IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 16, 2021, accepted April 14, 2021, date of publication April 20, 2021, date of current version May 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074125

Multi-Agent System Combined With Distributed Data Mining for Mutual Collaboration Classification

MAIS HAJ QASEM[1], NADIM OBEID[1], AMJAD HUDAIB[1], MOHAMMED AMIN ALMAIAH[2], ALI AL-ZAHRANI[2], AND AHMAD AL-KHASAWNEH[3]

[1]King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan
[2]College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia
[3]Department of Computer Information System, The Hashemite University, Zarqa 13115, Jordan

Corresponding authors: Mohammed Amin Almaiah (malmaiah@kfu.edu.sa) and Mais Haj Qasem (mais_hajqasem@hotmail.com)

ABSTRACT Distributed Data Mining (DDM) has been proposed as a means to deal with the analysis of distributed data, where DDM discovers patterns and implements prediction based on multiple distributed data sources. However, DDM faces several problems in terms of autonomy, privacy, performance and implementation. DDM requires homogeneity regarding environment, control, administration and the classification algorithm(s), and such that requirements are too strict and inflexible in many applications. In this paper, we propose the employment of a Multi-Agent System (MAS) to be combined with DDM (MAS-DDM). MAS is a mechanism for creating goal-oriented autonomous agents within shared environments with communication and coordination facilities. We shall show that MAS-DDM is both desirable and beneficial. In MAS-DDM, agents could communicate their beliefs (calculated classification) by covering private and non-sharable data, and other agents decide whether the use of such beliefs in classifying instances and adjusting their prior assumptions about each class of data. In MAS-DDM, we will develop and use a modified Naive Bayesian algorithm because (1) Naive Bayesian has been shown to be the most used algorithm to deal with uncertain data, and (2) to show that even if all agents in MAS-DDM use the same algorithm, MAS-DDM preforms better than DDM approaches with non-communicating processes. Point (2) provide an evidence that the exchange of information between agents helps in increasing the accuracy of the classification task significantly.

INDEX TERMS Classification, FIPA standards, multi-agent system, Naïve Bayesian.

I. INTRODUCTION

In the last few years, we have witnessed a tremendous increase in distributed data, cloud computing, wide usability of micro-processor devices (e.g., mobiles and sensors), and data that is generated or obtained at multiple data acquisition devices. Such trend makes it very difficult to have all the data transferred into centralized data warehouse that is needed by the traditional classification methods which utilize the stored training dataset to establish a model. Furthermore, some secured data, such as financial records and medical data, cannot be transferred or shared, as data during transformation might be subject to exposure. Moreover, data centralization is also affected by bandwidth limitation, where the transfer of large data volumes over networks requires considerable time and financial resources. Subsequently, to be effective, the data classification task has to naturally be distributed which led to the emergence of Distributed Data Mining (DDM) environment [1].

DDM discovers patterns and implements prediction based on multiple distributed data sources. DDM has dealt with the issue of requiring that data be located in a single unit in order to execute its process, where DDM could mine data sources wherever their physical locations are. The decentralized architecture of DDM can reach every networked business; hence, DDM has become a key component of knowledge-based systems. The business intelligence market is one of the fastest growing and most profitable areas in the software industry, which consequently leads to the popularization of DDM. Subsequently, various DDM techniques has

The associate editor coordinating the review of this manuscript and approving it for publication was Yichuan Jiang.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

been developed in the literature, such as, distributed clustering [2], distributed frequent pattern mining [3] and distributed classification [4].

Overall, DDM techniques can be classified into two groups on the basis of the level of information sharing. First; low-level DDM where each site is trained on the basis of its own data. Afterward, all the sites are given the same task to solve or deal with, which can be an instance to be classified, clustered, or analyzed. All the results of these sites/processes are combined to produce an output by the control of the global administration. Second; high-level DDM where each site/process shares its learned model with the global model to produce a single-learning model for mining the input data.

What characterizes DDM (whether at low- or high-level) is that most of the studies which used DDM (low- or high-level) work in the same environment or organization and are under the control of the same administration, including their agreement on the same classification algorithm, and there is no communication between the classifying sites/processes. Communication is only allowed between a classifying site/process and the global administration. Consequently, several problems can occur. These problems include:

1. In high-level DDM, distributed sites/processes have tremendous increase in their data that may require them to retrain their model in a short time. They may also be required to send their updated model to the global one as soon as possible. If the model is submitted late as a result of network interruptions and/or other possible faults, then the whole system could be affected.
2. In low-level DDM, all sites/processes are given the same case to be classified, and the results are then combined to produce an output. Therefore, all sites/processes are involved in the decision making. However, it is possible that certain sites specialized, and their dataset may allow them to classify a case, which may be uncommon from the archived cases of other sites. In such a situation, the final decision to produce an output may prove to be difficult to make or not accurate.

From the previous discussions, we single out, in particular, two essential problems related to autonomy and privacy as the main concern in this paper. DDM requires homogeneity regarding environment, control, administration and the classification algorithm(s). Such requirements are too strict and inflexible, where traditional DDM methods (low- and high-levels) cannot be applied for various applications that require distributed classification processes like medical clinic and cancer hospital due to one or more of the problems mentioned above. This is, because each medical clinic or cancer hospital has its own samples and prior beliefs about cases and the corresponding diagnosis, which may differ based on the region and the culture and belongs to a different environment or organization. Furthermore, they cannot share their local model due to possible exposure and patient privacy issues, but they still need collaboration in classifying symptoms that appear uncommon to one of them that in wise and intelligent way.

In this paper, we propose the employment of a Multi-Agent System (MAS) to be combined with DDM (MAS-DDM) to solve the problems we mentioned above for data intensive applications [5]. MAS can be defined as a collection of agents with their own problem-solving capabilities that can interact to reach an overall goal [6]. 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 pro-active behaviors [7].

MAS is appropriate for distributed problem solving because it allows the creation of autonomous, goal-oriented entities/agents that operate in shared environments with coordination and communication capabilities. This mechanism is beneficial for DDM as it allows us to combine and integrate different distributed clustering, prediction, and classification methods. In MAS environments, there are no assumptions regarding global control, global administration, and synchronization like in DDM. Thus, agents in MAS are assumed to operate with incomplete information or capabilities in solving problems. 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 requests 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 [8].

MASs frequently handle complex applications that need distributed problem solving. Meanwhile, DDM is a complex system focused on data mining processes and resource distribution over networks. Scalability lies at the core of DDM systems. Given that a system's configurations may sometimes change, DDM system design looks at many details regarding software engineering, such as extensibility, reusability, and robustness. Therefore, the characteristics of MAS are favorable for DDM systems. In addition, the decentralization of MAS property fits the DDM requirements well. A mining strategy is executed at each data site specifically for the certain data domain. However, data miners may prefer to test other existing or new strategies. A data site should conduct testing on several strategies for further analysis and seamlessly integrate with external methods [10].

The contribution in this paper involves collaborative distributed classification system that implemented in a MAS, where MAS-DDM system will inherit all the powerful properties of agents and acquire favorable features. MAS-DDM consists of a set of autonomous agents in shared environment with communication and coordination facilities, and allowing each distributed site to build their individual learned model, have direct communication between each sites, and to have its own classification form and decides to request information from other sites or not even if they are in different environment or organization. Agents in MAS-DDM requests help from other agents to collaborate their information to classify a new uncommon case from its stored cases and is difficult to classify locally to one of the agents.

The case study, in this paper, which will be focused on is cancer hospitals worldwide, that requires handling uncertainty cases in classification, and for that MAS-DDM will be using Naive Bayesian classification algorithm which will called MAS-DDM-NB, where the Naive Bayesian is the most capable of handling uncertainty cases in classification [11].

The proposed MAS-MC-NB is implemented and compared with normal DDM algorithm without any collaboration among agents, and traditional low- and high-level DDM where all the agents have no communication also with each other, to show that MAS-DDM-NB is more efficient while each site have a own control. This paper is organized as follows. Section 2 reviews works closely related with the use of MAS and DDM.Section 3 presents our proposed system. Section 4 presents the experimental results. The conclusion is provided in Section 5.

II. 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. 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 [12] 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 [13]. In classification, a clear example of such approach is bagging and boosting approaches, which allow multiple classifiers to operate on different sets of data [14]. Bayesian classifiers are operated in distributed environment by averaging the local model of the distributed sources to obtain a global one [15], [16].

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 [17]and BODHI [18], were proposed for this purpose. Accordingly, researchers on DDM have revealed the advantages of applying MAS in organizing, implementing, and controlling distributed sources.

Various MAS-based DDM approaches have been proposed. EMADS [19] was proposed as MAS-based on the ensample classification, which provides weights for each distributed classifier or select ones 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 [20], [21]. A brief overview of these approaches is described in [22].

MAS-based DDM allows agents to establish 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 non-sharable 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 [23]. 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 [24]. 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 non-sharable data. Table 1 summarizes the advantages of the discussed literature.

III. MAS-DDM-NB

MAS-DDM-NB is a general approach that developed to facilitate collaborative classification between distributed agents worldwide. MAS-DDM-NB approach consists of multiple agents, each agent represents one of the distributed sites that have its own dataset and prior beliefs on cases, which may differ from site to site. Therefore, each agent builds their individual learned model, have direct communication between each site, and to have its own classification form and decides to request information from other sites or not.

Agent in various regions should collaborate their information with another agents that is about to classify a new uncommon case from its stored cases and is difficult to classify locally to one of the agents. Therefore, 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 threshold "$\boldsymbol{t}$". In such a case, the classification model used by the local agent is considered sufficient to classify the input instance.

TABLE 1. Summary of the related work.

Reference	Distributed-Sources	Central-based Collaboration	Mutual Collaboration	Information Hiding	Particularity	MAS
Distributed Clustering [14]	√	√			√	
Distributed Clustering [15].	√	√			√	
Distributed Classification [16].	√	√			√	
Distributed Classification (Bayesian) [17]	√	√			√	
JAM [18]	√	√			√	
BODHI [19]	√	√			√	
EMADS [20]	√	√			√	√
MAS-based Classification [21]	√	√			√	√
MAS-based Classification [22]	√	√			√	√
MAS-based Classification [24]	√	√	√		√	√
MAS-based Clustering [25]	√	√	√	√	√	√

However, if the classification output does not reach the specific threshold "t", 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 "t". After the agents sent their result to the requested agent a creative method that appropriateness to Bayesian network that tested to help the requested hospital agent decide whether to accept the received class label from other agents, and which class label that more accurate to take that it will discussed later in this paper. Figure 1 illustrated the proposed system.

The following three phase presents the implementation of the proposed work in phases. The first phase demonstrates the dataset preparation. The second phase describes the Bayesian classification algorithm that each agent applies to build their model and explains the feedback processing that to help the initiator decide whether to accept the received class label from other agents, whereas the MAS protocols that agents use in their interactions and communications are described in the third phase. Appendix in the last paper showed example for our proposed work.

First Phase–Data Preparation: The case study, in this paper, which will be focused on is cancer hospitals worldwide, where each hospital agent stores its datasets related to patient cases and prior beliefs about cases and the corresponding diagnosis, which may differ based on the region and the culture and belongs to a different environment or organization. In normal scenario a different cancer dataset for the same disease must be collected from various sources. But this approach cannot be sustained in a long research stage because of the scarcity of datasets in the open online repository. Therefore, the same dataset is distributed through a k-mean clustering algorithm to verify the most effective cluster number and select the most suitable number of agents that can communicate and work with one other. K-mean is one of the most popular clustering algorithms that allows the specification of the required cluster number [25].

After clustering the dataset, each cluster given to different agent, and each one of them carries out feature selection on the given cluster. Feature selection is the process in which the features that contribute most to a target prediction variable or output are automatically or manually selected. Irrelevant features in data can reduce model accuracy and cause these models to learn on the basis of irrelevant features. Selecting features after the agent is given a cluster is important. Considering that each agent worldwide may require features that vary from those required by other agents, datasets are distributed vertically and not horizontally. The proposed system is applied on two datasets for experimentation and benchmarking.

1. 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 [37], representing instances for multiclass breast cancer detection classification [38]. This dataset consists of 1 million instances, which are described by 13 attributes to predicting if the patients have cancer or not, where the class label attributes including two labels that are no-recurrence-events and recurrence-events.
2. 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 [39]. This dataset consists of 4024 instances described by nine attributes to determine whether or not patients will survive in the form of cancer, where the class label attributes include two live or dead labels.

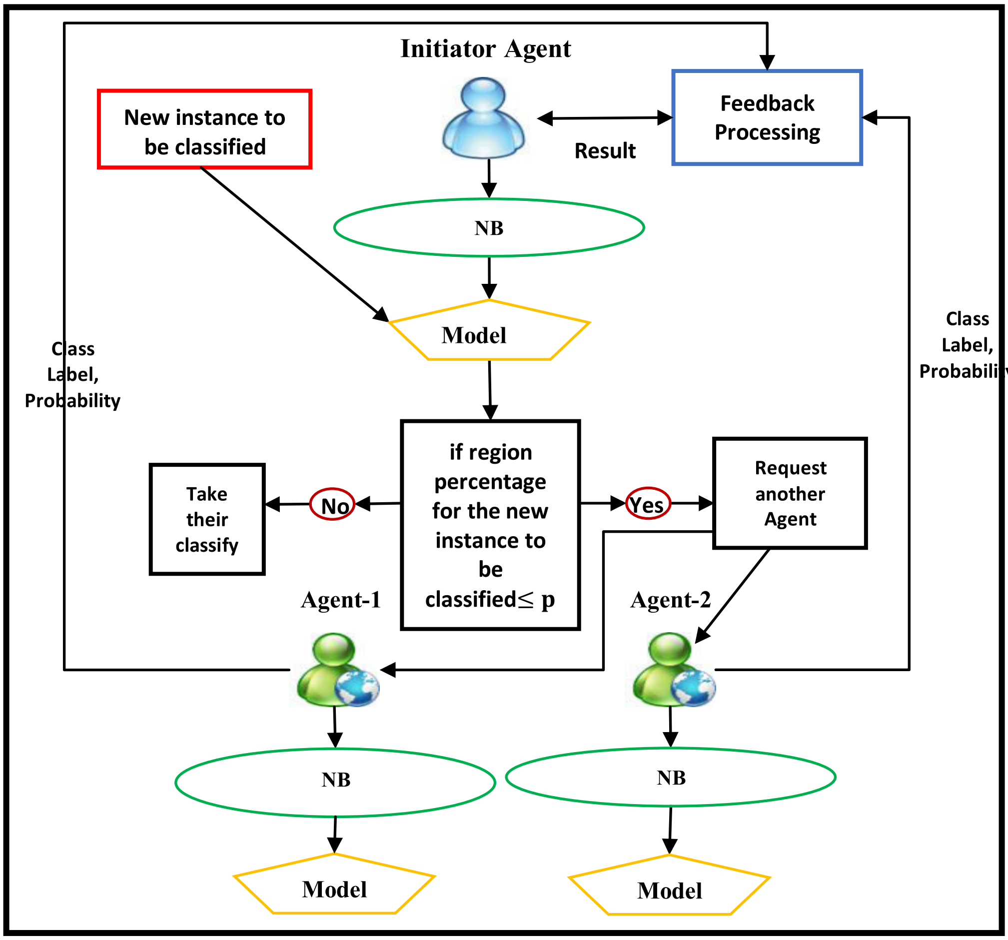

FIGURE 1. Proposed collaborative classification framework.

Second Phase–Collaborative Naive Bayesian Algorithm: The case study, in this paper, which will be focused on is cancer hospitals worldwide, that requires handling uncertainty cases in classification, and for that MAS-DDM will be using Naive Bayesian classification algorithm, where the Naive Bayesian is the most capable of handling uncertainty cases in classification [11]. The datasets are plotted, and the results indicate uncertain data, as shown in Figure 2. The Naive Bayesian (NB) model is found to be the most capable of handling uncertainty cases in classification and showing the results of involved agents. Hence, choosing this model is efficient [25].

Data uncertainty arises naturally in many applications due to various factors. These factors include the random nature of the physical data generation and collection process, measurement and decision errors, unreliable data transmission and stale data such as sensor networks, data values that continuously change, and outdated recorded information. Uncertainty may also be caused by repeated measurements [28].

- Uncertainty is an obstacle in building classification modeling, as described in the following:
- Uncertainty cases are those in which objective probability distribution is absent because of limited data availability. These cases depend on personal experiences derived from individuals' personal judgment or own experience, hence the term "personal probability distribution."
- Uncertainty emerges when human users or robots need to make a decision but lack a full perception of the surrounding environment.
- Some cases involve an uncertain prediction of probability based on a given input.
- Uncertainty cases sometimes involve the same data values but different outputs.

Therefore, uncertainty cases require unconventional techniques for building classification models. Despite the abundance of classification algorithms, building classification based on uncertain data remains a great challenge. The Bayesian network is the most capable of handling uncertainty cases in classification. In the presence of complicated relationships between independent variables, the Bayesian network represents the common probability distribution for a set of random variables that share potentially interrelated causal relationships [29].

The NB classification algorithm is highly appealing because of its simplicity, elegance, and robustness. It is one

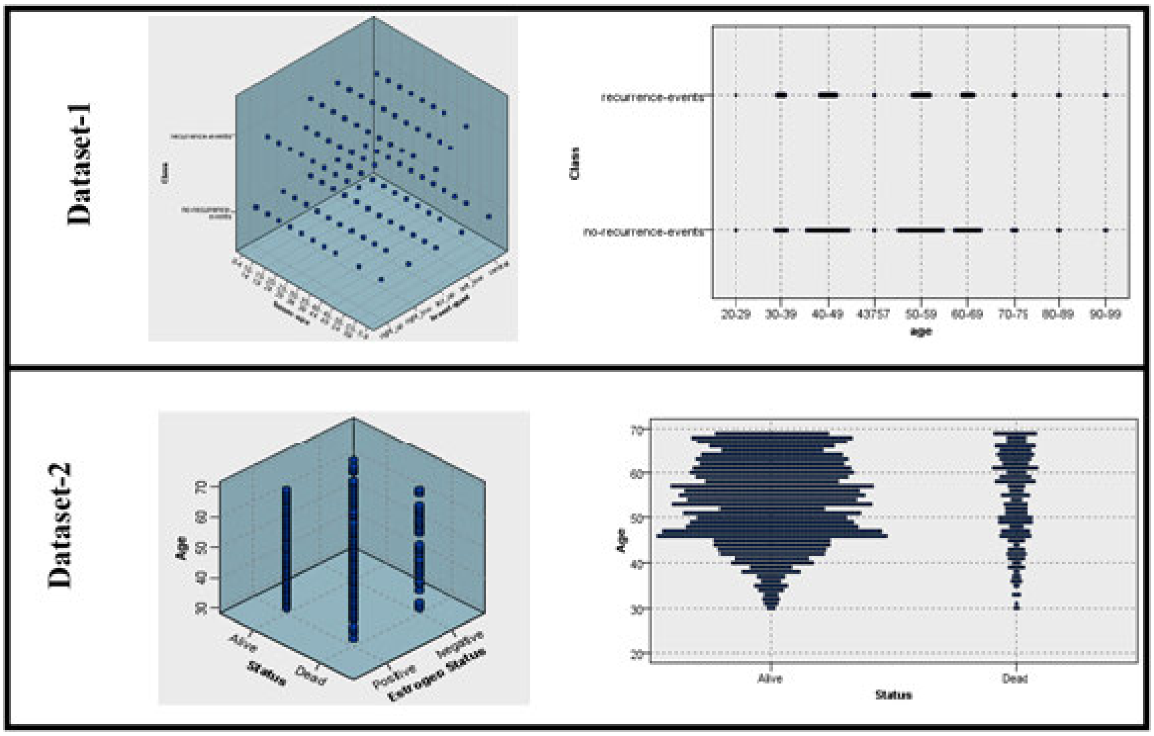

FIGURE 2. Datasets plotting.

of the oldest formal classification algorithms, and it is often surprisingly effective [30]. A large number of modifications have been introduced by statistical, data mining, machine learning, and pattern recognition communities in an attempt to make the algorithm increasingly flexible. It is widely used in various areas, including text classification and spam filtering.

In a typical classification task in NB, class probability depends on the frequency or stochastic analysis of each class. Conditional probabilities are calculated on the basis of the ratio between the joint presence of the attributes and the classes to the presence of the class itself. Classification, in general, and Bayesian classification, in particular, exhibit poor performance in the classification process when the number of samples representing a given class is low [31]. One of the solutions, which can be easily implemented, is to combine the probabilities and conditional probabilities of different sources; this solution works exactly the same when all data are combined in a single source [32]. Nevertheless, this process eliminates the specialty and privacy of distributed sources, which are highly important in many applications, such as disease diagnosis based on the region of the data acquired [33], [34].

While there are various classifiers exists in the literature with robust performance, Bayesian classifier, has properties that make it suitable for the intended distributed classification task, accordingly it is selected to be integrated with the distributed agents.

- **First**: Each agent should be able to incorporate knowledge acquired from other agents, Bayesian as a statistical classifier, has the ability to integrate other source of knowledge aside from the training data. SVM, for example is not capable of incorporate such information in the classification process.
- **Second**: Each agent should be able to deal with missing values, as these values will be incrementally added to the model, Bayesian, unlike other classifiers, such as SVM, has the ability 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 the whole training samples, Bayesian has the ability to share and hide some of the components of the models, which are the priori probability, conditional probability and class probability.
- **Finally**, the scalability is required for the whole system to allow adding new classes and features without re-constructing models. Bayesian is scalable compared to other classifiers such as SVM and decision tree [35].

To facilitate collaborative agent classification, a collaborative MAS-DDM-NB is developed. Each agent stores its samples and establishes its prior assumption based on these data. In the classification of new data item, each agent uses NB classifier with reference to its own prior assumption and uses shared information from other agents when required. Accordingly, agents involved in the collaboration classification tasks promote information diversity and particularity based on their own data and share information among one another to enhance the results. 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:

- Sharing information about data among agents to facilitate agent collaboration in the classification task;
- Promote information diversity among agents and agent particularity based on its own data;
- Promote information privacy among agents by sharing limited data about the constructed model at each agent;
- Enhance agents results by incorporating inputs from other agents;
- Ensure limited communication overhead among the distributed agents and prevent data reallocation and centralization; and
- Ensure low processing cost by processing data blocks at each agent independently.

The components of the proposed collaborative classification technique are discussed in the following subsections.

A. MUTUAL COLLABORATIVE NAIVE BAYESIAN CALCULATION

The principles of the Bayesian-based classification remain the same with the collaborative classification when sharing joint probabilities among collaborative agent. Hence, for classification task with discrete random variable and nominal attributes, distributed naive Bayesian classification is implemented using the maximum a posterior (MAP), as given in Equation 1:

$$\hat{y} = \underset{k \in \{k1,k2,\ldots,kn\}}{\operatorname{argmax}} \; p\left(C_k\right) \prod\nolimits_{i=1}^{n} P(X_i|C_k) \tag{1}$$

where p(C_k), is the probability of the class k, and p($X_i|C_k$), is the conditional probability of attribute value Xi with class k. Accordingly, MAP selects the class that maximizes the posterior based on conditional and class probabilities. These probabilities are calculated at each agent as model building prior to any classification or information sharing process. p(C_k), and p($X_i|C_k$), are calculated using Equations 2 and 3, respectively. Another important probability p(X_i) is also calculated at each agent using Equation 4:

$$p\left(C_k\right) = {}^{n_k}\!/_{n} \tag{2}$$

$$p\left(X_i|C_k\right) = {}^{n_{ik}}\!/_{n} \tag{3}$$

$$p\left(X_i\right) = {}^{n_i}\!/_{n} \tag{4}$$

where n is the total number of instances at the agent, n_k is the number of instances with the class value k, n_i is the number of instances with the attribute value i, and n_{ik} is the number of instances with the attribute value i and class value k.

Given an instance to be classified is introduced to the local agent, the agent calculates the probability of each class independently without any collaboration if and only if the probability of the produced classification output is above some threshold t. In such case, the classification model at the local agent is considered sufficient to classify the input instance. If the best class, as calculated by MAP, shows low probability that is below t, then the agent requests information from other agents. This request is followed with calculation in a set of processes.

First, in calculating class probability and request collaboration, the probabilities of all class labels that are known to the local agent, $\hat{y}$, are calculated when a set of attribute values is provided to that agent, as given in Equation 1.

In case that the probabilities of all these classes do not reach a specific threshold, which differentiates between the asserted and randomized values, the agent requests information from other agents. The other agents send the following probabilities to the initiator:

- Class labels C_k above the threshold;
- Joint probability of the set of attributes conditionally dependent on each class that is Above the threshold $\prod_{i=1}^{n} P(X_i|C_k)$; and
- Joint probabilities of the attribute set $\prod_{i=1}^{n} P(X_i)$.

To ensure the privacy of agent data, none of the following values used to calculate the shared information are exposed:

- Actual probability value calculated for $P(C_k|X_i)$;
- Number of instances, in which C_k and X_i occur jointly;
- Number of instances, in which C_k is presented in the dataset;
- Total number of instances stored in that agent.

Second, in feedback processing, the initiator agent compares its most remarkable joint attribute probability that is conditionally dependent on its corresponding probability received by other agents and the received joint attribute probability with its corresponding probability. With inputs from multiple agents, the agent with the most desirable combined probability, that is, $\prod_{i=1}^{n} P(X_i|C_k) \times \prod_{i=1}^{n} P(X_i)$, which indicates the most remarkable knowledge about the attributes and the given class, is considered. On the basis of the selected response, the initiator accepts the received class label only if the received probabilities are above its own probabilities to prevent agents with high-class but low-attribute probability from strongly influencing the results. In the proposed technique with distributed data among multiple agents, each agent shares the information highly considered correct according to its knowledge of the specific attributes, specific class, and other agents; this information is highly considered missing according to its knowledge of the specific attributes and class.

Third, in model updating, 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 is used to update the conditional probability, as given in Equation 5.

$$P\left(X_i \mid C_k\right) = \prod\nolimits_{i=1}^{n} P'(X_i|C_k). \tag{5}$$

Moreover, the attribute and class probabilities for each involved attribute value and class label are increased by one divided by the number of instances saved at the local agent. This process is re-established in the local model, which is similar to the effect of adding new training instance to that model.

Hence, the conditional probability shows no dependence on the number of instances but on the received probability, without affecting the probability of the attributes. However, this update will influence the classification of such attribute toward the underlying class.

B. COLLABORATIVE VALIDATION WITH DISTRIBUTED DATA

The proposed collaborative Bayesian classifier is identical to the typical one. Nevertheless, the probabilities used to update and calculate classifier labels during sharing are different from the typical ones.

An example of data distributed horizontally among two parties is considered; in this example, each party consists of different set of classes all described using the same set of attributes. The $P(X_i)$ values will be high. Each party constructs its model by calculating the probability of each class, the probability of each attribute value, and the conditional probability of each attribute value with each class. A set of input attribute values is provided to the first party, and three possibilities are available for this input, which are described as follows:

- **Input is a set of values that are frequently occurring jointly with a specific class label in the first party.** Accordingly, the $\prod_{i=1}^{n} P(X_i|C_k)$ and $P(C_k)$ values will be high. In this case, according to the collaborative technique described above, the calculated value, that is, $P(C_k \mid X_i)$, will be high, and the class label will be assigned locally without collaboration from the second party. However, if the same set of attributes is assigned different class labels in the second party, then this assignment must be considered in the proposed technique to preserve the particularity of each party by assigning the class label, which is considerably common to that party.
- **Input is a set of values that frequently occur in the first party but jointly occur with multiple class labels.** Accordingly, the $\prod_{i=1}^{n} P(X_i|C_k)$ value will be low, but the $P(X_i)$ value will be high. In this case, the calculated value of $P(C_k \mid X_i)$ will not be high, and the first party will request collaboration from the second party. In this case, if the values of $\prod_{i=1}^{n} P(X_i|C_h)$ and $P(X_i)$ in the second party are higher than those of $\prod_{i=1}^{n} P(X_i|C_k)$ in the first party, then the class is assigned on the basis of the second party because of the frequent appearance of the feature set X_i and good joint probability that links the attributes to a specific class. Additionally, although both parties display different values for $P(X_i)$, when the (X_i) value and the received value $\prod_{i=1}^{n} P(X_i|C_h)$ are high, the next identical set of attributes will be assigned class label C_h because it is higher than any conditional probability of the first party.
- **Input is a set of values that do not frequently occur in the first party, and the joint probability with all the classes is low.** Accordingly, the $\prod_{i=1}^{n} P(X_i|C_k)$ and $P(C_k)$ values will be low. In this case, the calculated $P(C_k \mid X_i)$ value will also be low. Hence, if the $\prod_{i=1}^{n} P(X_i|C_h)$ and $P(X_i)$ values in the second party are high, then the class is assigned on the basis of the second party because of the frequent appearance of the feature set X_i and high joint probability that links the attributes to a specific class. In addition, given that the received value $\prod_{i=1}^{n} P(X_i|C_h)$ is high, the next identical set of attributes will be assigned class label C_h because it is higher than any conditional probability of the first party. In the second and third cases, if the second party also exhibits low probabilities, then both parties possess insufficient knowledge of the input set of attributes. Accordingly, the first party will receive no value from the second one. This party will assign label locally to preserve the particularity of each party by assigning the class label that is remarkably common to that party.

Overall, in the horizontally distributed data, the proposed technique considers collaborative inputs to enhance the output classification results and preserve particularity of different parties when both parties display similar knowledge of the input set of attributes. Although each party may assign different class labels, this party preserves the particularity of each party by assigning the class label that is considerably common to that party.

In vertically distributed data, all the parties present the same set of classes, with each party described using different values of the same set of features. Otherwise, the $P(X_i)$ values will be low in one party and high in another similarly. $\prod_{i=1}^{n} P(X_i|C_k)$ will be low in one party and high in another. An input set of attribute values is provided to the first party, and three possibilities for this input are as follows:

- **Input is a set of values frequently occurring in the first party jointly with a specific class label.** Accordingly, the $\prod_{i=1}^{n} P(X_i|C_k)$ and $P(C_k)$ values will be high. In this case, only the local party can assign the correct class label because the second party is expected to exhibit low probabilities. Consequently, the class label will be assigned locally without collaboration from the second party.
- **Input is a set of values frequently occurring in the first party but jointly occurring with multiple class labels.** Accordingly, the $\prod_{i=1}^{n} P(X_i|C_k)$ value will be low, whereas that of $P(X_i)$ will be high. In this case, the calculated value, $P(C_k \mid X_i)$, will not be high, and the first party will request collaboration from the second party. Thus, $\prod_{i=1}^{n} P(X_i|C_h)$ and $P(X_i)$ in the second party will probably be low because data are distributed vertically. The class is assigned locally because both parties show low attribute probabilities. The local party is given advantage with its high $P(X_i)$ value.
- **Input is a set of values not frequently occurring in the first party, and the joint probability with all the classes is low.** Accordingly, $\prod_{i=1}^{n} P(X_i|C_k)$ and $P(C_k)$ values will be low. In this case, the calculated $P(C_k \mid X_i)$ value will be low. Consequently, the $\prod_{i=1}^{n} P(X_i|C_h)$ and $P(X_i)$ values in the second party will be high. The class is assigned on the basis of the second party because of the

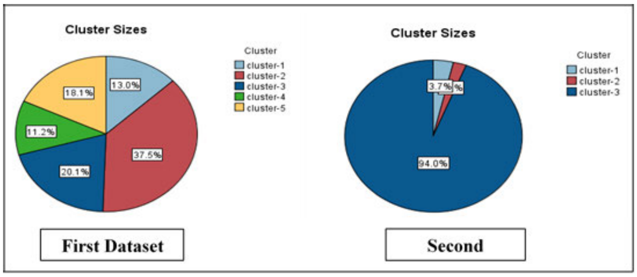

FIGURE 3. Illustrates the percentage for each cluster for the two datasets.

FIGURE 4. Feature selection results.

frequent appearance of the feature set X_i and high joint probability that links the attributes to a specific class. Moreover, given that the received value $\prod_{i=1}^{n} P(X_i|C_h)$ is high, the next identical set of attributes will be assigned class label C_h because it is higher than any conditional probability of the first party.

As noted in the vertically distributed data, the proposed technique considers high collaboration because each party is knowledgeable of the specific input set, which makes the proposed technique similar to the typical classification approach. Furthermore, unlike the typical classification, the proposed technique allows limited sharing of information. Thus, private data remain secured and unexchanged among the collaborative parties.

Third Phase–Mas Protocol: In this phase MAS-DDM-NB is implemented by using that same MAS Protocol that we used in our pervious paper that found in [36].

Accordingly, the resulting data are distributed among the agents, and each agent is given training and testing sets. Each agent applies the NB classifier, as illustrated in Figure 5. Table 2 summarizes the model accuracy for each agent for the first and second datasets.

IV. EXPERIMENTS AND RESULTS

To validate the applicability of the proposed technique and estimate the performance of the involved collaborative classification, we develop a MAS for classification tasks using JADE. 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. Where when a large number for clustering was tested for the both dataset the feature lost worth, and there is no overlap between the agent where most of important feature is dropped in the feature selection stage. Figure 3 illustrates the percentage for each cluster for the 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 4 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

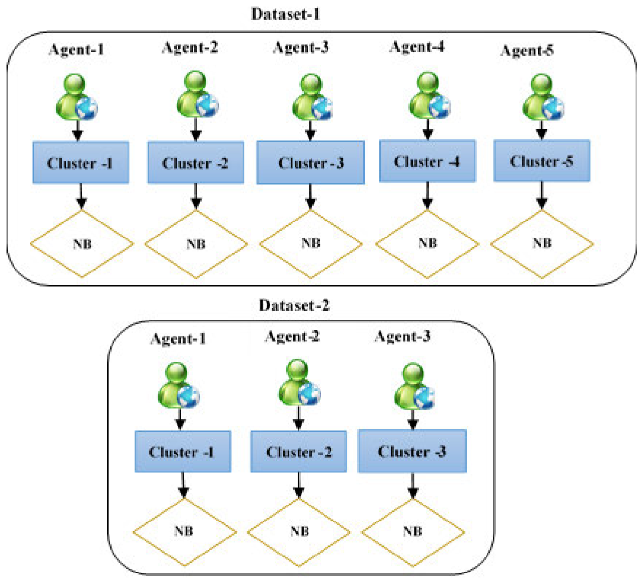

FIGURE 5. Data distribution among the agent.

TABLE 2. Model accuracy for each agent.

Dataset-1	
Agent #	Model Accuracy
Agent 1	75.1%
Agent 2	78.06%
Agent 3	79.39%
Agent 4	75.17%
Agent 5	81.89%
Dataset-2	
Agent #	Model Accuracy
Agent 1	90.27%
Agent 2	86.91%
Agent 3	94.79%

agent applies the NB classifier, as illustrated in Figure 5. Table 2 summarizes the model accuracy for each agent for the first and second datasets.

MAS for classification task is developed using JADE that evolved full-featured FIPA platform. After each agent builds their model, the MAS-MCC-NB is executed to validate the process sequence and communication.

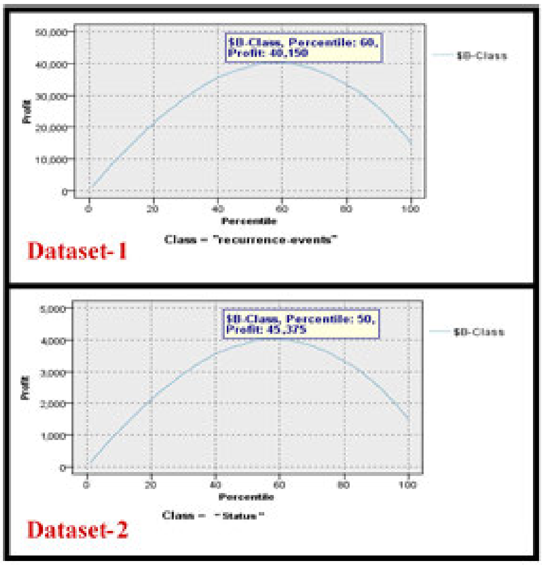

FIGURE 6. Cumulative profit charts.

MAS-MCC-NB works on cancer datasets, where the most important factor in this filed is to minimize the risk of incorrect detection, and that is why we must define a carbide threshold for their classifier that indicates whether their model can calculate the instance without or without any collaboration. For that reason, we used a cumulative profit charts to detect the well-defined peak somewhere in the middle of the chart that indicate the lowest right probability of the model as shown in Figure 6 for the two datasets.

After that we noticed that the most of 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 for the first dataset and 0.5 for the second dataset. If the produced classification output does not reach this probability, then the agent requests information from other agents. Thereafter, the proposed system is compared with

- Normal DDM algorithm without any collaboration between agents,
- Traditional DDM low-level,
- Traditional DDM high-level.

A. MAS-MCC-NB VS NON-COMMUNICATING AGENT RESULTS

In this section the MAS-MCC-NB compared with normal DDM algorithm for the classification algorithm that each agent used without any collaboration and communication between them.

1) DATASET- 1 RESULTS

The results for the first dataset contacted that the MAS-MCC-NB is overpowered and more accurate than the non-communicating system as illustrated in Tables 3 and shown in Figures 7 – 11 for each agent.

TABLE 3. MAS-MCC-NB vs without collaboration results.

	Agent-1	
Test #	MAS-MCC-NB	Without Collaboration
1	75.67%	68.22%
2	72.50%	68.13%
3	74.25%	65.83%
4	75.33%	70.90%
5	71.80%	67.00%
	Agent-2	
Test #	MAS-MCC-NB	Without Collaboration
1	74.67%	72.00%
2	75.50%	74.00%
3	73.50%	73.00%
4	76.78%	75.67%
5	72.10%	71.30%
	Agent-3	
Test #	MAS-MCC-NB	Without Collaboration
1	72.67%	69.56%
2	70. 88%	70.25%
3	74.88%	67.00%
4	75.22%	71.33%
5	72.50%	69.00%
	Agent-4	
Test #	MAS-MCC-NB	Without Collaboration
1	71.33%	71.00%
2	69.88%	69.33%
3	73.75%	71.65%
4	76.12%	73.91%
5	71.80%	67.80%
	Agent-5	
Test #	MAS-MCC-NB	Without Collaboration
1	70.33%	64.00%
2	73.83%	71.00%
3	75.01%	74.29%
4	77.33%	76.22%
5	73.60%	73.40%

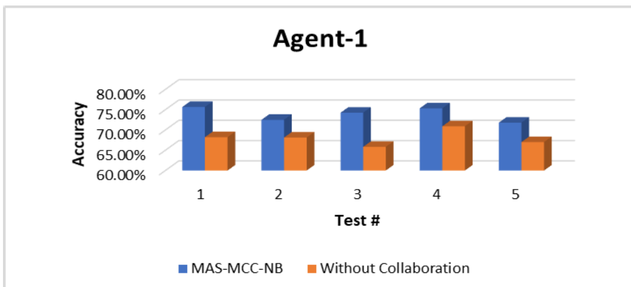

FIGURE 7. Agent 1 results (Dataset-1).

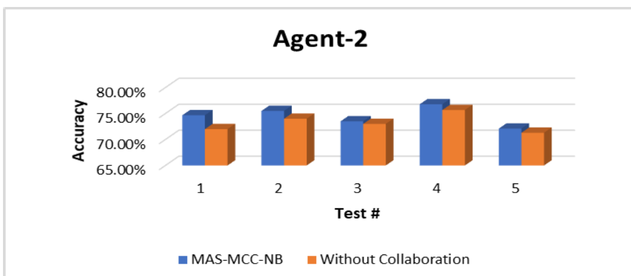

FIGURE 8. Agent 2 results (Dataset-1).

At the end of the experimental results for the first dataset show that MAS-MCC-NB is excellent in enabling the decision whether to accept the received class label from other agents. This finding shows the ability of the utilized MAS-MCC-NB in enhancing the results of the contacted agents,

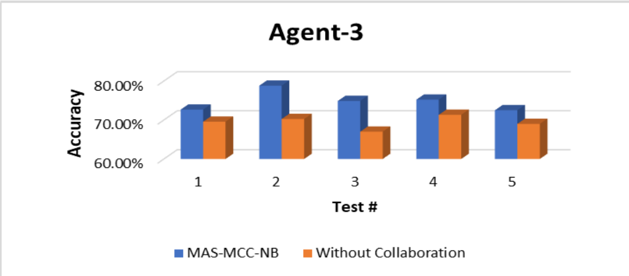

FIGURE 9. Agent 3 results (Dataset-1).

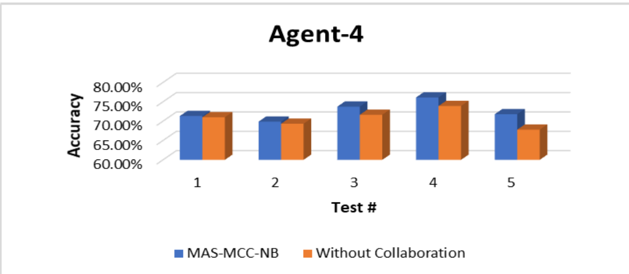

FIGURE 10. Agent 4 results (Dataset-1).

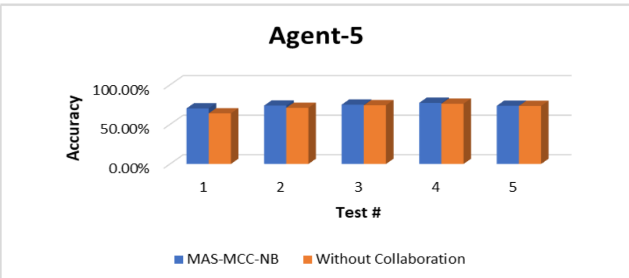

FIGURE 11. Agent 5 results (Dataset-1).

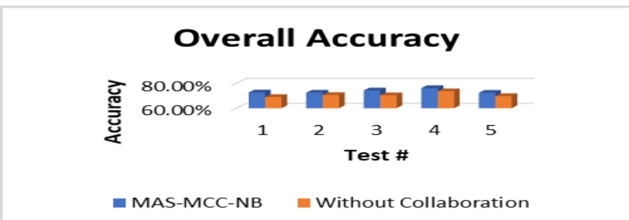

FIGURE 12. Overall accuracy (Dataset-1).

TABLE 4. Overall accuracy (Dataset-1).

Test #	MAS-MCC-NB	Without Collaboration
1	72.60%	68.96%
2	72.47%	70.54%
3	74.28%	70.35%
4	76.16%	73.61%
5	72.36%	69.70%

with the rate of 1.93 % to 3.64 %. The results for the first dataset are illustrated in Tables 4 and shown in Figures 12.

2) DATASET- 2 RESULTS

The results for the first dataset contacted that the MAS-MCC-NB is overpowered and more accurate than the non- communicating system as illustrated in Tables 5 and shown in Figures 13 – 15 for each agent.

TABLE 5. MAS-MCC-NB vs without collaboration results.

Agent-1		
Test #	MAS-MCC-NB	Without Collaboration
1	74.00%	62.00%
2	74.26%	64.00%
3	84.16%	65.48%
Agent-2		
Test #	MAS-MCC- NB	Without Collaboration
1	82.00%	60.00%
2	75.25%	65.13%
3	86.14%	65.50%
Agent-3		
Test #	MAS-MCC- NB	Without Collaboration
1	76.00%	72.66%
2	77.25%	76.00%
3	82.18%	73.48%

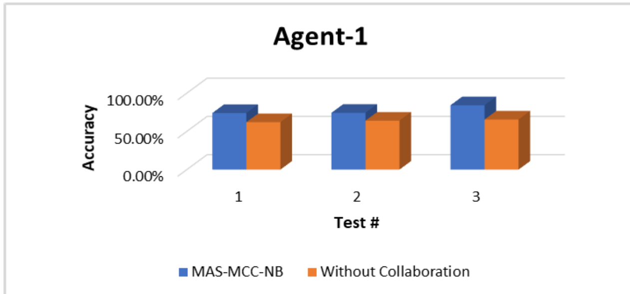

FIGURE 13. Agent 1 results (Dataset-2).

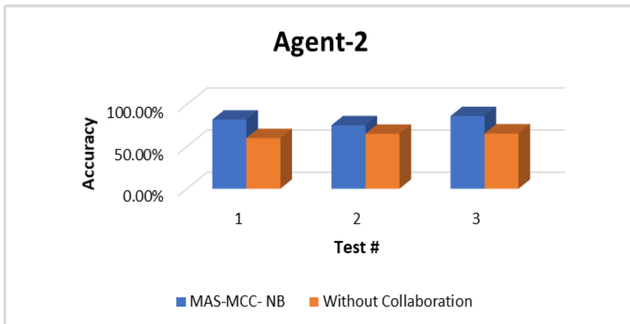

FIGURE 14. Agent 2 results (Dataset-2).

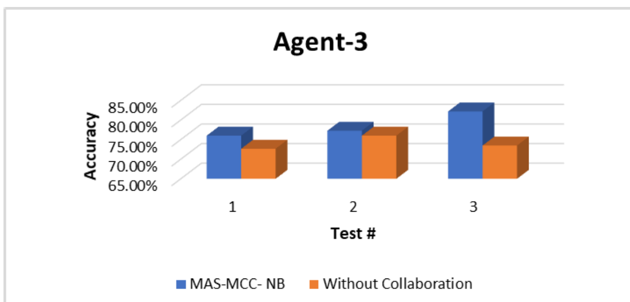

FIGURE 15. Agent 3 results (Dataset-2).

At the end of the experimental results for the second dataset show that MAS-MCC-NB is excellent in enabling the decision whether to accept the received class label from other agents. This finding proves the ability of the utilized

TABLE 6. Overall accuracy (Dataset-2).

Test #	MAS-MCC-NB	Without Collaboration
1	77.33%	64.89%
2	75.59%	68.38%
3	84.16%	68.15%

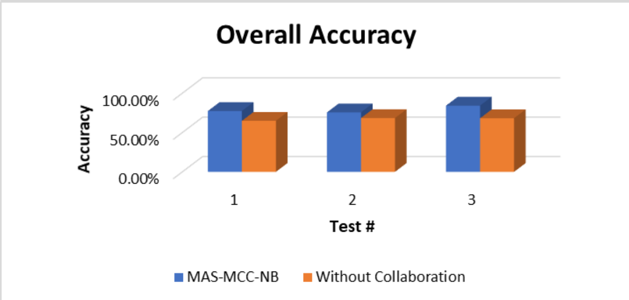

FIGURE 16. Overall accuracy (Dataset-2).

TABLE 7. Model accuracy for each agent and global model.

Dataset-1		
Agent #	Model Accuracy	Global Model
Agent 1	75.1%	74.14%
Agent 2	78.06%	
Agent 3	79.39%	
Agent 4	75.17%	
Agent 5	81.89%	
Dataset-2		
Agent #	Model Accuracy	Global Model
Agent 1	90.27%	90.30%
Agent 2	86.91%	
Agent 3	94.79%	

MAS-MCC-NB in enhancing the results of the contacted agents, with the rate of 7.21 % to 16.01 %. The results for the first dataset are illustrated in Tables 6 and shown in Figures 16.

B. MAS-MCC-NB VS HIGH-LEVEL, AND LOW-LEVEL DDM METHOD RESULTS

In this section, the MAS-MCC-NB compared with DDM Low-level, and DDM high-level. Each agent of the high-level DDM shares its learned model with the global model to produce a single learning model for mining the input data, and low-level DDM is integration of voting results of multiple independent decision making. Tables 7 the model accuracy for each agent, and the global model for first and second datasets.

1) DATASET- 1 RESULTS

The results for the first dataset contacted that the MAS-MCC-NB is overpowered and more accurate than the high-level DDM, and low-level DDM as illustrated in Tables 18 and shown in Figures 17 – 21 for each agent.

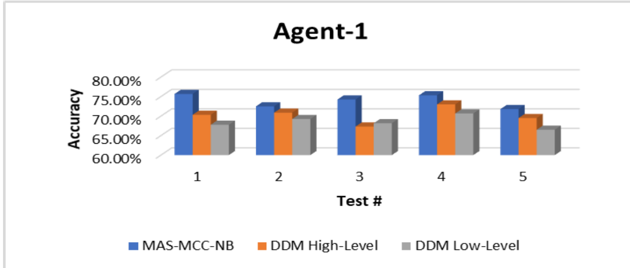

FIGURE 17. Agent 1 results (Dataset-1).

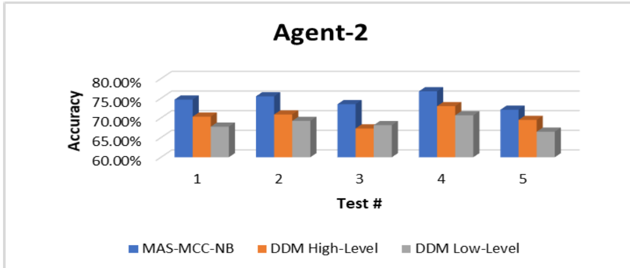

FIGURE 18. Agent 2 results (Dataset-1).

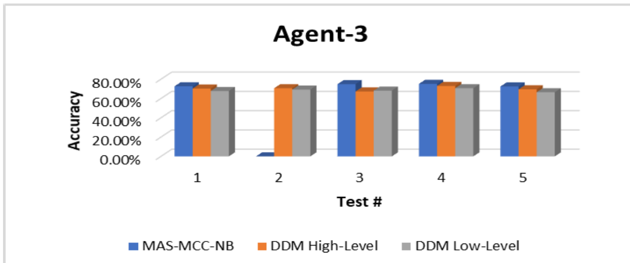

FIGURE 19. Agent 3 results (Dataset-1).

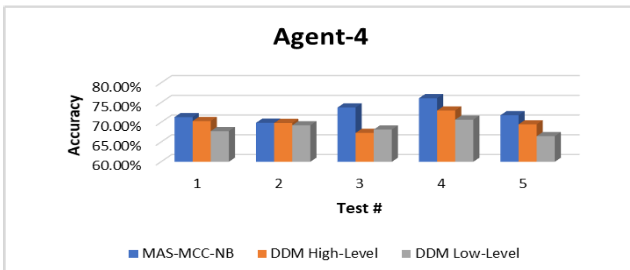

FIGURE 20. Agent 4 results (Dataset-1).

This finding proves the ability of the utilized MAS-MCC-NB in enhancing the results of the contacted agents. It obtains a rate of 2.60% - 6.95% in forming the high-level DDM for NB and a rate of 3.27% – 6.11% in forming the low-level DDM for NB. The results for the first dataset are illustrated in Tables 9 and shown in Figures 22.

2) DATASET- 2 RESULTS

The results for the second dataset contacted that the MAS-MCC-NB is overpowered and more accurate than

TABLE 8. MAS-MCC-NB vs high-level DDM and low-level DDM results.

Test #	MAS-MCC-NB	DDM High-Level	DDM Low-Level
Agent-1			
1	75.67%	70.33%	67.78%
2	72.50%	70.88%	69.25%
3	74.25%	67.33%	68.17%
4	75.33%	73.00%	70.67%
5	71.80%	69.50%	66.50%
Agent-2			
1	74.67%	70.33%	67.78%
2	75.50%	70.88%	69.25%
3	73.50%	67.33%	68.17%
4	76.78%	73.00%	70.67%
5	72.10%	69.50%	66.50%
Agent-3			
1	72.67%	70.33%	67.78%
2	70. 88%	70.67%	69.25%
3	74.88%	67.33%	68.17%
4	75.22%	73.00%	70.67%
5	72.50%	69.50%	66.50%
Agent-4			
1	71.33%	70.33%	67.78%
2	69.88%	69.83%	69.25%
3	73.75%	67.33%	68.17%
4	76.12%	73.00%	70.67%
5	71.80%	69.50%	66.50%
Agent-5			
1	70.33%	70.33%	67.78%
2	73.83%	69.83%	69.25%
3	75.01%	67.33%	68.17%
4	77.33%	73.00%	70.67%
5	73.60%	69.50%	66.50%

TABLE 9. Overall accuracy (Dataset-1).

Test #	MAS-MCC-NB	DDM High-Level	DDM Low-Level
1	72.93%	70.33%	67.78%
2	72.52%	69.83%	69.25%
3	74.28%	67.33%	68.17%
4	76.16%	73.00%	70.67%
5	72.36%	69.50%	66.50%

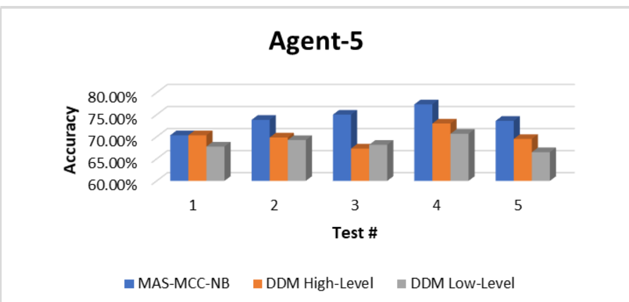

FIGURE 21. Agent 5 results (Dataset-1).

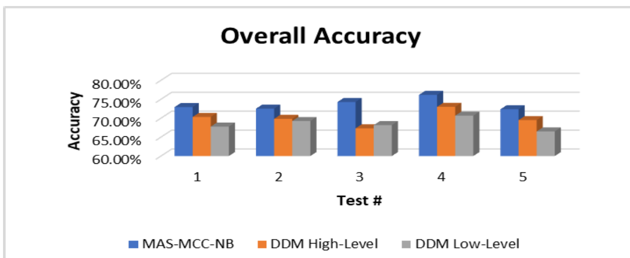

FIGURE 22. Overall accuracy (Dataset-1).

the high-level DDM, and low-level DDM as illustrated in Tables 10 and shown in Figures 23 – 25 for each agent.

This finding proves the ability of the utilized MAS-MCC-NB in enhancing the results of the contacted agents. It obtains a rate of 11.45% - 21.34% in forming the high-level DDM for NB and a rate of 10.39% – 12.82% in forming the low-level

TABLE 10. MAS-MCC-NB vs high-level DDM and low-level DDM results (Dataset-2).

Agent-1			
Test #	MAS-MCC-NB	DDM High-Level	DDM-Low-Level
1	74.00%	57.00%	65.52%
2	74.26%	67.11%	68.17%
3	84.16%	67.75%	67.26%
Agent-2			
Test #	MAS-MCC-NB	DDM High-Level	DDM-Low-Level
1	82.00%	57.00%	65.52%
2	75.25%	67.11%	68.17%
3	86.14%	67.75%	67.26%
Agent-3			
Test #	MAS-MCC-NB	DDM High-Level	DDM-Low-Level
1	76.00%	57.00%	65.52%
2	77.25%	67.11%	68.17%
3	82.18%	67.75%	67.26%

TABLE 11. Overall accuracy (Dataset-2).

Test #	MAS-MCC-NB	DDM High-Level	DDM-Low-Level
1	78.34%	57.00%	65.52%
2	78.56%	67.11%	68.17%
3	79.54%	67.75%	67.26%

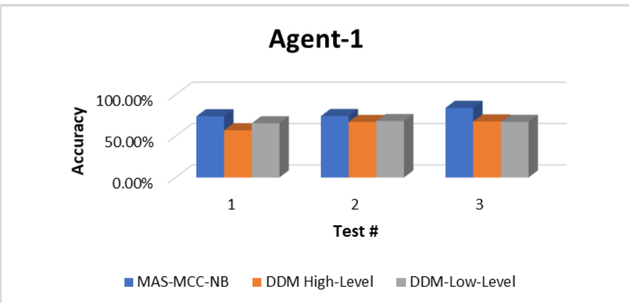

FIGURE 23. Agent 1 results (Dataset-2).

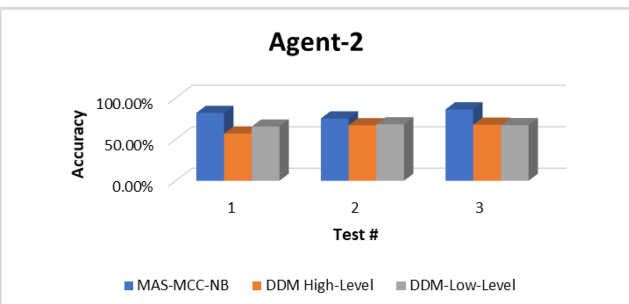

FIGURE 24. Agent 2 results (Dataset-2).

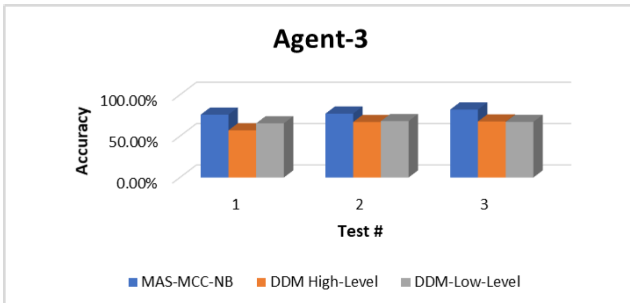

FIGURE 25. Agent 3 results (Dataset-2).

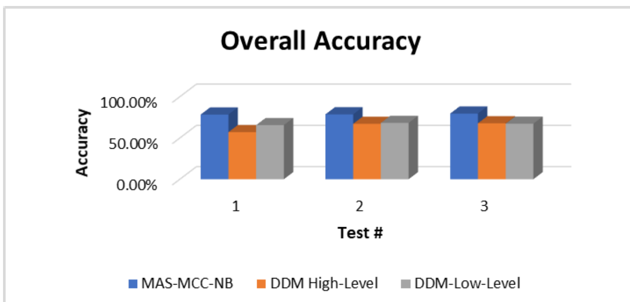

FIGURE 26. Overall accuracy (Dataset-2).

DDM for NB. The results for the first dataset are illustrated in Tables 11 and shown in Figures 26.

The experimental results of the MAS-MCC-NB show that each agent makes efficient and accurate decision on the new cases they receive than high-level DDM, and low-level DDM. This result proved the ability of the utilized MAS-MCC-NB in enhancing the results of the involved agents about 6.95% for the first dataset and 21.34% for the second dataset. In addition to accuracy, the MAS-MCC-NB promotes information diversity and agent particularity, preserves data coverage.

V. CONCLUSION

Mutual collaboration agent classification using naive Bayesian classification technique is developed and implemented within a MAS. In the proposed technique, each agent stores its samples and establishes its prior assumption on the basis of these data. In classifying new data item, each agent uses naive Bayesian classifier with reference to its own prior assumption and shared information from other agents when required. Accordingly, agents involved in the collaboration classification tasks promote information diversity and particularity based on their own data when sharing information among one another to enhance the output. The MAS-MCC-NB is identical to the typical one. However, in the sharing of this classifier, the probabilities used to update and calculate classifier labels are different from those of typical one.

The experimental result for the two datasets shows that the proposed MAS-MCC-NB is always more accurate and with enhanced results compared with the non – collaboration system. Similarly, the experimental results show that the MAS-MCC-NB successfully enabled the initiator agent to decide whether to accept the received class label from the other agents. MAS-MCC-NB achieved much better results than the non-collaboration system. Also, the experimental results indicate that the model is highly accurate and efficient for the cases of the agent when the model is built on the basis of the characteristics of the dataset. Compiling a global model from different local models make the model highly generalized. However, this approach is inefficient for a few local models specialized for certain cases.

For Future work we will testing our system MAS-DDM for different classifier and for different application.

APPENDIX

The example, presented below, involves three agents that have to decide whether or not to play golf which depends of whether attribute, where each one of them have different attribute shows the specific producers for the MAS-DDM-NB framework:

Each agent has the following dataset:

The MAP can be calculated by first constructing a frequency table for each attribute against the target. The frequency tables to likelihood tables are then transformed and NB equation is used to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction.

Initiator Agent

Outlook	Temp	Humidity	Play Golf
Rainy	Hot	High	No
Rainy	Hot	High	No
Overcast	Hot	High	Yes
Sunny	Mild	High	Yes
Sunny	Cool	Normal	Yes

Agent-1

Outlook	Temp	Humidity	Play Golf
Sunny	Hot	High	No
Rainy	Hot	High	No
Overcast	Hot	High	No
Overcast	Hot	High	No
Overcast	Mild	High	Yes
Sunny	Cool	Normal	Yes

Agent-2

Outlook	Temp	Humidity	Play Golf
Sunny	Hot	High	No
Overcast	Hot	High	Yes
Rainy	Mild	High	Yes
Rainy	Cool	Normal	Yes

Each agent constructs its model by calculating the probability of each class, the probability of each attribute value, and the conditional probability of each attribute value with each class.

In the first scenario, the set of input attribute values provided to the requested agent will be as follows:

Outlook	Temp	Humidity
Rainy	**Hot**	**High**

In calculating class probability and request collaboration, the probabilities of all class labels known to the local agent are calculated when a set of attribute values is provided to that agent, as given in Equation 1.

$$\hat{y} = p(\text{Rainy}/Yes) \times p(Hot/Yes) \times p(\text{High}/Yes) \times p\,(Yes)$$

InitiatorAgent

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	2	0
	Overcast	1	0
	Rainy	0	2
Temp	Hot	1	2
	Mild	1	0
	Cool	1	0
Humidity	High	2	2
	Normal	1	0

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	2/3	0/2	2/5
	Overcast	1/3	0/2	1/5
	Rainy	0/3	2/2	2/5
		3/5	2/5	
Temp	Hot	1/3	2/2	3/5
	Mild	1/3	0/2	1/5
	Cool	1/3	0/2	1/5
		3/5	2/5	
Humidity	High	2/3	2/2	4/5
	Normal	1/3	0/2	1/5
		3/5	2/5	

Agent-1

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	0	1
	Overcast	1	0
	Rainy	0	2
Temp	Hot	1	1
	Mild	1	0
	Cool	1	0
Humidity	High	2	1
	Normal	1	0

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	1/3	0/1	1/4
	Overcast	1/3	0/1	1/4
	Rainy	0/3	2/1	2/4
		2/4	2/4	
Temp	Hot	1/3	1/1	2/4
	Mild	1/3	0/1	1/4
	Cool	1/3	0/1	1/4
		3/4	1/4	
Humidity	High	2/3	1/1	3/4
	Normal	1/3	0/1	1/4
		3/4	2/4	

Agent-2

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	1	1
	Overcast	1	2
	Rainy	0	1
Temp	Hot	0	4
	Mild	1	0
	Cool	1	0
Humidity	High	1	4
	Normal	1	0

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	1/2	1/4	2/6
	Overcast	1/2	2/4	3/6
	Rainy	0/2	1/4	1/6
		2/6	4/6	
Temp	Hot	0/2	4/4	4/6
	Mild	1/2	0/4	1/6
	Cool	1/2	0/4	1/6
		2/6	4/4	
Humidity	High	1/2	4/4	5/6
	Normal	1/2	0/4	1/6
		2/6	4/6	

$$\hat{y} = p(\text{Rainy}/No) \times p(Hot/No) \times p(\text{High}/No) \times p(No)$$
$$= 0/3 \times 1/3 \times 2/3 \times 3/5 = 0$$
$$= 2/2 \times 2/2 \times 2/2 \times 2/5 = 0.4$$

The local agent decided that the class of the input attribute values will be "***no***".

The requested agent then compares their class probability with the allowed thresholds, which in this example is "0.3", where $0.4 > 0.3$ causes the requested agent to take their predicted value and not request help from another agent.

In the second scenario, the set of input attribute values provided to the requested agent will be as follows:

Outlook	Temp	Humidity
Overcast	Hot	High

$$\hat{y} = p(\text{Overcast}/Yes) \times p(Hot/Yes) \times p(\text{High}/Yes) \times p\,(Yes)$$
$$= 1/3 \times 1/3 \times 2/3 \times 3/5 = 0.0437 = 0/2 \times 2/2 \times 2/2 \times 2/5 = 0$$
$$\hat{y} = p(\text{Overcast}/No) \times p(Hot/No) \times p(\text{High}/No) \times p\,(No)$$
$$= 0/2 \times 2/2 \times 2/2 \times 2/5 = 0$$

The local agent decided that the class of the input attribute values will be "***yes***".

The requested agent compares their class probability with the threshold, which in this example is "0.3", where $0.0437 < 0.3$ causing the requested agent to request for help from another agent. The other agents send the following probabilities to the requested agent:

- Class labels C_k above the threshold;
- Joint probability of the set of attributes conditionally dependent on each class that is Above the threshold $\prod_{i=1}^{n} P(X_i|C_k)$; and
- Joint probabilities of the attribute set $\prod_{i=1}^{n} P(X_i)$.

$$\hat{y} = p(\text{Overcast}/Yes) \times p(Hot/Yes) \times p(\text{High}/Yes) \times p\,(Yes)$$
$$\hat{y} = p(\text{Overcast}/No) \times p(Hot/No) \times p(\text{High}/No) \times p\,(No)$$
$$= 0/1 \times 1/1 \times 1/1 \times 1/4 = 0$$

Therefore, the agent decided that the class of the input attribute values will be "***yes***", but it will not replay since $0.0574 < 0.3$.

$$\hat{y} = p(\text{Overcast}/Yes) \times p(Hot/Yes) \times p(\text{High}/Yes) \times p\,(Yes)$$
$$= 1/2 \times 0/2 \times 1/2 \times 2/6 = 0$$
$$\hat{y} = p(\text{Overcast}/No) \times p(Hot/No) \times p(\text{High}/No) \times p\,(No)$$
$$= 2/4 \times 4/4 \times 4/4 \times 4/6 = 0.34$$

The local agent decided that the class of the input attribute values will be "***no***", and the requested agent will be sent the following information:

$$\textit{\textbf{Class label}} = no$$
$$\prod_{i=1}^{n} P(X_i|C_k) = 2/6 * 4/6 * 4/6 = 0.15$$
$$\prod_{i=1}^{n} P(X_i) = 3/6 * 4/6 * 5/6 = 0.28$$

In feedback processing, the initiator agent compares its most remarkable joint attribute probability, which is conditionally dependent on its corresponding probability received by other agents, with the received joint attribute probability with its corresponding probability. Thus, the result will be as follows:

$$\begin{aligned} \textbf{Initiator1} &= \prod_{i=1}^{n} P(X_i|C_k) \times \prod_{i=1}^{n} P(X_i) \\ &= 0.016 * 0.096 = 0.0015 \\ \textit{\textbf{Agent2}} &= \prod_{i=1}^{n} P(X_i|C_k) \times \prod_{i=1}^{n} P(X_i) \\ &= 0.15 * 0.28 = 0.042 \end{aligned}$$

where agent 2 is most desirable combined probability, which indicates that agent has the most remarkable knowledge on the attributes and the given class is considered and where requested agent will take the class '***no***''.

REFERENCES

[1] A. A. Ali, P. Varacha, S. Krayem, P. Zacek, and A. Urbanek, ''Distributed data mining systems: Techniques, approaches and algorithms,'' in *Proc. MATEC Web Conf.*, vol. 210, 2018, Art. no. 04038.

[2] Q. Tong, X. Li, and B. Yuan, ''Efficient distributed clustering using boundary information,'' *Neurocomputing*, vol. 275, pp. 2355–2366, Jan. 2018.

[3] S. K. Tanbeer, C. F. Ahmed, and B. S. Jeong, ''Parallel and distributed frequent pattern mining in large databases,'' in *Proc. 11th IEEE Int. Conf. High Perform. Comput. Commun.*, Jun. 2009, pp. 407–414.

[4] A. Amir, B. Srinivasan, and A. I. Khan, ''Distributed classification for image spam detection,'' *Multimedia Tools Appl.*, vol. 77, no. 11, pp. 13249–13278, Jun. 2018.

[5] S. Bandyopadhyay, C. Giannella, U. Maulik, H. Kargupta, K. Liu, and S. Datta, ''Clustering distributed data streams in peer-to-peer environments,'' *Inf. Sci.*, vol. 176, no. 14, pp. 1952–1985, Jul. 2006.

[6] N. Obeid and A. Moubaiddin, ''Towards a formal model of knowledge sharing in complex systems,'' in *Smart Information and Knowledge Management*. Berlin, Germany: Springer, 2010, pp. 53–82.

[7] N. Obeid and A. Moubaiddin, ''On the role of dialogue and argumentation in collaborative problem solving,'' in *Proc. 9th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2009, pp. 1202–1208.

[8] C. Y. Chen and J. J. Huang, ''Double deep autoencoder for heterogeneous distributed clustering,'' *Information*, vol. 10, no. 4, p. 144, Apr. 2019.

[9] W. Gan, J. C. W. Lin, H. C. Chao, and J. Zhan, ''Data mining in distributed environment: A survey,'' *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 7, no. 6, p. e1216, 2017.

[10] V. Sawant and K. Shah, ''A review of distributed data mining using agents,'' *Int. J. Adv. Technol. Eng. Res.*, vol. 3, no. 5, pp. 27–33, 2013.

[11] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, ''Naive Bayes classification of uncertain data,'' in *Proc. 9th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 944–949.

[12] E. Januzaj, H. P. Kriegel, and M. Pfeifle, ''DBDC: Density based distributed clustering,'' in *Proc. Int. Conf. Extending Database Technol.* Berlin, Germany: Springer, 2004, pp. 88–105.

[13] M. Klusch, S. Lodi, and G. Moro, ''Distributed clustering based on sampling local density estimates,'' in *Proc. IJCAI*, vol. 3, 2003, pp. 485–490.

[14] N. Chawla, S. Eschrich, and L. O. Hall, ''Creating ensembles of classifiers,'' in *Proc. IEEE Int. Conf. Data Mining*, San Jose, CA, USA, Nov./Dec. 2001, pp. 580–581.

[15] M. A. Almaiah and A. Al-Khasawneh, ''Investigating the main determinants of mobile cloud computing adoption in university campus,'' *Educ. Inf. Technol.*, vol. 25, no. 4, pp. 3087–3107, Jul. 2020.

[16] 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, Jun. 2018.

[17] 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,'' in *Proc. KDD*, Newport Beach, CA, USA, vol. 97, Aug. 1997, pp. 74–81.

[18] M. Adil, R. Khan, M. A. Almaiah, M. Binsawad, J. Ali, A. Al Saaidah, and Q. T. Ta, ''An efficient load balancing scheme of energy gauge nodes to maximize the lifespan of constraint oriented networks,'' *IEEE Access*, vol. 8, pp. 148510–148527, Aug. 2020.

[19] K. A. Albashiri, F. Coenen, and P. Leng, ''EMADS: An extendible multi-agent data miner,'' in *Proc. Int. Conf. Innov. Techn. Appl. Artif. Intell.* Cambridge, U.K.: Springer, 2008, pp. 263–275.

[20] J. Tozicka, M. Rovatsos, and M. Pechoucek, ''A framework for agent-based distributed machine learning and data mining,'' in *Proc. 6th Int. Joint Conf. Autonomous Agents Multiagent Syst.*, Honolulu, HI, USA, May 2007, pp. 1–8.

[21] L. Cao, G. Weiss, and P. S. Yu, ''A brief introduction to agent mining,'' *Auto. Agents Multi-Agent Syst.*, vol. 25, no. 3, pp. 419–424, 2012.

[22] M. Adil, M. A. Almaiah, A. A. Omar, and O. Almomani, ''An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks,'' *Sensors*, vol. 20, no. 8, p. 2311, Jan. 2020.

[23] A. N. Moghadam and R. Ravanmehr, ''Multi-agent distributed data mining approach for classifying meteorology data: Case study on Iran's synoptic weather stations,'' *Int. J. Environ. Sci. Technol.*, vol. 15, no. 1, pp. 149–158, Jan. 2018.

[24] S. Chaimontree, K. Atkinson, and F. Coenen, ''A multi-agent based approach to clustering: Harnessing the power of agents,'' in *Proc. Int. Workshop Agents Data Mining Interact.* Taipei, Taiwan: Springer, May 2011, pp. 16–29.

[25] M. Adil, R. Khan, M. A. Almaiah, M. Al-Zahrani, M. Zakarya, M. S. Amjad, and R. Ahmed, ''MAC-AODV based mutual authentication scheme for constraint oriented networks,'' *IEEE Access*, vol. 8, pp. 44459–44469, Mar. 2020.

[26] W. Liu and J. Wang, ''A brief survey on nature-inspired metaheuristics for feature selection in classification in this decade,'' in *Proc. IEEE 16th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2019, pp. 424–429.

[27] H. Liu, X. Zhang, and X. Zhang, ''PwAdaBoost: Possible world based AdaBoost algorithm for classifying uncertain data,'' *Knowl.-Based Syst.*, vol. 186, Dec. 2019, Art. no. 104930.

[28] A. Prati and M. Sbracia, ''Uncertainty and currency crises: Evidence from survey data,'' *J. Monetary Econ.*, vol. 57, no. 6, pp. 668–681, Sep. 2010.

[29] B. Qin, Y. Xia, and F. Li, ''A Bayesian classifier for uncertain data,'' in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2010, pp. 1010–1014.

[30] B. Qin, Y. Xia, S. Wang, and X. Du, ''A novel Bayesian classification for uncertain data,'' *Knowl.-Based Syst.*, vol. 24, no. 8, pp. 1151–1158, Dec. 2011.

[31] Y. Sun, A. K. Wong, and M. S. Kamel, ''Classification of imbalanced data: A review,'' *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 4, pp. 687–719, 2009.

[32] D. M. J. Tax, M. van Breukelen, R. P. W. Duin, and J. Kittler, ''Combining multiple classifiers by averaging or by multiplying?'' *Pattern Recognit.*, vol. 33, no. 9, pp. 1475–1485, Sep. 2000.

[33] M. N. Khan, H. U. Rahman, M. A. Almaiah, M. Z. Khan, A. Khan, M. Raza, M. Al-Zahrani, O. Almomani, and R. Khan, ''Improving energy efficiency with content-based adaptive and dynamic scheduling in wireless sensor networks,'' *IEEE Access*, vol. 8, pp. 176495–176520, Sep. 2020.

[34] S. Merugu and J. Ghosh, ''Privacy-preserving distributed clustering using generative models,'' in *Proc. 3rd IEEE Int. Conf. Data Mining*, Nov. 2003, pp. 211–218.

[35] N. Friedman and M. Goldszmidt, *Learning Bayesian Networks From Data*. San Mateo, CA, USA: Morgan Kaufmann, 1999.

[36] M. H. Qasem, A. Hudaib, and N. Obeid, ''Multiagent system for mutual collaboration classification for cancer detection,'' *Math. Problems Eng.*, vol. 2019, pp. 1–17, Dec. 2019.

[37] M. Adil, R. Khan, J. Ali, B. H. Roh, Q. T. Ta, and M. A. Almaiah, ''An energy proficient load balancing routing scheme for wireless sensor networks to maximize their lifespan in an operational environment,'' *IEEE Access*, vol. 8, pp. 163209–163224, Aug. 2020.

[38] J. Vanschoren. (2014). *BNG (Breast-Cancer, Nominal, 1000000)*. [Online]. Available: https://www.openml.org/d/77

[39] J. Teng. (2019). *SEER Breast Cancer Data*. [Online]. Available: https://ieee-dataport.org/open-access/seer-breast-cancer-data