilience. Table 1 summarizes the advantages of the discussed literature.

Table 1. Summary of The Literature Reviews.

References	Application	Techniques	Collaboration	MAS-Based
[11]	Supply Chain Network	Reviewed Paper		✓
[12]	Supply Chain Management	Linear Physical Programming	✓	✓
[13]	Smart Grid	Reviewed Paper		✓
[14]	Alarm Network	Naïve Bayes		
[15]	Distributed Knowledge Discovery	Clustering	✓	
[16]	Network	Clustering	✓	
[17]	Banks	Classification	✓	
[18]	Customer Data Management	Classification	✓	
[19]	Knowledge Discovery in Databases	Ensemble Classification	✓	✓
[20]	Several Application	Reviewed Paper		✓
[21]	Several Application	Reviewed Paper		✓
[22]	Several Application	Reviewed Paper	✓	✓
[23]	Document Retrieval	K-Nearest Neighbors	✓	✓
[24]	Network	Clustering	✓	✓
[25]	Vehicle	Reinforcement Learning	✓	✓

3. Proposed System

Our proposed system consists of a group of agents and a communication protocol, each of which presents its own stored data, calculation device, and communication capability. Each agent also has its own data and performs its processing task independently. We proposed a productive distributed classification system that combined MAS with a Bayesian-based approach. When one of the system's agents must classify a new instance, it can decide to do so independently and without collaborating with any other agents—but only if the probability of the result exceeds a certain threshold. Otherwise, it must work together with other agents in the system and submit a request to categorize the instance.

The other agents provide the requested agent with their class classification results, probabilities, and conditional probabilities so that the requested agent may decide whether or not to use the other agents' results while processing feedback.

Class probability in a typical classification task is dependent on the frequency or random analysis of each class. Calculating conditional probabilities is based on the ratio between the joint presence of the attributes and the classes to the presence of the class itself. In general, classification and, particularly, Bayesian classification perform poorly when there are few samples available to represent a given class [9].

Among the solutions, merging the probabilities and conditional probabilities of many different sources is one approach that is simple to implement; this technique functions precisely the same when all data is combined in a single source [15]. However, this process eliminates distributed sources' specialization and privacy, which are crucial in many applications, such as illness diagnosis based on the geographic location of the data gathered [22,26]. Therefore, the proposed approach is implemented in a multi-agent system (MAS) that is a mechanism for creating goal-oriented autonomous agents in a shared environment with coordination facilities and communication. An agent is a software unit that implements tasks in changing the configuration environment using its states and

behavior components. Distributed system configuration (security and scalability) changes over time without a centralized controller of the implemented data transactions [23]. MAS consists of a set of agents and communication protocol, each of which presents its own stored data, calculation device, and communication ability, as illustrated in Figure 1.

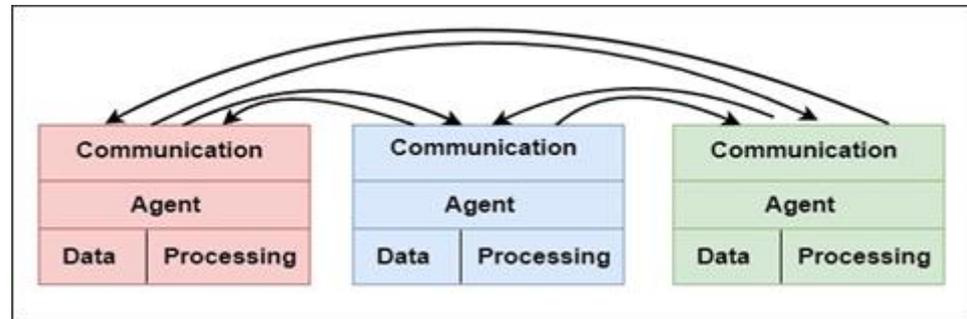


Figure 1. The Proposed Collaborative Environment for Classification.

Agent communication language (ACL) messages that contain requests for or replies to types of information are used by agents to communicate with one another utilizing the Foundation for Intelligent Physical Agents (FIPA) query interaction protocol (IP) and ACL.

One of the FIPA [24,25] protocols that is used by several multi-agent developing environments is Query IP [27]. The IP is a collection of specifications that let one agent, known as the initiator, seek the assistance of other agents to complete tasks [24]. The message type, message structure, and message exchanging states and circumstances are all covered by the specifications. The types of messages that may be utilized include requests and calls, which are described by the ACL [28]. The ACL also describes the message's structure, while the conditions for message exchange are specified by the protocol itself.

In the proposed approach, an agent is characterized by a set of behaviors that is triggered based on a set of states. The behavior of each agent is described as follows:

- Model building behavior is started by system execution. The agent creates its training model at this stage by computing the probabilities of the classes involved, the probabilities of the attribute values involved, and the conditional probabilities of each attribute with each class. These values are considered the trained model for each system. The agent in model building behavior allows no input request from other agents.
- Processing behavior is triggered when an input set of attribute values is presented to the agent. In this stage, the predicted value is calculated by the agent and decides whether to request collaboration from other agents or not.
- Query reference call is initiated by the initiator to all participant agents in the system, with the set of attributes to be classified.
- Participant response, that is, refusal or agreement, depends on the presence of calculated probabilities above the threshold.
- Participant results are sent to the initiator with the attached probabilities, or failure response is sent if a failure occurs with the initiator. The agent's behavior is illustrated in Figure 2. Each agent's states and behavior are shown in Figure 3.

The agent functionalities are specified using predicates to formalize the agent behaviors. Agent's functionalities are triggered with the presence of some conditions; these are: training data, training model, other agents, query, result for a query either satisfactory results or unsatisfactory that can be represented by exist and satisfied predicates with various subjects as given in the following:

exist (training-Data, agent)
exist (others, agent)
exist (results, query, agent)

exist (query, agent)
exist (trained-Model, agent)
satisfied (results, query, agent)

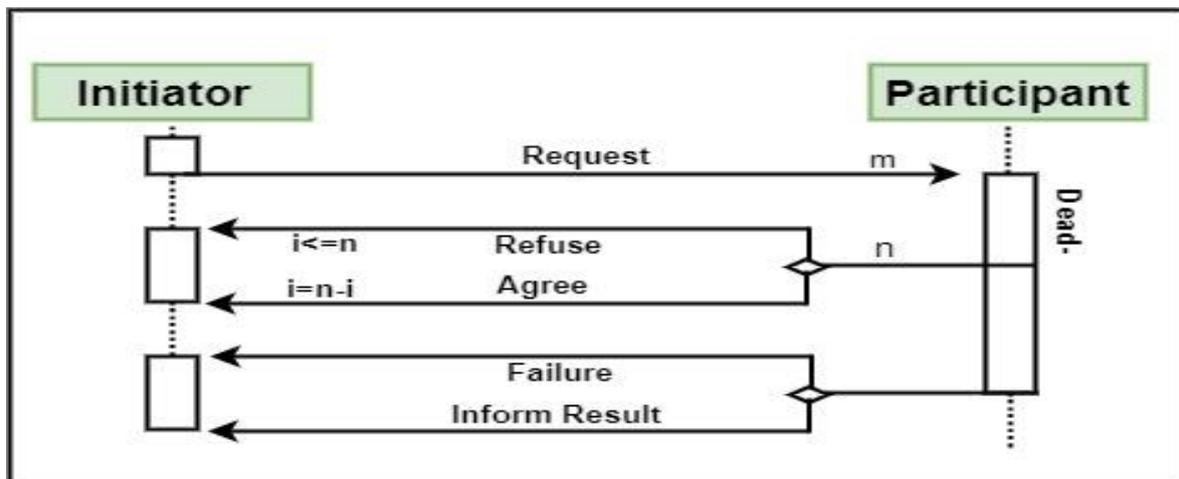


Figure 2. Interaction Protocol.

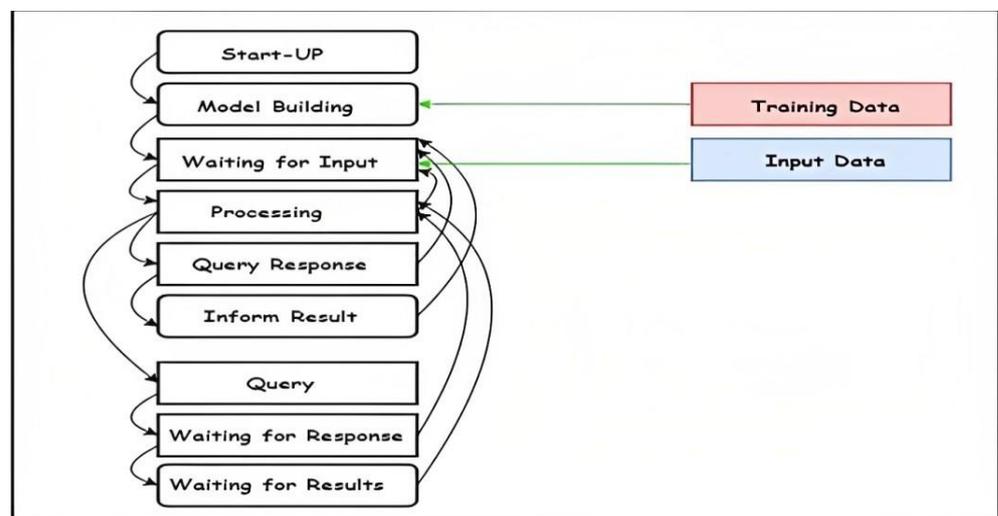


Figure 3. Agent States and Behaviors.

Accordingly, the functionalities of an agent, which are building a model, processing a query, request, and response, can be described as follows:

- model-Building* (training-Data, agent)
- request* (query, agent)
- AcceptandLearn* (results, query)
- processing* (query, agent)
- Response* (results, query, agent)
- Reject* (results, query)

The facts that represent the behavior of the agents are represented as follows:

- $exist(T, self) \wedge \neg exist(M, self) \rightarrow model-Building(T, self)$
- $model-Building(T, self) \rightarrow exist(M, self)$
- $exist(M, self) \wedge exist(Q, A) \rightarrow processing(Q, self)$
- $processing(Q, self) \rightarrow exist(R, Q, self)$
- $processing(Q, other) \rightarrow response(R, Q, other)$
- $exist(R, Q, self) \wedge \neg satisfied(R, Q, self) \wedge exist(other, self) \rightarrow request(Q, others)$
- $request(Q, self) \wedge response(R, Q, others) \wedge satisfied(R, Q, self) \rightarrow accept-and-Learn(R, Q)$
- $send(Q, others) \wedge received(results, Q, others) \wedge \neg satisfied(results, Q, self) \rightarrow reject(results, Q, others)$

3.1. Naive Bayesian Calculation

Although there are several robust classifiers available in the literature, the Bayesian classifier is chosen for integration with distributed agents because of the characteristics that

make it appropriate for the proposed distributed classification task. The main reasons for choosing the Bayesian classifier described as follows:

1. In our approach, each agent should be able to incorporate information gained from other agents, and Bayesian, as a statistical classifier, should be able to integrate knowledge from other sources besides the training data.
2. Because missing values will be progressively introduced to the model, each agent should be able to handle them. Bayesian, unlike other classifiers such as SVM, can deal with missing data by averaging across the different values that the attribute may have taken.
3. Each agent should be able to share a part of the trained model without having to share the entire training set. Bayesian models provide the flexibility to share and hide parts of the model components, such as the prior probability, conditional probability, and class probability.
4. In comparison to other classifiers, Bayesian has the scalability that is essential for the whole system to allow for the addition of new classes and features without the need to rebuild models [29].

The Bayesian classification principles remain the same as the proposed approach when sharing joint probabilities among agents. Therefore, distributed naive Bayesian classification using the maximum a posterior (MAP) is implemented for classification tasks with discrete random variables and nominal attributes. MAP is given in following equation:

$$\hat{y} = \operatorname{avgmax}_{k \in \{k_1, k_2, \dots, k_n\}} p(C_k) \prod_{i=1}^n P(X_i | C_k)$$

- $p(C_k)$ = probability of the class k , and
- $p(X_i | C_k)$ = conditional probability of attribute value X_i with class k .
- The $p(C_k)$, and $p(X_i | C_k)$, are calculated using the following equations:

$$p(C_k) = n_k/n$$

$$p(X_i | C_k) = n_{ik}/n$$

The $p(X_i)$ is another important probability that calculated at each agent the following equation:

$$p(X_i) = n_i/n$$

- n represents the overall instances at the agent
- n_k denotes the number of instances with the class value k
- n_i is the number of instances with the value i 's attribute
- n_{ik} illustrates the instances' number with the value i 's attribute and class value k

According to the class and conditional probabilities, MAP accordingly selects the class with the highest posterior. Prior to any classification or information sharing process, these probabilities are calculated at each agent during model building.

3.2. Naive Bayesian Combined with Multi-Agent System

When an agent is the proposed approach, that called initiator receives a new instance to classify; first it calculates the probability of each class based on its model. Then it decides to take its result independently without any collaboration only if the probability that calculated by MAP of the result is above a specific threshold " t ", which indicate that this agent model is considered sufficient to classify the instance without any collaboration.

In another case, if the probability that calculated by MAP is below " t ", then the initiator agent must have to collaborate with other agents in the system and send them a request to classify that instance. This request is then followed by a series of calculations, as shown

in Figure 4. Then, the system's other agents will forward the probabilities below to the requested agent only if their result above the same a specific "t":

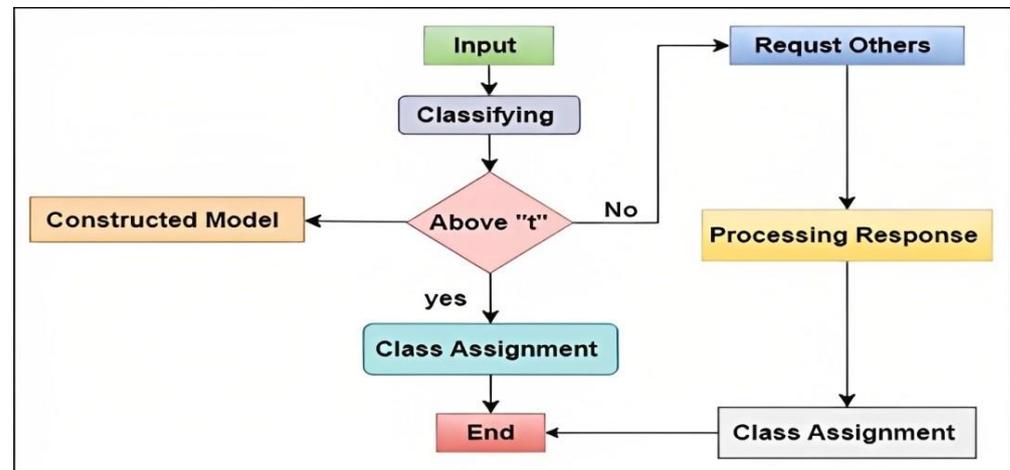


Figure 4. Collaborative Classification Framework.

- Class labels C_k
- $\prod_{i=1}^n P(X_i|C_k)$; and
- Joint probabilities ($\prod_{i=1}^n P(X_i)$)

When the initiator agent receives that information from another agent, it compares the most remarkable knowledge about the instance and the given class that is $\prod_{i=1}^n P(X_i|C_k) \times \prod_{i=1}^n P(X_i)$, which indicates the most desirable combined probability.

4. Results and Discussion

In this section, we will discuss the results of our proposed system and the discussion related to these results. To demonstrate the efficiency and robustness of our introduced system and evaluate the performance of the involved collaborative classification, an MAS for the classification task is improved using JADE. The dataset is obtained from the UCI repository; these datasets represent instances for multiclass document classification [30,31]. The dataset consists of 2463 instances, with a total of 2000 different words. This dataset is first processed by extracting words and subsequently calculating the frequency of each word. Afterward, the resulting data are discretized by converting numbers in the feature vector into discrete values. Feature selection is applied, which results in a feature vector of 50 features instead of 2000. Accordingly, the resulting data are distributed among the agents; each agent is given training and testing sets.

First, the system is implemented to demonstrate the sequence of the process and the communication procedure. Results are obtained as a set of interaction scenarios with two and three agents, as in the following scenarios:

1. First scenario: the threshold is set to a moderate value, where each agent requests help from other agents in the system only if the probability is under 60%, as illustrated in Figure 5.
2. Second scenario: the threshold is set to a low value, where each agent requests help from other agents in the system only if the probability is under 90%, as illustrated in Figure 6.
3. Third scenario: the threshold is set to a high value, where each agent requests help from other agents in the system only if the probability is under 20%, as illustrated in Figure 7.

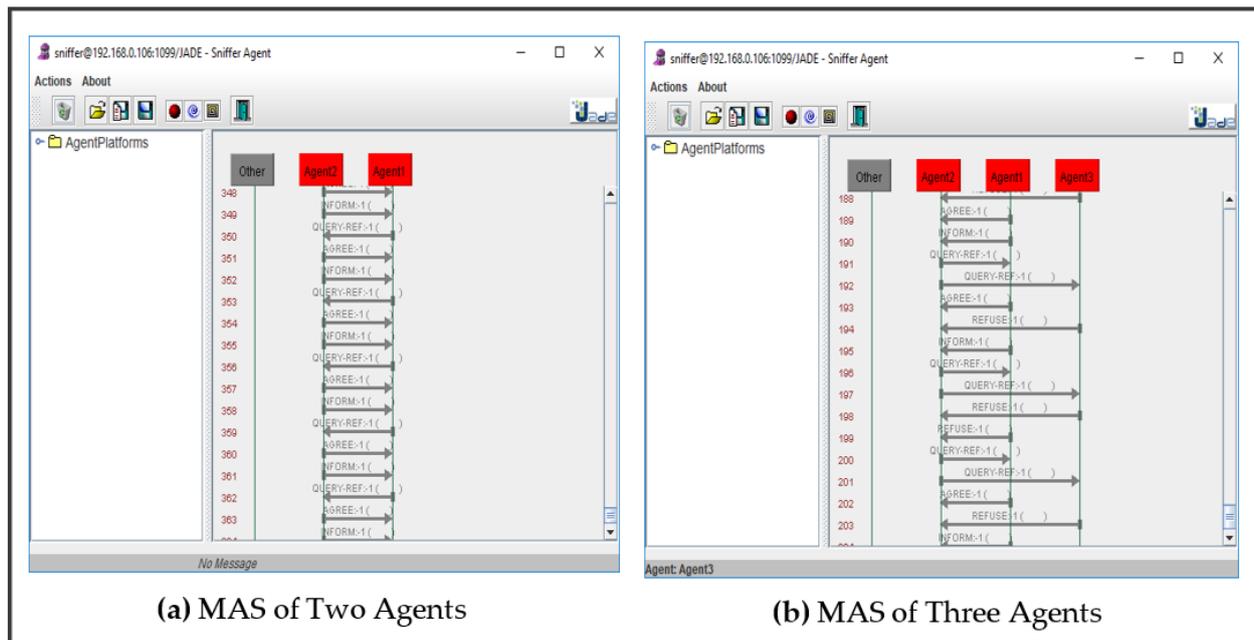


Figure 5. Communication with Moderate Threshold Value.

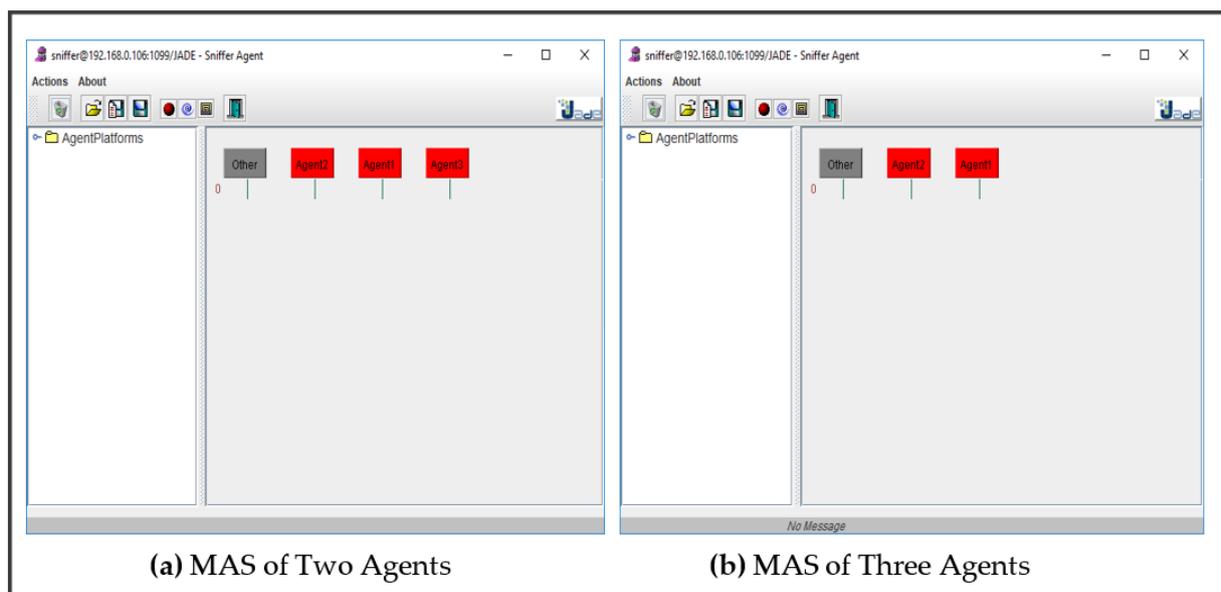


Figure 6. Communication with Low Threshold Value.

In Figure 5, agent communications are limited to some cases that require agent collaboration. The total number of message exchanges among agents in Figure 5a with two agents and Figure 5b with three agents is 364 and 204, respectively. Notably, the exact number of messages is insignificant because it depends on the data allocated to each agent and distributed among these agents, where each of the agents in that case is considered sufficient to classify the instance without any collaboration by using their model in most cases.

In Figure 6, agent communications are not presented because each agent can be considered sufficient to obtain a result for any case, which is expected to be low for some cases that are uncommon to that agent. Nonetheless, given that the threshold is considerably low, the probability obtained is satisfactory for each agent. No message exchange is observed among agents in Figure 6a with two agents and Figure 6b with three agents.

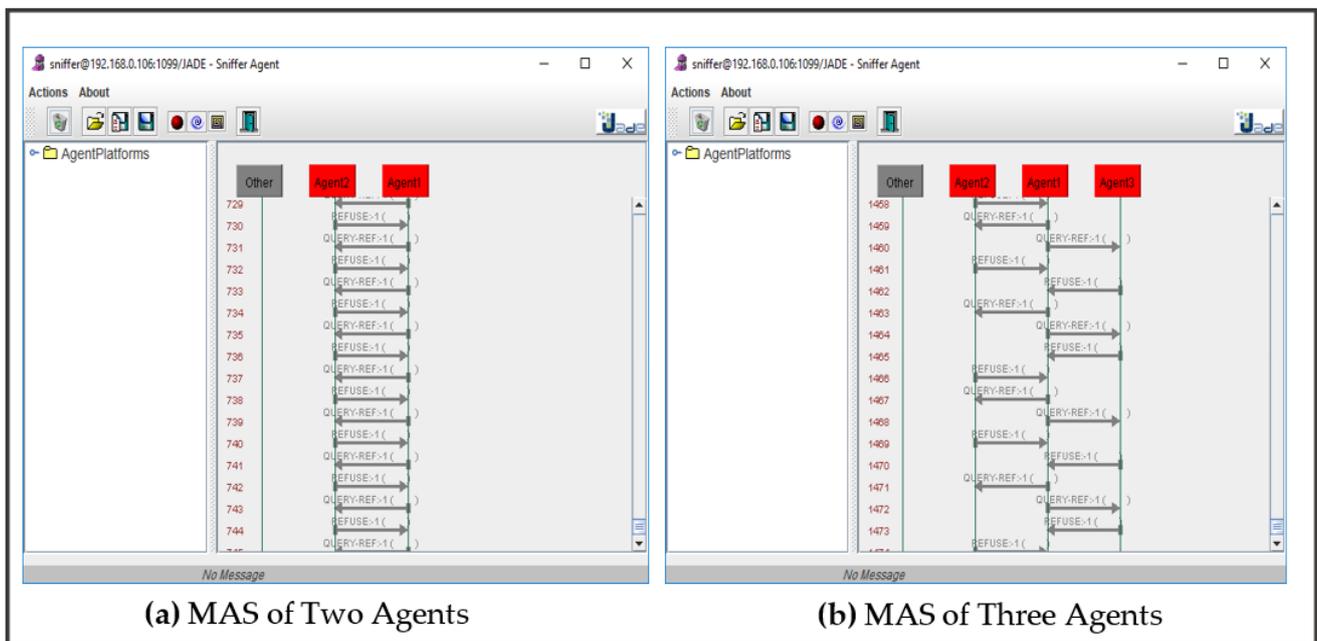


Figure 7. Communication with High Threshold Value.

In Figure 7, agent communications are decidedly required. The total number of message exchanges among agents in Figure 7a with two agents and Figure 7b with three agents is 745 and 1474, respectively. In that scenario, the model of each agent is poor in most cases for classifying the class and needs significant help and collaboration with other agents in the system.

Second, the same dataset is used with a specific partitioning scheme in which the dataset is divided into two partitions based on the class labels as follows:

1. Non-overlapping sets: data from the same case may be included in both columns being evaluated since the categories of a multiple answer variable are used to create the columns of a table.

Each dataset is assigned to an agent and categorized into testing and training sets. Remarkably, the data distributed among agents shares much in common because attribute values are not considered in the division process. Results from the non-communicated two-classification task are listed in Table 2 and illustrated in Figure 8.

2. Overlapping sets: numerical measurement that assesses the percentage of data points from the treatment phase that are greater than the most extreme value from the baseline phase.

Each dataset is partitioned into training and testing sets. The overlapping amount is determined by percentage (10%, 20%, 30%, and 50%) by several testing scenarios with data plotting and sorting. Results compared with the non-communicated two-classification task are listed in Table 3 and illustrated in Figure 9.

Table 2. Results Comparison of non-overlapped Datasets.

Agents No.	Proposed System Accuracy	Non-Communicated Accuracy
2—Agents	64.2%	57.1%
3—Agents	56.6%	41.3%
4—Agents	61.6%	45.3%
5—Agents	62.3%	43.8%

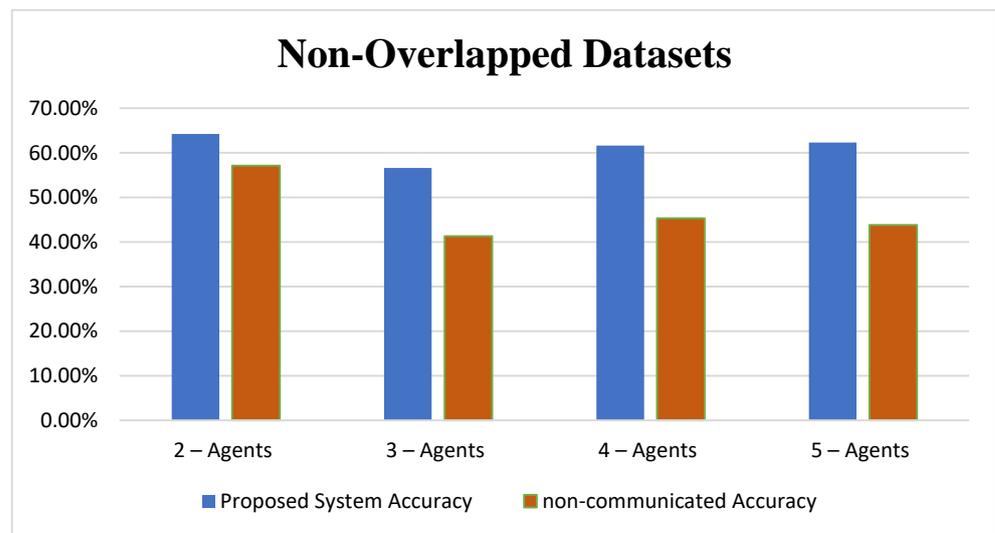


Figure 8. Accuracy Comparison of the Proposed System with Almost Non-Overlapping Data.

Table 3. Results Comparison of Overlapped Datasets.

Approximate	Proposed System Accuracy	Non-Communicated Accuracy
2—Agents		
10%	54.80%	33.90%
20%	56.50%	29.90%
30%	49.70%	26.50%
50%	48.90%	21.90%
3—Agents		
10%	53.90%	40.00%
20%	50.00%	30.80%
30%	50.30%	41.20%
50%	55.00%	35.80%
4—Agents		
10%	52.10%	44.00%
20%	54.70%	42.20%
30%	54.70%	42.20%
50%	54.70%	42.20%
5—Agents		
10%	59.50%	49.40%
20%	53.00%	53.00%
30%	53.00%	53.00%
50%	53.00%	53.00%

However, naturally overlapped data cannot be distributed into non-overlapped sets. Accordingly, these data can be considered minimally overlapped data. This type of partition evaluates how the system performs in easy tasks, where no or little information sharing is required to achieve satisfactory results. Results are calculated as the accuracy of all agents to the total number of instances in the testing set, which is the ratio between the correctly classified instances regardless of the classifier agent. Results showed that the proposed technique is more accurate than that of non-communicated classification.

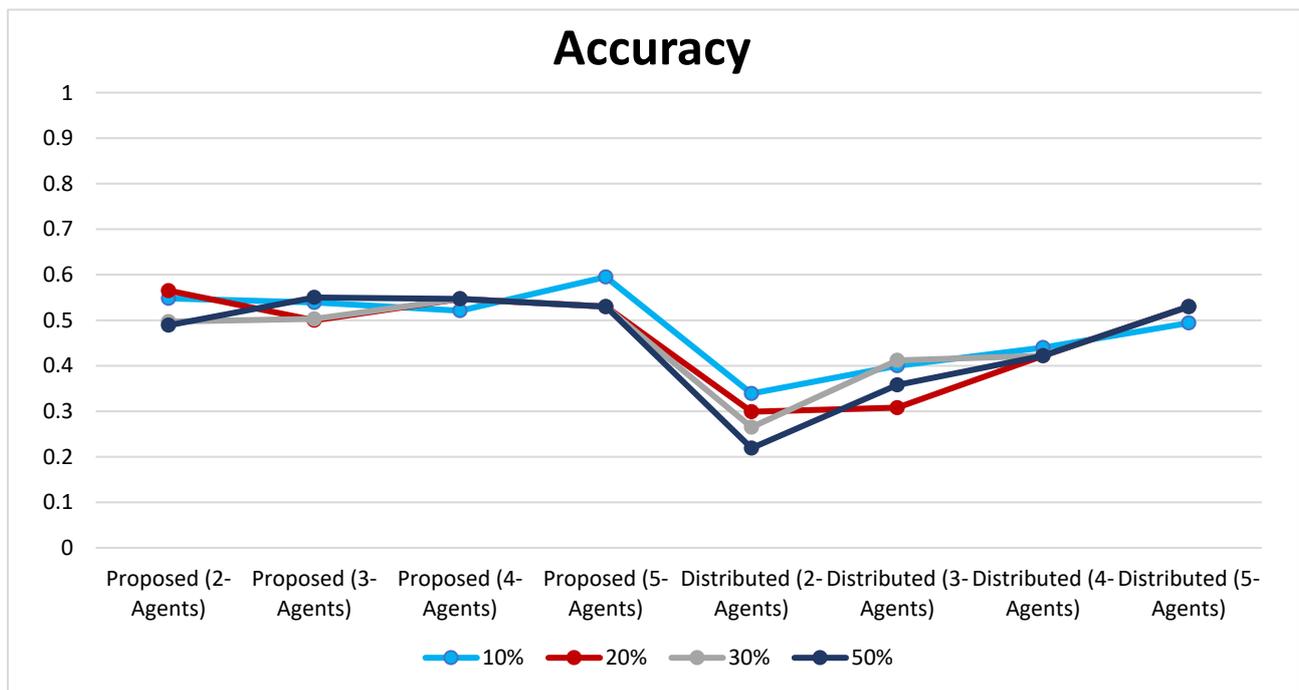


Figure 9. Accuracy Comparison of the Proposed Approach with Overlapped Data.

Results of the previous experiments are like those of the first experiments. The proposed technique is more accurate than the non-communicated classification. Moreover, the distributed classification results are reduced when the overlap among data is increased. This result is because many instances that are uncommon to an agent are added to its datasets due to the overlapping process. Overall, the results proved that the proposed technique can be accurately used in a collaborative environment.

5. Conclusions and Future Work

A productive distributed classification system that combined naive Bayesian classification with a multi-agent system was developed. In our proposed strategy, each agent stores its samples and bases its prior assumptions on these facts. In the classification of new data items, each agent can decide to take its result separately, without any collaboration, only if the probability that was calculated by MAP of the result is above a specific threshold. If the result is below this threshold, they need to collaborate with other agents in the system and send them a request to classify that instance. Each request agent can make the decision to take the other agent's result or not in feedback processing.

The experiments were conducted using various forms of overlapped and non-overlapped data that are generated from a document classification dataset. Each subset of data was assigned to an agent and divided into testing and training sets. The accuracy of all agents was calculated and compared to typical and non-communicated classification tasks. Results of the proposed approach are more accurate than the non-communicated classification, but the typical classification task obtained improved results. This result showed the capability of the utilized mutual collaboration approach in improving the results of the participated agents. Moreover, the proposed approach consolidates information diversity and agent specificity, preserves data coverage, and ensures low communication and processing overhead.

Our future work is model updating, where the initiator agent uses the considered conditional probability to update its probabilities to prevent sending various requests for identical data and learn about the infirm attributes. The selected joint conditional probability will be used to update the conditional probability.

Author Contributions: Conceptualization, M.H.Q.; data curation, M.H.Q. and M.A. (Mohammad Aljaidi); formal analysis, M.H.Q., M.A. (Mohammad Aljaidi) and A.A.; funding acquisition, M.H.Q., M.A., G.S. and R.A.; investigation, M.H.Q., M.A., G.S. and R.A.; methodology, M.H.Q. and M.A. (Mohammad Aljaidi); project administration, M.H.Q., M.A. (Mohammad Aljaidi), A.A. and G.S.; resources, A.A.; software, M.H.Q.; supervision, M.A. (Mohammad Aljaidi); validation, M.H.Q., M.A. (Mohammad Aljaidi) and A.A.; writing—original draft, M.H.Q. and M.A. (Mohammad Aljaidi); writing—review and editing, M.H.Q., M.A. (Mohammad Aljaidi), A.A., G.S., R.A. and M.A. (Mohammed Alshammari) All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Zarqa University, Jordan. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA, for funding this research work through project number “NBU-FFR-2023-0076”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be shared for review based on the editorial reviewer’s request.

Acknowledgments: The authors would like to extend their sincere appreciation to Zarqa University, Jordan and Northern Border University, Saudi Arabia for supporting this research.

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

References

- Zhai, Z.; Martínez, J.F.; Beltran, V.; Martínez, N.L. Decision support systems for agriculture 4.0: Survey and challenges. *Comput. Electron. Agric.* **2020**, *170*, 105256. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- Khemakhem, F.; Ellouzi, H.; Ltfi, H.; Ayed, M.B. Agent-based intelligent decision support systems: A systematic review. *IEEE Trans. Cogn. Dev. Syst.* **2020**, *14*, 20–34. [\[CrossRef\]](#)
- Sánchez-Marrè, M. *Intelligent Decision Support Systems*; Springer International Publishing: Cham, Switzerland, 2022; pp. 77–116.
- Manziuk, E.A.; Barmak, A.V.; Krak, Y.V.; Kasianiuk, V.S. Definition of information core for documents classification. *J. Autom. Inf. Sci.* **2018**, *50*, 25–34. [\[CrossRef\]](#)
- Shi, P.; Yan, B. A survey on intelligent control for multiagent systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 161–175. [\[CrossRef\]](#)
- Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [\[CrossRef\]](#)
- Rizk, Y.; Awad, M.; Tunstel, E.W. Decision making in multiagent systems: A survey. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *10*, 514–529. [\[CrossRef\]](#)
- 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. [\[CrossRef\]](#)
- 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.
- Dominguez, R.; Cannella, S. Insights on multi-agent systems applications for supply chain management. *Sustainability* **2020**, *12*, 1935. [\[CrossRef\]](#)
- Koketsu, J.; Ishigaki, A.; Ijuin, H.; Yamada, T. Concept of Sustainable Supply Chain Management Using Multi-agent System: Negotiation by Linear Physical Programming. In *Global Conference on Sustainable Manufacturing*; Springer International Publishing: Cham, Switzerland, 2022; pp. 119–126.
- Binyamin, S.S.; Ben Slama, S. Multi-Agent Systems for Resource Allocation and Scheduling in a Smart Grid. *Sensors* **2022**, *22*, 8099. [\[CrossRef\]](#) [\[PubMed\]](#)
- Dash, D.; Cooper, G.F. Model averaging for prediction with discrete Bayesian networks. *J. Mach. Learn. Res.* **2004**, *5*, 1177–1203.
- Januzaj, E.; Kriegl, H.-P.; Pfeifle, M. DBDC: Density based distributed clustering. *Adv. Database Technol.-EDBT* **2004**, *2004*, 529–530.
- Klusck, M.; Lodi, S.; Moro, G. Distributed clustering based on sampling local density estimates. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, IJCAI, Acapulco, Mexico, 9–15 August 2003*.
- Stolfo, S.J.; Prodromidis, A.L.; Tselepis, S.; Lee, W.; Fan, D.W.; Chan, P.K. JAM: Java Agents for Meta-Learning over Distributed Databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, Newport Beach, CA, USA, 14–17 August 1997.
- Kargupta, H.; Byung-Hoon, D.H.; Johnson, E. Collective data mining: A new perspective toward distributed data analysis. In *Advances in Distributed and Parallel Knowledge Discovery*; Citeseer: Princeton, NJ, USA, 1999.

19. Albashiri, K.A.; Coenen, F.; Leng, P. EMADS: An extendible multi-agent data miner. *Knowl.-Based Syst.* **2009**, *22*, 523–528. [[CrossRef](#)]
20. 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.
21. Cao, L.; Weiss, G.; Philip, S.Y. A brief introduction to agent mining. *Auton. Agents Multi-Agent Syst.* **2012**, *25*, 419–424. [[CrossRef](#)]
22. Rao, V.S. Multi agent-based distributed data mining: An overview. *Int. J. Rev. Comput.* **2009**, *3*, 83–92.
23. Fernandez-Luna, J.M.; Huete, J.F.; Osorio, G. Collaborative Document Classification: Definition, Application and Validation. In Proceedings of the Second International Conference on Advanced Collaborative Networks, Systems and Applications, Venice, Italy, 24–29 June 2012; pp. 42–49.
24. Chaimontree, S.; Atkinson, K.; Coenen, F. A Multi-agent Based Approach to Clustering: Harnessing the Power of Agents. In Proceedings of the ADAMI, Clemson, SC, USA, 14–16 April 2011; Springer: Berlin/Heidelberg, Germany, 2011.
25. 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. [[CrossRef](#)]
26. 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.
27. Samarati, P.; Sweeney, L. *Protecting Privacy When Disclosing Information: K-Anonymity and Its Enforcement through Generalization and Suppression*; Technical report; SRI International: Menlo Park, CA, USA, 1998.
28. Bordini, R.H.; Braubach, L.; Dastani, M.; Seghrouchni, A.E.F.; Gomez-Sanz, J.J.; Leite, J.; O’Hare, G.; Pokahr, A.; Ricci, A. A survey of programming languages and platforms for multi-agent systems. *Informatica* **2006**, *30*, 33–44.
29. FIPA. Fipa Communicative Act Library Specification. Foundation for Intelligent Physical Agents. 2008. Available online: <http://www.fipa.org/specs/fipa00037/SC00037J.html> (accessed on 30 June 2004).
30. Odell, J.J.; Parunak, H.V.D.; Bauer, B. Representing agent interaction protocols in UML. In *Agent-Oriented Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2001.
31. 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. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.