

Research Article

Multiagent System for Mutual Collaboration Classification for Cancer Detection

Mais Haj Qasem , Amjad Hudaib, and Nadim Obeid

King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

Correspondence should be addressed to Mais Haj Qasem; mais_hajqasem@hotmail.com

Received 26 August 2019; Accepted 8 October 2019; Published 6 December 2019

Academic Editor: Alessio Ishizaka

Copyright © 2019 Mais Haj Qasem et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A multiagent system (MAS) is a mechanism for creating goal-oriented autonomous agents in shared environments with communication and coordination facilities. Distributed data mining benefits from this goal-oriented mechanism by implementing various distributed clustering, classification, and prediction techniques. Hence, this study developed a novel multiagent model for distributed classification tasks in cancer detection with the collaboration of several hospitals worldwide using different classifier algorithms. A hospital agent requests help from other agents for instances that are difficult to classify locally. The agents communicate their beliefs (calculated classification), and others decide on the benefit of using such beliefs in classifying instances and adjusting their prior assumptions on each class of data. A MAS model state and behavior and communication are then developed to facilitate information sharing among agents. Regarding accuracy, implementing the proposed approach in comparison with typically different noncommunicated distributed classifications shows that sharable information considerably increases the classification task accuracy by 25.77%.

1. Introduction

Data mining (DM) is the process of analyzing data, discovering data patterns, and predicting future trends based on previously analyzed information. DM techniques require training data to be accessible to the running algorithm to be useful; hence, data should be located physically or virtually in a single unit [1]. As one of the DM techniques, classification task uses the stored training dataset to establish a model used in predicting the class of instances with unknown class labels. A typical classification task stores the training sample and trained model in a single accessible storage [2].

With the increase in distributed data, cloud computing, and multiagent systems (MASs) as well as the wide usability of microprocessor devices, such as mobiles and sensors, data generated or obtained at multiple data acquisition parties are not transferred into centralized data warehouses [3]. Secured data, such as financial records, cannot be transferred because of data transformation due to possibility of exposure [4].

Moreover, other data, such as medical information, cannot be shared or transferred into other locations [5]. Subsequently, a data classification task becomes a distribution task in nature.

In such cases, the available option is to share the learned element in a distributed data mining (DDM) environment. DDM is a field that deals with analyzing distributed data and proposes algorithmic solutions to perform different data analysis and mining operations in a distributed manner by considering resource constraints. Patterns are discovered, and predictions are implemented based on multiple distributed data sources. DDM also solved the DM problem that requires data to be located in a single unit to execute its process. Subsequently, various DDM techniques, such as distributed clustering [6], distributed frequent pattern mining [7], and distributed classification [8], have been developed in the literature. As a distributed system, the use of MAS combined with DDM for data-intensive applications is appealing [9].

The MAS is suitable for distributed problem solving and enables the creation of goal-oriented autonomous agents that operate in shared environments with communication and coordination capabilities. Such system can be defined as a collection of agents with their own problem-solving capabilities that can interact to reach an overall goal [10]. Agents are specialized problem-solving entities that have well-defined boundaries and the ability to communicate with other agents. They are designed to fulfill a specific purpose and exhibit flexible and proactive behaviors [11].

In MAS environments, there are no assumptions regarding global control, data centralization, and synchronization. Thus, agents in MAS are assumed to operate with incomplete information or capabilities in solving problems [12]. Communication is the key for agents to share the information they collect, to coordinate their actions, and to increase interoperation. Interactions between the agents can be requested for information, particular services, or an action to be performed by other agents as well as issues that concern cooperation, coordination, and/or negotiation to arrange interdependent activities [13].

The proposed system in this study is focused on cancer hospitals worldwide. Each hospital has its own samples and prior beliefs on cases and the corresponding diagnosis, which may differ on the basis of region, way of life, and other factors [14]. In many cases, some hospitals diagnose and classify symptoms that appear uncommon from their archived cases. In these cases, classifying symptoms requires wise and intelligent collaboration. Thus, hospitals in various regions worldwide should exhibit diversity in relation to medical diagnosis. Hospitals should also collaborate information with such hospital to diagnose and classify symptoms that appear uncommon to that hospital [15].

Each hospital stores its datasets related to patient cases and builds its model by choosing the most suitable classification technique depending on their characteristic datasets. Moreover, in the proposed system, hospitals in various regions should collaborate their information with a hospital that is about to classify a new uncommon case to one of the hospitals. Thus, a data classification task becomes a distribution task in nature. Consequently, multiple data acquisition parties that are not transferred into centralized data warehouses are present due to patient privacy and possibility of exposure [16].

For this reason, a multiagent system (MAS) for mutual collaboration classification is proposed in this study. The benefits of MAS include allowing agents to build their individual learned model and to control the transfer of their learned information or output results to other agents. This strategy will prove useful in obtaining better output by combining the results of multiple classifiers and maintaining the accuracy of each agent.

For classifying a new case, an agent independently calculates the probability of the case without any collaboration if and only if the probability of the produced classification output is above a certain probability " p ." In such a case, the classification model at the local agent is considered sufficient to classify the input instance. If the probability of the produced classification output is under " p ," an agent

requests help from other agents if the case is uncommon from their stored cases and is difficult to classify locally. In feedback processing, four methods are tested to help the initiator decide whether to accept the received class label from other agents, which will be discussed later.

Autonomous hospital agents can interact with other agents using a specific communication language that enables them to become involved in larger social networks, respond to changes, and/or achieve goals with the help of other agents. Hospital agent interactions and communications in MAS are controlled by communication protocols.

A mutual collaborative classification system is conducted while preserving two issues required in DDM applications. The first issue is to cover private and nonsharable data, and the second is to maintain the distinctiveness or particularity of each agent. This mutual collaborative classification system can be categorized as a special case of distributed classification, where each agent can have its own classification mechanism and can decide whether and when to request information from other agents.

Subsequently, the proposed system for solving the collaborative classification task is established on the basis of MAS, DM, and DDM. The proposed system is compared with the noncommunicated classification. This paper is organized as follows. Section 2 reviews works closely related with the use of MAS and DDM. Section 3 presents the proposed technique, and Section 4 describes the results. Section 5 provides the conclusion.

2. Related Work

DDM, which is implemented by multiple methods with unified goal, establishes the roles of data analysis at multiple distributed data sources. Accordingly, result sharing facilitates the contribution among distributed methods. Overall, DDM can be classified into two groups based on the level of information sharing:

- (i) Low-level DDM: integration of voting results of multiple independent decision-making
- (ii) High-level DDM: combination of learning results using meta-learning

In low-level DDM, each method (e.g., decision-maker) is trained on the basis of its own data. Afterward, all the decision-makers are given the same task for implementation, which can be an instance to be classified, clustered, or analyzed. All the results of these makers are combined to produce an output. DBDS [17] is a distributed clustering approach, which is implemented on dense clustering algorithms; this approach operates locally. The cluster centers, which are produced locally at each distributed source, are transformed with small number of data elements to a decision-making center, which recalculates the cluster centers based on the received centers and elements.

Similar other approaches for distributed clustering without element transformation were proposed in [18]. In classification, a clear example of such approach is bagging and boosting approaches, which allow multiple classifiers to operate on different sets of data [19]. Bayesian classifiers are

operated in a distributed environment by averaging the local model of the distributed sources to obtain a global one [20, 21].

In high-level DDM, each source shares its learned model with the global model to produce a single learning model for mining the input data. This DDM type is called meta-learning or meta-DM. In classification, various tools, such as JAM [22] and BODHI [23], were proposed for this purpose.

Relevant researchers of DDM realized the benefits of using MAS in implementing, organizing, and controlling distributed sources. MAS is a process for creating goal-oriented autonomous agents with coordination and communication capabilities in shared environments. DDM benefits from this goal-oriented mechanism by executing different distributed clustering, prediction, and classification techniques.

MASs frequently handle complex applications that require distributed problem solving. Meanwhile, DDM is a complex system that is focused on data mining processes and resource distribution over networks. Scalability lies at the core of DDM systems. Given the occasional changes in system configurations, the DDM system design considers many details regarding software engineering, such as extensibility, reusability, and robustness. Therefore, the characteristics of the agents are favorable for DDM systems [24].

The collective and individual behaviors of agents in many applications rely on the data observed from distributed sources. Distributed data analysis is a nontrivial problem in typical distributed environments because of various constraints, including distributed computing nodes, limited bandwidth (e.g., wireless networks), and privacy-sensitive data. In analyzing distributed data, the DDM domain addresses these challenges and provides numerous algorithmic solutions to conduct diverse mining operations and data analysis in a fundamentally distributed manner that is focused on resource constraints [25]. The combination of MAS and DDM is preferred for data-intensive applications because of the distributed nature of MASs.

Various MAS-based DDM approaches have been proposed. EMADS [26] was proposed as MAS based on the ensemble classification, which provides weights for each distributed classifier or select one to perform the classification task based on knowledge about the learning model at each classifier. Similarly, an abstract architecture for MAS-based distributed classification with various methods of result integration was proposed [27, 28]. A brief overview of these approaches is described in [29].

MAS-based DDM allows agents to establish an individual learned model and control the transformation of their learned information or results into global and central agents, which produce the final output. This approach is beneficial in obtaining enhanced results by combining results of multiple classifiers. Another benefit of this approach is the maintenance of the particularity of each agent so that each agent can be used individually when data particularity is needed, such as in medical and document classification examples. However, two limitations of these approaches

include the inability to share information among local agents to enhance the capabilities of autonomous agents and the uncovering of private and nonsharable data.

Mutual collaborative DM approaches were proposed recently. These approaches can improve the initial learned model at each agent using sharing information among agents. Semiautomatic distributed document classification was also proposed to enhance the classification results and allow mutual collaboration among indexers [30]. This framework implements mutual collaboration, and it requires human interference to determine the suitability of the collaborated information. MAS-based clustering framework that can improve the initial cluster centers at each agent was also proposed recently [31].

Results of the proposed collaborative clustering showed an improvement over noncollaborative agent-based clustering. This framework also maintains the particularity of each agent and allows information sharing among local agents to enhance the capabilities of autonomous agents and cover private and nonsharable data. In this research, as similar to the mutual collaborative clustering proposed in [31], an approach of mutual collaborative classification is proposed.

3. Proposed Work

The proposed work aims to develop a collaborative classification model using an appropriate MAS. The system proposed in this study focuses on cancer hospitals worldwide. Each hospital has its own samples and prior beliefs on cases and the corresponding diagnoses, which may differ on the basis of region, way of life, and other factors. Different cancer datasets for the same disease are collected from various sources. However, this approach cannot be sustained in a long research stage because of the scarcity of datasets in the open online repository. Therefore, the same cancer dataset is distributed through a clustering algorithm used in the proposed system to verify the most effective cluster number and select the most suitable number of agents that can communicate and work with one other.

After clustering the dataset and choosing the most suitable agent number, each cluster is given to a different agent. Following, each agent carries out feature selection on the given cluster. Selecting features after the agent is given a cluster is important. Considering that each hospital worldwide may require features that vary from those required by other hospitals, datasets are distributed vertically and not horizontally. In addition, the clusters obtained via the clustering algorithm exhibit related features. As irrelevant or less-important features can negatively affect model performance, feature selection and data cleaning should be the first and most important steps in model design.

Each agent builds its model by choosing the most suitable classification technique depending on the characteristics of the given cluster. Typically, machine learning algorithm varies for each agent, and auto classification is applied so that each agent can use the algorithm of the proposed system to select the most suitable model for its dataset. Autoclassification means that each agent applies a

different machine learning algorithm that is available for use in the proposed system.

Given an instance to be classified, a local agent calculates independently the probability of the instance without any collaboration if and only if the probability of the produced classification output is above a certain probability “ p .” In such a case, the classification model used by the local agent is considered sufficient to classify the input instance.

However, if the classification output does not reach the specific probability p , then the agent requests information from other agents. Other agents help in calculating the probability of the case, and they communicate their findings or beliefs (calculated classification). They decide on the benefit of using such beliefs in classifying instances and adjusting their prior assumptions on each class of data in an exchange only if the probability of the classification output is above “ p .”

Agents communicate with one another using the query interaction protocol (IP) of the Foundation for Intelligent Physical Agents (FIPA) that allows one agent to request another agent to perform a certain action. Through this IP, agent communication language (ACL) messages that hold the request or response to specific kinds of information are delivered. The representation of this IP will be discussed later.

MAS for classification task is developed using SPADE that evolved full-featured FIPA platform. SPADE agents have behaviors that is a task that an agent can execute using repeating patterns; these behaviors types help to implement the different tasks that an agent can perform. Behaviors have registered states and transitions between states. This kind of behavior allows SPADE agents to build behaviors in the agent model.

Each SPADE agent has an internal message dispatcher component. This message dispatcher acts as a mailman: when a message for the agent arrives, it places it in the correct “mailbox,” and when the agent needs to send a message, the message dispatcher does the job, putting it in the communication stream. The message dispatching is done automatically by the SPADE agent library whenever a new message arrives or is to be sent.

In feedback processing, four methods are tested to help the initiator decide whether to accept the received class label from other agents, and these methods will also be discussed later. The math model of the proposed system is summarized in the following pseudocode and illustrated in Figure 1. Table 1 annotates the variables used in the pseudocode presented in Algorithm 1.

The main objective of the proposed system is to preserve privacy. Hospitals in various regions should collaborate their information with one another to classify new and uncommon cases that are local to any of them. Therefore, the data classification task becomes a distribution task in nature. Consequently, the multiple data acquisition parties that are not transferred into the centralized data warehouses are present because of the security and privacy issues of the patients in each hospital, as well as the possibility of exposure. The proposed technique aims to promote

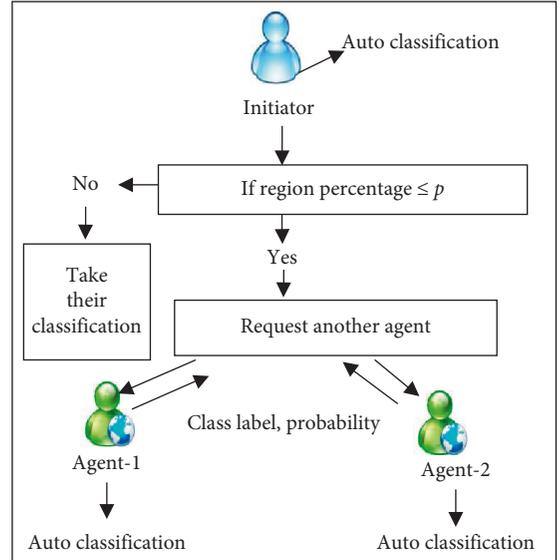


FIGURE 1: Proposed collaborative classification framework.

TABLE 1: Explanation of valuables in the pseudocode of classification algorithm.

The explanation of valuables in multiagent systems for mutual collaboration classification	
Valuable	Explanation
D	Dataset for building model
M	The most appropriate algorithm for their respective problems
NI	New instance to be classified
P	Probability
C	Class label
R	Result

information privacy among agents by sharing limited data on the constructed model of each agent.

In achieving the intended collaboration task, the following characteristics require consideration when building the proposed system. First, each agent should be able to incorporate knowledge acquired from other agents and the utilized classifier should be able to integrate other sources of knowledge aside from its training data in mutual collaboration tasks. Second, each agent should be able to deal with missing values, which are added incrementally to the model, and the utilized classifier should be able to deal with missing data by averaging over the possible values that the attribute might have taken. Third, each agent should be able to share part of the trained model without sharing whole training samples and the utilized classifier should be able to share and hide some components of the models. Finally, each agent will apply the different classification algorithm where each one of the agents chooses the suitable classification algorithm based on their dataset.

The objectives of this research are as follows:

- (i) To critically analyze the capabilities of existing classification algorithms in preserving privacy and particularity features in mutual collaboration

```

Agent (NI, D)
{
    M = Auto Classification (D) //select the most suitable model for their dataset
    Classify(NI) //Classify the new instance
    {
        C, P = M(NI)
        if P ≤ T
            RFinal = C
        Else
            Cn, Pn = Agentn.Classify(NI)
            //here the communication message between the agent will start
            RFinal = Feedback_Processing (Cn, Pn)
            CFinal = RFinal
    }
}
Auto_Classification (D)
{
    Accuracy = Classification_algorithm (Di)
    Mi = Max (Accuracy)
    Return Mi
}
Feedback_Processing (Ci, Pi)
{
    //four methods are tested to help the initiator decide whether to accept the received class label from other agents
    Return CFinal
}

```

ALGORITHM 1: Pseudocode of the proposed system.

classification. This objective covers other sub-objectives, which include the following: (1) review the existing algorithms for data classification; (2) review the existing distributed classification approach and highlight the advantages, disadvantages, and its performance; and (3) analyze the privacy and particularity requirements.

- (ii) To propose and adapt mutual collaborative classification approaches that preserve private and nonsharable data, maintain particularity of each agent and produce accurate classification results in distributed environment.
- (iii) To propose a system with a different learning model, where each agent can use the most suitable classification algorithm based on their dataset. In that case, efficiency of different classification algorithms will be seen in improving the classification results.
- (iv) To analyze critically the capabilities of the existing multiagent protocols in terms of load and fault tolerance. This objective covers other sub-objectives, including (1) review the existing communication protocols; (2) compare the usability of these protocols based on the MAS task; and (3) analyze the load and fault tolerance requirements.
- (v) To propose a MAS protocol to control the system structure, standardize the communications among

agents and satisfy the requirements of the mutual collaborative classification task.

- (vi) To implement, validate, evaluate, and compare the results of the proposed system with the typical distributed classification.

Overall, the proposed technique involves agents that work independently and collaborate with other agents in the field. The aims of the developed technique are as follows:

- (i) Share information on data among agents to facilitate agent collaboration in the classification task
- (ii) Promote information diversity among agents and agent particularity based on its own data
- (iii) Promote information privacy among agents by sharing limited data on the constructed model of each agent
- (iv) Enhance agent results by incorporating inputs from other agents
- (v) Ensure limited communication overhead among the distributed agents and prevent data reallocation and centralization
- (vi) Ensure low processing cost by processing data blocks at each agent independently

The fifth phase presents the implementation of the proposed work in phases. The first phase demonstrates the techniques used to distribute the dataset among the agents. The second phase describes the feature selection

method used to select the most important feature for each agent dataset. The classification algorithm that each agent applies to build their model is discussed in the third phase, whereas the MAS protocols that agents use in their interactions and communications are described in the fourth phase. Finally, the fifth phase explains how the proposed system works. The proposed methodology phases are illustrated in Figure 2 and discussed in the following section.

3.1. System Constructed Phases. In this research, a set of phases are followed to achieve the desired goal.

3.1.1. First Phase. Different cancer datasets for the same disease from different sources are collected, In fact, this was the only data set we could have access. Therefore, the first phase is the distribution of the same cancer dataset used in the proposed system and the selection of the most suitable agent number that would communicate and work with each other for each dataset. In this phase, the dataset is clustered by using the K -mean algorithm to verify the most effective cluster number. K -mean is one of the most popular clustering algorithms that allow the specification of the required cluster number [32].

3.1.2. Second Phase. The second phase involves feature selection. The data features used to train machine learning models greatly influence the models' performance. Therefore, the clusters obtained from the K -means algorithm will exhibit related features and irrelevant or less-important features that can negatively impact model performance [33]. Feature selection and data cleaning should ultimately be the first and most important step in model design.

Feature selection after clustering is important because each hospital may require features that are different from those required by other hospitals. This phenomenon is the reason why datasets are distributed vertically and not horizontally.

The feature selection mechanism consists of three steps: (1) screening, wherein unimportant and problematic inputs, records, or cases (e.g., input fields with too many missing values or with too much or too little variation) are removed; (2) ranking, in which the remaining inputs are sorted to assign ranks according to importance; and (3) selecting, in which the subset of features to be used in subsequent models is identified—for example, by preserving only the most important inputs and filtering or excluding the others.

At the end of this phase, the second objective will be achieved, that is, the most important feature for each agent is selected, and the vertically distributed data are applied to hospital systems.

3.1.3. Third Phase. The third phase is classification distribution, in which the existing data classification algorithms—along with their extended distributed classification approach, advantages and disadvantages, and performance—are analyzed.

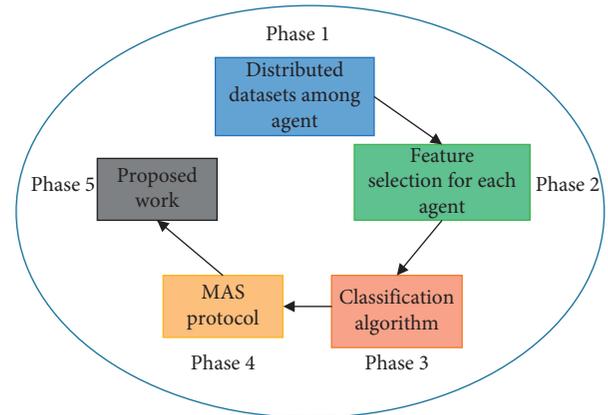


FIGURE 2: Proposed system phases.

In addition, a critical analysis of the algorithms' capabilities of preserving privacy and particularity in a distributed environment is conducted to attain the first objective, which includes highlighting algorithm performance.

After the setting of their datasets and selection of features, each hospital agent will have either distinct or similar features, and then they will choose the most appropriate algorithm for their respective problems. Typically, the machine learning algorithm will be different for each agent, and auto classification will be applied so that each agent can use the algorithm of the proposed system to select the most suitable model for their dataset.

Auto classification means that everyone of the involved agents could choose, from among a number of classification algorithms, the algorithm that most suits its data set.

- (1) Logistic regression is a statistical technique to classify records based on the values of input fields. It targets a categorical rather than a numeric field, despite its similarity to linear regression [34].
- (2) Support vector machine is a robust classification technique that maximizes the predictive accuracy of a model without overfitting the training data. Support vector machine (SVM) is particularly suited for data analysis with extremely large numbers [35].
- (3) Bayesian network is a traditional probabilistic technique successfully used by various machine learning methods to help solve different problems in various domains [36].
- (4) Neural network is a simplified model of the human brain's mechanism to process information. It simulates a large number of interconnected processing units resembling abstract versions of neurons [37].

Logistic regression, SVM, Bayesian network, and neural network are selected for application on the agent and enable them to choose the most suitable model for their dataset. The reason for this selection is that after analysis and search among several machine learning algorithms, these classification algorithms can automate the detection of cancer and disease prediction. In addition, these algorithms can be used for classification tasks that require more accuracy and efficiency of data.

Completion of this phase achieves the third objective embodied in the critical analysis of the algorithm capabilities to preserve privacy and particularity together with high-lighting performance.

3.1.4. Fourth Phase. The fourth phase is concerned with existing MAS protocols, which are identified accordingly. The advantages, disadvantages, and the load and robustness capabilities of these protocols are analyzed.

The MAS is suitable for distributed problem solving and enables the creation of goal-oriented autonomous agents that operate in shared environments with communication and coordination capabilities. Such system can be defined as a collection of agents with their own problem-solving capabilities that can interact to reach an overall goal [38]. Agents are specialized problem-solving entities that have well-defined boundaries and the ability to communicate with other agents. They are designed to fulfill a specific purpose and exhibit flexible and proactive behaviors.

Agents are autonomous in the sense that they can operate on their own without need for guidance and can use the most suitable classification algorithm for their datasets without limitation like DDM. They have control over their internal state and actions. An agent can decide on its own whether to perform a requested action [39]. They are capable of exhibiting flexible problem-solving behavior in pursuit of the objectives they are designed to fulfill. Autonomous agents can interact with other agents using a specific communication language that enables them to become involved in larger social networks, respond to changes, and/or achieve goals with the help of other agents [40].

Agents usually operate in a dynamic, nondeterministic complex environment. In MAS environments, there are no assumptions regarding global control, data centralization, and synchronization. Thus, agents in MAS are assumed to operate with incomplete information or capabilities in solving problems [41]. Communication is the key for agents to share the information they collect, to coordinate their actions, and to increase interoperation. Interactions between the agents can be requested for information, particular services, or an action to be performed by other agents as well as issues that concern cooperation, coordination, and/or negotiation to arrange interdependent activities.

Agents need to interact with one another, either to achieve their individual objectives or to help others. Such interaction is conceptualized as taking place at the knowledge level [42]. That is, interaction can be conceived in terms of which goals should be followed, at what time, and by which agent. Furthermore, because agents are flexible problem solvers that only have partial control and partial knowledge of the environment in which they operate, interaction must be handled in a flexible manner and agents need to make run-time decisions on whether to initiate interactions based on their nature and scope [43].

Agent interactions and communications in MAS are controlled by communication protocols. MAS protocols determine external agent behavior side by side with the internal agent structure without focusing on the internal

agent behavior [44]. Together with the task implemented by the underlying MAS, these specifications can determine the requirements of internal agent behavior. Various MAS-based DDM approaches have been proposed. Their benefits include allowing agents to build their individual learned model and to control the transfer of their learned information or output results to other agents. This may prove useful in obtaining better output by combining the results of multiple classifiers and maintaining the accuracy of each agent.

DDM can benefit from such a mechanism in implementing various distributed clustering, classification, and prediction techniques. For instance, an agent may request help from other agents in instances that are difficult to classify locally. The agents communicate their findings or beliefs (calculated classification) and others decide on the benefit of using such beliefs in classifying instances and adjusting their prior assumptions on each class of data in an exchange that can be described as a mutual collaborative classification task.

Existing protocols for agent communication can be classified broadly into two categories, centralized and decentralized:

- (i) In centralized protocols, the agent that requests to interact with others is called the initiator. The request is sent to a center agent called the broker, which in turn selects the intended agent or participant based on the specifications given by the initiator and forwards the request accordingly. This protocol category has the advantage of allowing the broker to select the most suitable participant. However, the disadvantages are the necessity to record, manage, and exchange the history of the participant to avoid incorrect selection.
- (ii) In the decentralized protocols, the requesting agent interacts with the participant directly, having the address and status information from the registry agent that serves as the yellow pages with no information on agent activities. The advantage of this category is the robustness and adaptability in dynamic situations. Similarly, the disadvantages of needing to record, manage, and exchange the history of the participant to avoid incorrect selection remain, although the disadvantage is the load created in a large system. The proposed system uses the decentralized protocols.

Each agent in the proposed technique possesses its own data and independently processes tasks when sharing with and acquiring information from other agents [45]. Agents communicate with one another using the Foundation for Intelligent Physical Agents (FIPA) query interaction protocol (IP) that allows one agent to request to perform some kind of action on another agent and Agent Communication Language (ACL) messages that hold the request or response to specific kinds of information. Query IP is one of the FIPA [46] protocols implemented by various multiagent developing environments [47]. The representation of this IP is given in Figure 3.

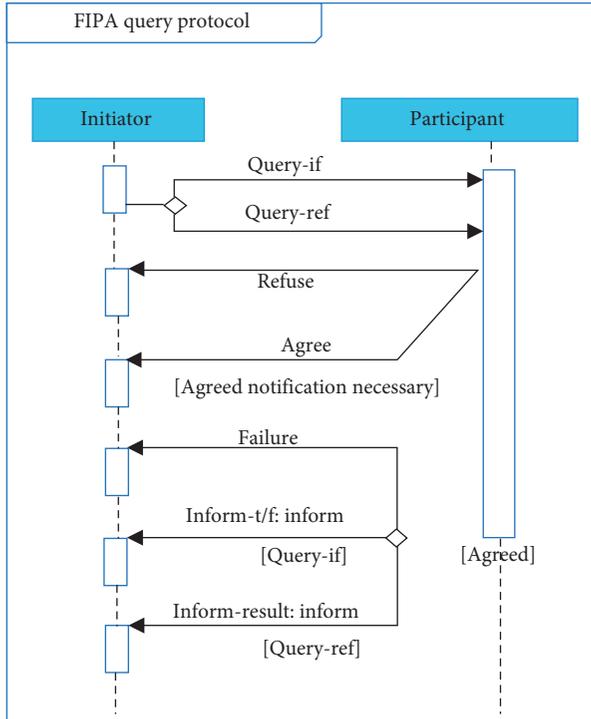


FIGURE 3: FIPA query interaction protocol.

The IP is a set of specifications that allows one agent, referred to as the initiator, to request to perform tasks with other agents [48]. The specification covers the message type, structure, and exchanging states and conditions. The messages used can be a type request or call as defined by ACL [49], which also characterizes the message structure itself. The message-exchanging conditions are defined by the protocol.

A FIPA ACL message contains a set of one or more message parameters. The precise parameters required for an effective agent communication will vary according to the situation; the only parameter that is mandatory in all ACL messages is the performative one although most ACL messages are expected to contain sender, receiver, and content parameters.

Completion of this phase achieves the fourth objective embodied in the critical analysis of the MAS protocols by highlighting the load and robustness capabilities.

3.1.5. Fifth Phase. The fifth phase is concerned with feedback processing. The following four methods tested to help the initiator decide whether to accept the received class label from the other agents.

(1) First Method (Voting). The requesting agent receives a class label from all other agents that produced a classification output reaching a specific probability " p " and then takes the class label with the most votes. Table 2 shows the example summarizing the voting method.

The class label output in Table 2 from all contacted agents shows 4 yes vs 1 no. Therefore, the initiator agent will rely on the yes answer that garnered the most votes.

TABLE 2

Agents	Class label
Initiator agent	Yes
Agent 1	No
Agent 2	Yes
Agent 3	Yes
Agent 4	Yes

(2) Second Method (Small K -Nearest Neighbors). All the agents will apply the K -nearest neighbors (KNN) algorithm on their datasets. The KNN algorithm is one of the most flexible supervised learning tools used for classification and prediction, such as that in DM and machine learning. KNN is a method for classifying cases based on their similarity with other cases.

The number of the nearest neighbors, represented as k , can be specified for investigation. After all the agents applied KNN algorithm including the requested agent, the requested agent will receive class label from all other agents and the sum of distance for k neighbors for the new instance that wants to classify. The requested agent will rely the class label with the smallest KNN distance [50]. Table 3 shows the example summarizing the small KNN method.

The class label output in Table 3 from all the contacted agents shows that small KNN is Agent 1. Therefore, the initiator agent will rely on the yes answer.

(3) Third Method (High Probability). The requesting agent receives a class label and its probability from all other agents. Probability means the certainty of other agents on their class label classification depending on their model. Therefore, the initiator agent will take the class label with the highest probability. Table 4 shows the example summarizing the high probability method.

The class label output in Table 4 from all contacted agents shows that the highest probability is from Agent 2. Therefore, the initiator agent will rely on the yes answer.

(4) Fourth Method (Mix of the Three Methods Above). This method has two cases. In the first case, the voting method produces different but equal class labels. In this case, no class label will win the voting, as in the example given in Table 5.

The class label output in Table 5 from all the contacted agents shows 2 yes vs 2 no answers, and thus, no class label has the most votes. In this case, a mix between two methods is necessary to determine which class label to take. As such, all the contacted agents likewise send their class label probability to the initiator agent, which then compares the responses and takes the higher probability. Table 6 shows the example summarizing the mixed method.

The class label output in Table 6 from all the contacted agents shows that the two different class labels (yes and no) are equal and the voting method fails. Therefore, the initiator agent relies on the highest probability, which is the no answer.

In the second case, at times, the KNN method shows that two agents have equal distance but different class labels, and thus, no class label wins, as in the example given in Table 7.

TABLE 3

Agents	Class label	KNN sum distance
Initiator agent	<i>Yes</i>	4.243
Agent 1	<i>No</i>	1
Agent 2	<i>Yes</i>	0
Agent 3	<i>Yes</i>	4.878
Agent 4	<i>Yes</i>	4.512

TABLE 4

Agents	Class label	Probability
Initiator agent	<i>Yes</i>	0.696
Agent 1	<i>No</i>	0.604
Agent 2	<i>Yes</i>	0.783
Agent 3	<i>Yes</i>	0.756
Agent 4	<i>Yes</i>	0.679

TABLE 5

Agents	Class label
Initiator agent	<i>Yes</i>
Agent 1	<i>No</i>
Agent 2	<i>No</i>
Agent 3	<i>Yes</i>

TABLE 6

Agents	Class label	Probability
Initiator agent	<i>Yes</i>	0.696
Agent 1	<i>No</i>	0.604
Agent 2	<i>No</i>	0.783
Agent 3	<i>Yes</i>	0.756

TABLE 7

Agents	Class label	KNN sum distance
Initiator agent	<i>Yes</i>	4.243
Agent 1	<i>No</i>	0
Agent 2	<i>Yes</i>	0
Agent 3	<i>Yes</i>	4.878
Agent 4	<i>Yes</i>	4.512

The class label output in Table 7 from all contacted agents shows two different class labels (yes and no) from agents 1 and 2, which have the smallest and equal KNN, and thus, the small KNN method fails. In this case, a mix between two methods is necessary to determine which class label to take. As such, all contacted agents also send their class label probability to the initiator agent, which then compares the responses and takes the higher probability. Table 8 shows the example summarizing the mixed method.

The class label output in Table 8 from all contacted agents with two different class labels (yes and no) from agents 1 and 2 has the smallest and equal KNN. Therefore, the requesting agent relies on the highest probability from agent 2, which is the *yes* answer.

TABLE 8

Agents	Class label	KNN sum distance	Probability
Initiator agent	<i>Yes</i>	4.243	0.696
Agent 1	<i>No</i>	0	0.604
Agent 2	<i>Yes</i>	0	0.783
Agent 3	<i>Yes</i>	4.878	0.756
Agent 4	<i>Yes</i>	4.512	0.679

Therefore, all other agents send the following information to the requesting agent:

- (i) Class labels *C*
- (ii) Probability for the class
- (iii) *K*-nearest neighbors

The initiator agent must then choose one of the voting methods or a mix of methods in case of one of the two instances mentioned above. In the proposed technique with distributed data among multiple agents, each agent shares information that is highly considered correct according to its knowledge of the specific attributes, class, and other agents. This information is highly considered missing by the requesting agent according to its knowledge of specific attributes and class.

Accordingly, agents involved in collaboration classification tasks pursue information diversity and particularity based on their own data while sharing information among each other to enhance the output results.

4. Results

To validate the applicability of the proposed technique and estimate the performance of the involved collaborative classification, a MAS for classification task is developed using open-source tools IBM® SPSS® Modeler software by integrating it with Python to use Smart Python multiagent development environment (SPADE) that is a framework for the development, execution, and management of MAS in distributed computer environments.

The proposed system is applied on two datasets for experimentation and benchmarking. The first dataset is obtained from the openML repository; these datasets of breast cancer were obtained from the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia [51], representing instances for multiclass breast cancer detection classification [52]. This dataset consists of 1 million instances, which are described by 13 linear or nominal attributes.

The second dataset is obtained from the IEEE data port. These datasets of breast cancer patients were obtained from the 2017 November update of the SEER Program of the NCI, which provides information on population-based cancer statistics [53]. This dataset consists of 4024 instances described by nine attributes.

The two datasets were distributed by using the *K*-mean algorithm clustering. After testing for the most suitable agent number that will communicate with each other, the most effective cluster number is 5 for the first dataset and 3 for the second dataset. Figure 4 illustrates the percentage for each cluster for the two datasets.

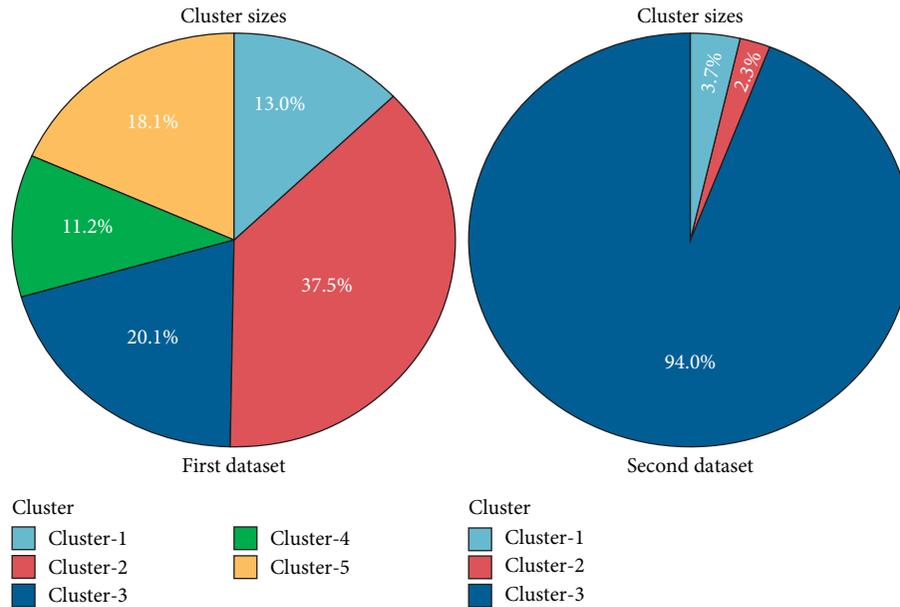


FIGURE 4: Cluster size for two datasets.

After clustering, the feature selection is applied in each cluster and the most important feature is selected for each agent. Feature selection displays the importance of each input relative to a selected target. Figure 5 illustrates the results for several agents.

Accordingly, the resulting data are distributed among the agents, and each agent is given training and testing sets. Each agent applies auto classification and chooses the most suitable model for their dataset from the classification algorithms described in phase three from the previous section. Tables 9 and 10 summarize the overall accuracy for each classification algorithm in each agent for first and second datasets, respectively.

Tables 10 and 11 show the result of contacted agents and that the reason for choosing one of these algorithms is very efficient, where these classification algorithms can automate the detection of cancer and disease prediction. In addition, these algorithms can be used for classification tasks that require more accuracy and efficiency of data.

MAS for classification task is developed using SPADE that evolved full-featured FIPA platform. SPADE agents have behaviors that is a task that an agent can execute using repeating patterns; those behavior types help to implement the different tasks that an agent can perform. Behavior has registered states and transitions between states. This kind of behavior allows SPADE agents to build behaviors in agent model.

After each agent builds their model, the proposed system is executed to validate the process sequence and communication. The states and behavior of each agent is shown in Figure 6 and summarized in Table 11. Figure 7 illustrates the interaction between agents.

The proposed mutual collaboration classification is implemented by using the four methods mentioned in phase four in the previous section and then compared them with the normal DDM algorithm for the classification

algorithm that each agent used without any communication between agents.

4.1. Dataset 1 Result. After several experiments and analyses for each contacted agent in the first dataset, most of the agents can calculate the instance probability independently without any collaboration if and only if the probability of the produced classification output is above 0.6. If the produced classification output does not reach this probability, then the agent requests information from other agents.

To ensure effectiveness of the proposed system, each agent in the first dataset is verified by using five tests that contain 20% of instances without class label different from the original dataset with all the features and require classification. The results for the first dataset are illustrated in Tables 12–16 and shown in Figures 8–12 for each agent.

The results from the first contacted agent show the proposed system is overpowered and more accurate than the distributed system, where the voting method is the best one to enable the initiator to decide whether to accept the received class labels from the other agents.

From the results of the second contacted agent, the proposed system is overpowered and more accurate than the distributed system. The second method with high probability is the best among the first three testing sets and the voting method is the best among the fourth testing set. Meanwhile, the three methods provided the same improvement among the other methods for the fifth testing to allow the initiator to decide whether to accept the received class label from the other agents.

The results of the third contacted agent show the second method with high probability is the best among the first two of the testing set, and voting method is the best for the rest of

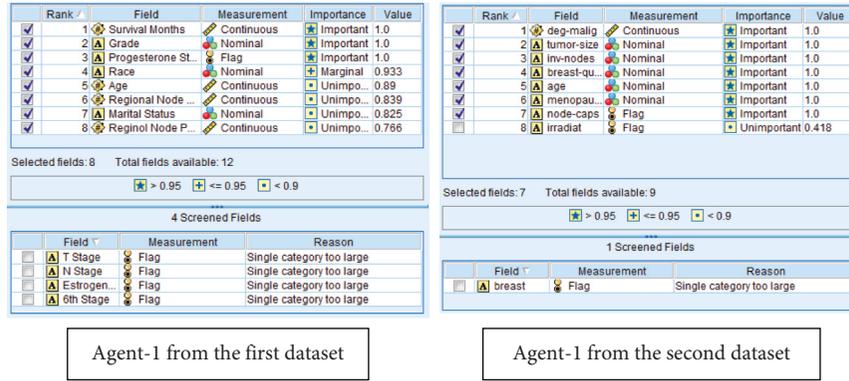


FIGURE 5: Feature selection result.

TABLE 9: Overall accuracy for each classification algorithm (dataset 1).

Model	Max profit	Overall accuracy (%)	Area under curve
<i>Agent 1</i>			
SVM	42,100.0	76.50	0.835
Naive Bayesian	40,367.857	75.1	0.825
Neural network	38842.667	73.93	0.814
<i>Agent 2</i>			
Naive Bayesian	2667.889	78.06	0.737
Neural network	2,127.183	77.74	0.742
Logistic regression	2126.429	77.68	0.723
<i>Agent 3</i>			
SVM	3,635.0	79.99	0.722
Naive Bayesian	2,965.0	79.39	0.722
Logistic regression	2,145.329	78.51	0.758
<i>Agent 4</i>			
SVM	22,040.0	78.45	0.864
Naive Bayesian	19,343.333	75.17	0.831
Neural network	17460.00	72.67	0.802
<i>Agent 5</i>			
Naive Bayesian	1778.889	81.89	0.763
Logistic regression	1,170.0	81.46	0.751
Neural network	820.0	81.14	0.749

the testing set to enable the initiator decide to accept the received class label from the other agents. In addition, the proposed system is overpowered and more accurate than the distributed system.

The results of the fourth contacted agent also show that the system is overpowered and more accurate than the distributed system, where the voting method is the best among the other methods to enable the initiator to decide whether to accept the received class label from the other agents or not.

The results of the fifth contacted agent show that the proposed system is overpowered and more accurate than the distributed. The first voting method is the best among the first testing set, and the second method with high probability is the best among the rest of the testing set to enable the initiator to decide whether to accept the received class label from the other agents.

TABLE 10: Overall accuracy for each classification algorithm (dataset 2).

Model	Max profit	Overall accuracy (%)	Area under curve
<i>Agent 1</i>			
Naive Bayesian	135.0	90.27	0.816
Neural network	105.0	89.38	0.812
Logistic regression	120.0	88.58	0.817
<i>Agent 2</i>			
SVM	1,015.0	88.17	0.892
Naive Bayesian	880.0	86.91	0.875
Logistic regression	870.0	86.66	0.866
<i>Agent 3</i>			
Naive Bayesian	150.0	94.79	0.902
Neural network	130.0	94.56	0.880
SVM	115.0	94.25	0.890

At the end of the experimental results for the first dataset, it is seen that all the methods are excellent in enabling the decision whether to accept the received class label from other agents, while the voting method is shown as the best in most cases. This finding proves the ability of the utilized mutual collaboration approach in enhancing the results of the contacted agents, with the rate of 2.93% to 5.98%. The results for the first dataset are illustrated in Table 17 and shown in Figure 13.

4.2. Dataset 2 Result. Several experiments and analysis were also conducted for each contacted agent in the second datasets. Most agents can calculate the probability of the instance independently without any collaboration if and only if the probability of the produced classification output is above 0.5. If the produced classification output does not reach this probability, then the agent requests information from the other agents.

In addition, to ensure effectiveness of the proposed system, each agent in the second dataset is tested by using three tests that contain 20% of instances different from the original dataset and require classification. The result for the second dataset is illustrated in Tables 16–20 and shown in Figures 14–16 for each agent.

TABLE 11: Main agent state list.

State ID	From behavior	To behavior	Condition
S1	Start-up	Model building	The training data is giving to the agent
S2	Model building	Waiting for input	The model is built using the training data
S3	Waiting for input	Processing	An instance to be classified is giving to the agent, by local input mechanism or as query
S4	Processing	Query response	The instance received by query is processed
S5	Query response	Inform results	The instance received by query is accepted
S6	Inform results	Waiting for input	The results have been informed to the initiator
S7	Query response	Waiting for input	The instance received by query is rejected
S8	Processing	Waiting for input	The local instance is processed
S9	Processing	Query	Agent need collaboration
S10	Query	Waiting for response	Agent sent query for collaboration
S11	Waiting for response	Waiting for results	Agent received acceptance from other(s)
S12	Waiting for results	Processing	Agent received results from other(s)

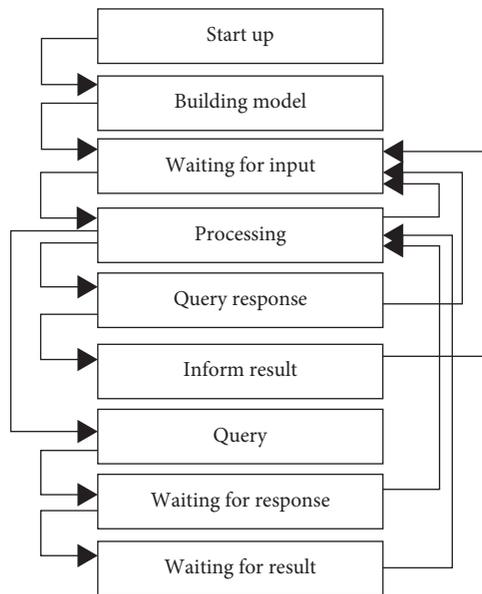


FIGURE 6: Agent states and behaviors.

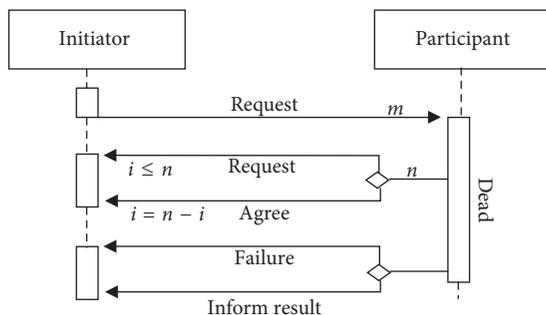


FIGURE 7: Interaction protocol.

The results of the first contacted agent show that the voting method is the best for all the testing set, while method two with high probability also provided the same enhancement for the first method in the third testing set. The proposed system achieved a superior and more accurate result than that of the distributed system.

TABLE 12: Agent 1 results (dataset 1).

Test #	Agent 1			
	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	77.44	76.56	76.67	70.56
2	76.33	74.67	74.67	74.33
3	77.67	76.17	76.17	68.50
4	78.88	77.75	77.88	71.38
5	75.70	74.80	75.00	69.50

The results of the proposed system of the second contacted agent achieved more accurate results than the distributed system. The results of the first method was the best for all the testing sets like the results of the first agent, while method two with high probability also provided the same enhancement for the first method in the third testing set.

The results of the third contacted agent showed that the third method, small KNN, is the best among all the

TABLE 13: Agent 2 results (dataset 1).

Test #	Agent 2			
	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	78.00	79.00	76.00	72.00
2	76.00	78.50	76.00	74.00
3	74.67	76.00	74.67	73.00
4	78.14	77.57	77.14	75.67
5	93.00	93.00	93.00	71.30

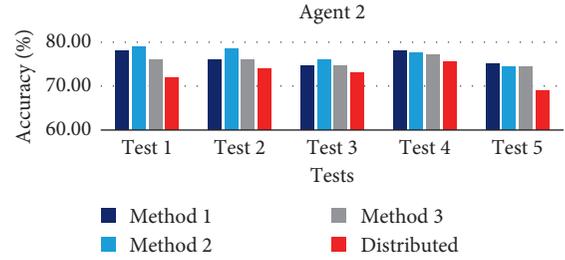


FIGURE 9: Agent 2 results (dataset 1).

TABLE 14: Agent 3 results (dataset 1).

Test #	Agent 3			
	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	75.50	76.50	74.50	73.50
2	75.00	76.00	74.33	73.67
3	77.86	76.86	76.00	74.82
4	77.75	76.88	76.13	75.91
5	74.30	73.40	73.20	68.70

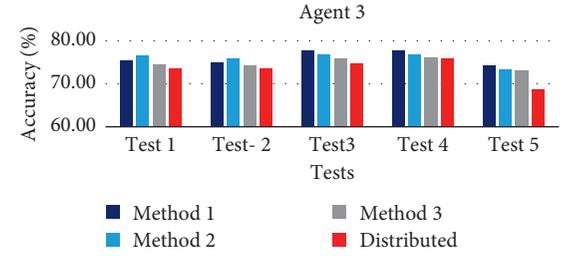


FIGURE 10: Agent 3 results (dataset 1).

TABLE 15: Agent 4 results (dataset 1).

Test #	Agent 4			
	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	77.00	73.00	73.50	71.00
2	75.67	70.00	70.33	69.33
3	77.50	75.17	75.50	71.65
4	78.86	76.86	77.14	73.91
5	75.00	73.60	73.90	67.80

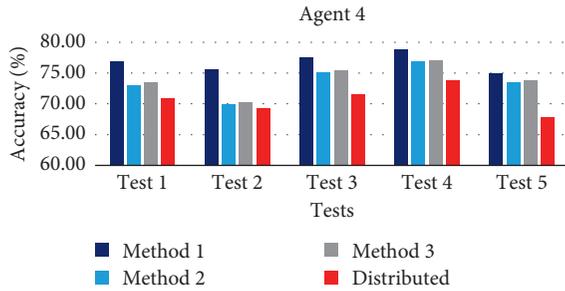


FIGURE 11: Agent 4 results (dataset 1).

TABLE 16: Agent 5 results (dataset 1).

Test #	Agent 5			
	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	73.00	73.00	71.00	64.00
2	74.00	76.00	75.00	71.00
3	76.17	76.83	76.33	74.29
4	77.71	78.29	77.86	76.22
5	74.50	74.80	74.90	74.40

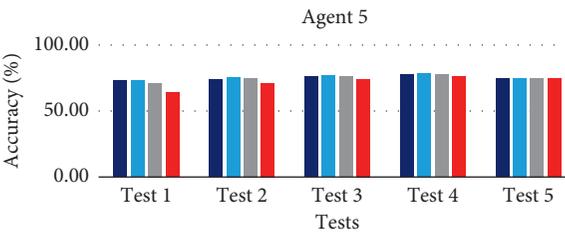


FIGURE 12: Agent 5 results (dataset 1).

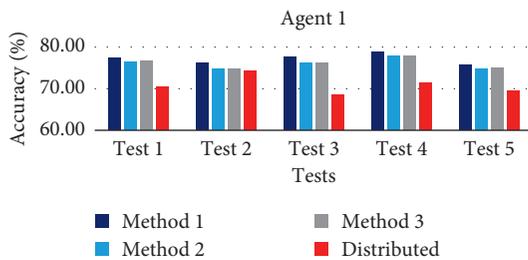


FIGURE 8: Agent 1 results (dataset 1).

testing methods. The proposed system is also over-powered and more accurate than the distributed system. In addition, in the dataset, the voting method is

considered the best among the other methods in most cases. This finding proves the ability of the utilized mutual collaboration approach in enhancing the results of the contacted agents, with the rate 22.56% to 25.77%. The results for the first dataset are illustrated in Table 21 and shown in Figure 17.

The experimental result for the two datasets shows that the proposed system is always more accurate and with enhanced results compared with the distributed system.

TABLE 17: Overall accuracy (dataset 1).

Tests	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	76.19	75.61	74.33	70.21
2	75.40	75.03	74.07	72.47
3	76.77	76.20	75.73	72.45
4	78.27	77.47	77.23	74.62
5	74.92	74.20	74.28	69.86

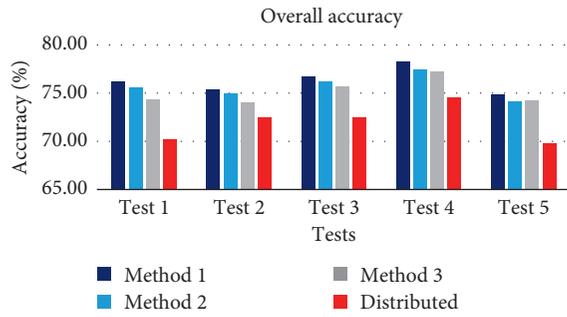


FIGURE 13: Overall accuracy (dataset 1).

TABLE 18: Agent 1 results (dataset 2).

Agent 1				
Test #	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	87.00	85.00	81.00	62.00
2	91.67	88.00	86.01	64.00
3	93.48	93.48	85.51	65.48

TABLE 19: Agent 2 results (dataset 2).

Agent 2				
Test #	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	93.00	92.00	92.00	67.00
2	96.34	95.00	95.67	67.33
3	97.00	97.00	96.25	67.75

TABLE 20: Agent 3 results (dataset 2).

Agent 3				
Test #	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	95.67	92.33	99.67	72.66
2	87.00	87.00	99.00	76.00
3	93.53	93.53	99.50	73.48

Similarly, the experimental results show that the three methods successfully enabled the initiator agent to decide whether to accept the received class label from the other agents. All the methods achieved much better results than the distributed system, and the goal of the proposed system is achieved effectively.

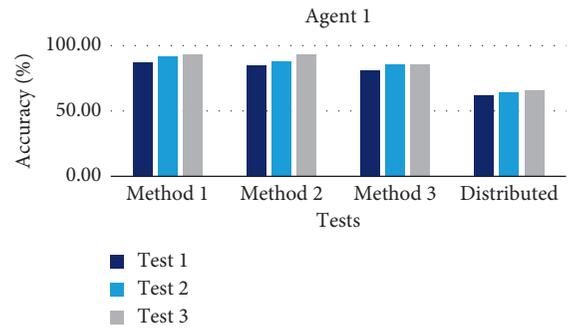


FIGURE 14: Agent 1 results (dataset 2).

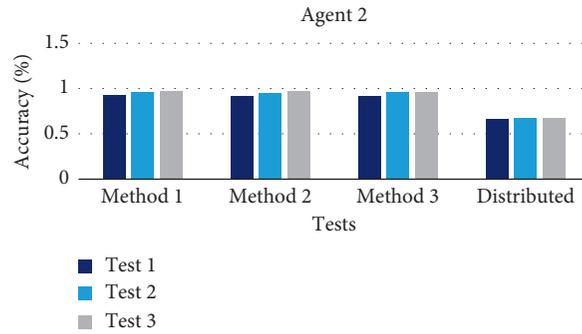


FIGURE 15: Agent 2 results (dataset 2).

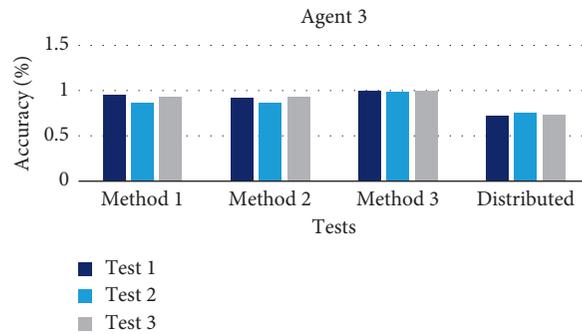


FIGURE 16: Agent 3 results (dataset 2).

TABLE 21: Overall accuracy (dataset 2).

Test #	Method 1 (%)	Method 2 (%)	Method 3 (%)	Distributed (%)
1	91.89	89.78	90.89	67.22
2	91.67	90.00	93.56	69.11
3	94.67	94.67	93.75	68.90

5. Conclusion

Mutual collaboration agent classification using logistic regression, SVM, Bayesian network, and neural network classification techniques is developed and implemented within MAS. In the proposed technique, each agent stores datasets and builds models by selecting the most suitable

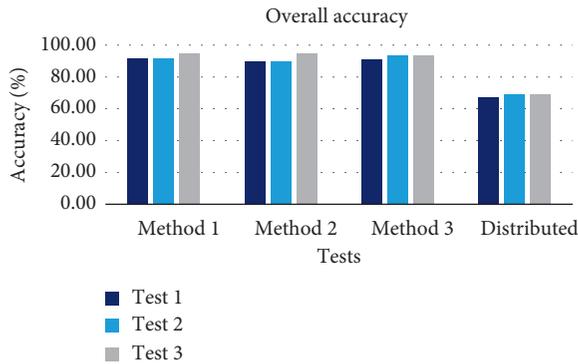


FIGURE 17: Overall accuracy (dataset 2).

classification technique depending on their characteristic dataset. In classifying a new instance, each agent uses their classification technique with reference to the respective prior assumption if the technique is sufficient to classify the new instance and the shared information from other agents when required. When an agent requests information from other agents, the other agents communicate their findings or beliefs (calculated classification) and reply to the requesting agent only if they can aid in the new instance. Otherwise, a message is sent to the requesting agent to inform that the requested agent refrains to respond.

In conclusion, the benefits of MAS include allowing agents to build their individual learned model and control the transfer of their learned information or output results to other agents. This strategy is useful in obtaining improved outputs by combining the results of multiple classifiers and maintaining the accuracy of each agent. Accordingly, agents involved in the collaboration classification tasks promote information diversity and particularity on the basis of their own data when sharing information among one another to enhance the output.

The two datasets that use subsets are assigned to an agent and partitioned into training and testing sets. Each agent builds their model using several classification techniques and selecting the most suitable one. The results were calculated to determine the accuracy of all agents and compared with typical and noncommunicated classification tasks.

The experiment results from the two datasets of the proposed technique are more accurate than those of the noncommunicated classification. This finding proves the capability of the utilized mutual collaboration approach in enhancing the results of the contacted agents, with rates of 5.98% and 25.77% for the first and second datasets, respectively. In addition to accuracy, the proposed technique promotes information diversity and agent particularity, thereby preserving data coverage and meeting the research goal.

Data Availability

The breast cancer data used to support the findings of this study have been deposited in the OpenML repository (<https://www.openml.org/d/77>) and the IEEE Data port

repository (<https://iee-dataport.org/open-access/seer-breast-cancer-data>).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2011.
- [2] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: a review of classification techniques," *Emerging Artificial Intelligence Applications in Computer Engineering*, vol. 160, pp. 3–24, 2007.
- [3] A. A. Ali, P. Vařacha, S. Krayem, P. Žáček, and A. Urbanek, "Distributed data mining systems: techniques, approaches and algorithms," in *Proceedings of the 22nd International Conference on Circuits, Systems, Communications and Computers, MATEC Web of Conferences*, Crete Island, Greece, July 2018.
- [4] J. Gantz and D. Reinsel, "Extracting value from chaos," *IDC Iview*, vol. 1142, pp. 1–12, 2011.
- [5] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: K -anonymity and its enforcement through generalization and suppression," Menlo Park, CA, USA, Technical Report, 1998.
- [6] Q. Tong, X. Li, and B. Yuan, "Efficient distributed clustering using boundary information," *Neurocomputing*, vol. 275, pp. 2355–2366, 2018.
- [7] S. K. Tanbeer, C. F. Ahmed, and B. S. Jeong, "Parallel and distributed frequent pattern mining in large databases," in *Proceedings of the 2009 11th IEEE International Conference on High Performance Computing and Communications*, pp. 407–414, IEEE, Seoul, South Korea, June 2009.
- [8] A. Amir, B. Srinivasan, and A. I. Khan, "Distributed classification for image spam detection," *Multimedia Tools and Applications*, vol. 77, no. 11, pp. 13249–13278, 2018.
- [9] C. Giannella, R. Bhargava, and H. Kargupta, "Multi-agent systems and distributed data mining," in *Proceedings of the International Workshop on Cooperative Information Agents*, pp. 1–15, Springer, Delft, The Netherlands, September 2004.
- [10] N. Obeid and A. Moubaidin, "Towards a formal model of knowledge sharing in complex systems," in *Smart Information and Knowledge Management*, pp. 53–82, Springer, Berlin, Germany, 2010.
- [11] A. Moubaidin and N. Obeid, "Partial information basis for agent-based collaborative dialogue," *Applied Intelligence*, vol. 30, no. 2, pp. 142–167, 2009.
- [12] H. Al-Juaidy, L. A. Jaradeh, D. Qutaishat, and N. Obeid, "On representing and sharing knowledge in collaborative problem solving," in *Contemporary Challenges and Solutions in Applied Artificial Intelligence*, pp. 61–66, Springer, Heidelberg, Germany, 2013.
- [13] K. Kravari and N. Bassiliades, "A survey of agent platforms," *Journal of Artificial Societies and Social Simulation*, vol. 18, no. 1, p. 11, 2015.
- [14] World Health Organization, *National Cancer Control Programmes: Policies and Managerial Guidelines*, World Health Organization, Geneva, Switzerland, 2002.
- [15] A. Banerjee, "Cross-cultural variance of schizophrenia in symptoms, diagnosis and treatment," *Journal of Health Sciences*, vol. 6, no. 2, pp. 18–24, 2012.

- [16] S. M. Bauer, H. Schanda, H. Karakula et al., "Culture and the prevalence of hallucinations in schizophrenia," *Comprehensive Psychiatry*, vol. 52, no. 3, pp. 319–325, 2011.
- [17] E. Januzaj, H.-P. Kriegel, and M. Pfeifle, "DBDC: density based distributed clustering," in *Advances in Database Technology—EDBT 2004*, pp. 88–105, Springer, Berlin, Germany, 2004.
- [18] M. Klusch, S. Lodi, and G. Moro, "Distributed clustering based on sampling local density estimates," in *Proceedings of the IJCAI*, vol. 3, pp. 485–490, Acapulco, Mexico, August 2003.
- [19] N. Chawla, S. Eschrich, and L. O. Hall, "Creating ensembles of classifiers," in *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 580–581, IEEE, San Jose, CA, USA, November–December 2001.
- [20] D. Dash and G. F. Cooper, "Model averaging for prediction with discrete Bayesian networks," *Journal of Machine Learning Research*, vol. 5, pp. 1177–1203, 2004.
- [21] C. Jia, Y. Zuo, and Q. Zou, "O-GlcNAcPRED-II: an integrated classification algorithm for identifying O-GlcNAcylation sites based on fuzzy undersampling and a K-means PCA oversampling technique," *Bioinformatics*, vol. 34, no. 12, pp. 2029–2036, 2018.
- [22] S. J. Stolfo, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, and P. K. Chan, "JAM: Java agents for meta-learning over distributed databases," *KDD*, vol. 97, pp. 74–81, Newport Beach, CA, USA, August 1997.
- [23] H. Kargupta, B. Park, D. Hershberger, and E. Johnson, "Collective data mining: a new perspective toward distributed data mining," *Advances in Distributed and Parallel Knowledge Discovery*, vol. 2, pp. 131–174, 1999.
- [24] V. Sawant and K. Shah, "A review of Distributed Data Mining using agents," *International Journal of Advanced Technology and Engineering Research (IJATER)*, vol. 3, no. 5, pp. 27–33, 2013.
- [25] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2013.
- [26] K. A. Albashiri, F. Coenen, and P. Leng, "EMADS: an extendible multi-agent data miner," in *Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 263–275, Springer, Cambridge, UK, December 2008.
- [27] J. Tozicka, M. Rovatsos, and M. Pechoucek, "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*, p. 96, ACM, Honolulu, HI, USA, May 2007.
- [28] L. Cao, G. Weiss, and P. S. Yu, "A brief introduction to agent mining," *Autonomous Agents and Multi-Agent Systems*, vol. 25, no. 3, pp. 419–424, 2012.
- [29] Fernández-Luna, M. Juan, J. F. Huete, and G. Osorio, "Collaborative document classification: definition, application and validation," in *Proceedings of the 2nd International Conference on Advance Collaborative Networks, Systems and Applications (COLLA 2012)*, pp. 42–49, Venice, Italy, June 2012.
- [30] A. N. Moghadam and R. Ravanmehr, "Multi-agent distributed data mining approach for classifying meteorology data: case study on Iran's synoptic weather stations," *International Journal of Environmental Science and Technology*, vol. 15, no. 1, pp. 149–158, 2018.
- [31] S. Chaimontree, K. Atkinson, and F. Coenen, "A multi-agent based approach to clustering: harnessing the power of agents," in *Proceedings of the International Workshop on Agents and Data Mining Interaction*, pp. 16–29, Springer, Taipei, Taiwan, May 2011.
- [32] A. Kumara, H. S. Bharadwaj, and N. S. Ramaiah, "A survey on K-means algorithm centroid initialization," *SSRN Electronic Journal*, 2019.
- [33] I. Rojas, O. Valenzuela, F. Rojas, and F. M. Ortuño, Eds., *Bioinformatics and Biomedical Engineering: 7th International Work-Conference, IWBBIO 2019, Granada, Spain, May 8–10, 2019, Proceedings, Part I*, Vol. 11465, Springer, Berlin, Germany, 2019.
- [34] V. Bewick, L. Cheek, and J. Ball, "Statistics review 14: logistic regression," *Critical Care*, vol. 9, no. 1, p. 112, 2005.
- [35] I. Steinwart and A. Christmann, *Support Vector Machines*, Springer Science & Business Media, Berlin, Germany, 2008.
- [36] X. Zheng, W. Yao, Y. Xu, and Y. Zhao, "Algorithms for complex systems reliability analysis based on Bayesian network," in *Proceedings of the 13th International Conference on Applications of Statistics and Probability in Civil Engineering*, Seoul, South Korea, May 2019.
- [37] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions*, vol. 30, no. 4, 2000.
- [38] A. Moubaidin, F. Shawarbeh, and N. Obeid, *Using Intelligent Agents in E-Learning*, International Information Institute Tokyo, Tokyo, Japan, 2013.
- [39] N. Obeid and S. Al-Areqi, "Using agents for dynamic components redeployment and replication in distributed systems," in *Contemporary Challenges and Solutions in Applied Artificial Intelligence*, pp. 19–25, Springer, Heidelberg, Germany, 2013.
- [40] N. Obeid and A. Moubaidin, "Towards a formal model of knowledge sharing in complex systems," in *Smart Information and Knowledge Management*, pp. 53–82, Springer, Berlin, Germany, 2010.
- [41] A. Moubaidin and N. Obeid, "Dialogue and argumentation in multi-agent diagnosis," in *New Challenges in Applied Intelligence Technologies*, pp. 13–22, Springer, Berlin, Germany, 2008.
- [42] A. Moubaidin, I. Salah, and N. Obeid, "A temporal modal defeasible logic for formalizing social commitments in dialogue and argumentation models," *Applied Intelligence*, vol. 48, no. 3, pp. 608–627, 2018.
- [43] N. Obeid, "Towards a model of learning through communication," *Knowledge and Information Systems*, vol. 2, no. 4, pp. 498–508, 2000.
- [44] N. Obeid and A. Moubaidin, "On the role of dialogue and argumentation in collaborative problem solving," in *Proceedings of the 2009 Ninth International Conference on Intelligent Systems Design and Applications*, pp. 1202–1208, IEEE, Pisa, Italy, November 2009.
- [45] A. Moubaidin and N. Obeid, "Towards a formal model of knowledge acquisition via cooperative dialogue," in *Proceedings of the ICEIS*, vol. 5, pp. 182–189, Funchal, Portugal, June 2007.
- [46] A. Edwardes, D. Burghardt, and M. Neun, *FIPA Communicative Act Library Specification. Foundation for Intelligent Physical Agents*, Foundation for Intelligent Physical Agents, University of Maine, Orono, Maine, 2008, [http://www.fipa.org/specs/fipa00037/SC00037\].html.30.6.\(2004\)](http://www.fipa.org/specs/fipa00037/SC00037].html.30.6.(2004)).
- [47] J. J. Odell, H. V. D. Parunak, and B. Bauer, "Representing agent interaction protocols in UML," *International Workshop on Agent-Oriented Software Engineering*, Springer, pp. 121–140, Springer, Limerick, Ireland, June 2000.

- [48] F. Bellifemine, A. Poggi, and G. Rimassa, "Developing multi-agent systems with a FIPA-compliant agent framework," *Software: Practice and Experience*, vol. 31, no. 2, pp. 103–128, 2001.
- [49] Y. Labrou, T. Finin, and Y. Yun Peng, "Agent communication languages: the current landscape," *IEEE Intelligent Systems*, vol. 14, no. 2, pp. 45–52, 1999.
- [50] N. Bhatia, "Survey of nearest neighbor techniques," 2010, <http://arxiv.org/abs/1007.0085>.
- [51] O. L. Mangasarian and W. H. Wolberg, *Cancer Diagnosis via Linear Programming*, University of Wisconsin-Madison Department of Computer Sciences, Madison, WI, USA, 1990.
- [52] J. Vanschoren, "BNG (breast-cancer, nominal, 1000000)," 2014, <https://www.openml.org/d/77>.
- [53] J. Teng, "SEER breast cancer data," 2019, <https://iee-dataport.org/open-access/seer-breast-cancer-data>.



Hindawi

Submit your manuscripts at
www.hindawi.com

