A Novel Association Rule-Based Data Mining Approach for Internet of Things Based Wireless Sensor Networks

AHMED M. KHEDR1,4, WALID OSAMY2, AHMED SALIM3,4, AND SOHAIL ABBAS1

1Department of Computer Science, University of Sharjah, Sharjah 27272, UAE
2Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
3Department of Computer Science, College of Science and Arts, Qassim University, Buraidah 51931, Saudi Arabia
4Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44516, Egypt

Corresponding author: Walid Osamy (walid.osamy@fci.bu.edu.eg)

ABSTRACT Wireless Sensor Network (WSN) is one of the fundamental technologies used in the Internet of Things (IoT) which is deployed for diverse applications to carry out precise real-time observations. The limited resources of WSN with massive volume of fast-flowing IoT data make the aggregation and analytics of data more challenging. Recently, data mining-based solutions have been proposed to effectively handle the data being generated by the sensors and to analyze the data patterns for deducing the required information from it. The increasing need of these techniques motivated us to propose a distributed and efficient data mining technique that not only handles the massive and rapidly generated data by the nodes, but also increases the life span of the network. In this paper, we propose a novel scheme for the IoT based WSN that mines the sensor data using association rule without moving it to any Cluster Head (CH) or Base Station (BS). The new proposed scheme enables sensors to perform computations locally and only the minimum higher-level statistical summaries of the data at Cluster Members (CMs) are exchanged with their CH. This considerably reduces the communication overhead which ultimately prolongs the network lifetime. The proposed scheme is evaluated via extensive simulations and the results obtained demonstrate that the integration of the proposed scheme in the existing protocols significantly reduces the communication overhead which ultimately prolongs the network lifetime and stability.

INDEX TERMS Association rules, distributed databases, Internet of Things, wireless sensor networks based-clustering, network lifetime, stability, energy consumption.

I. INTRODUCTION

With the advent of advanced communication technologies and sophisticated protocols, new paradigms have emerged on the technological horizon. The most prevalent one is the Internet of Things (IoT) which has attained extensive popularity and acceptance due to its broad range of applications in our daily life. One of the main functions and integral parts of the IoT is environment sensing using sensors that are usually deployed as standalone or embedded in other objects like smartphones, cars, building, etc. For effective decision making, data is gathered continuously or continually in a large scale from a domain of interest such as battle fields, natural disasters, health monitoring, coal mines, agriculture, weather, etc.. However, for large scale scenarios, sensor nodes are interconnected wirelessly to form a self-organized network, called Wireless Sensor Network (WSN), which is considered as the main source of huge volume of data generation. According to the International Data Corporation (IDC) report, IoT devices would reach a threshold of 6 billion whereas, data being communicated would raise to 175 Zettabytes in the span of 2018-2025, giving a 30 percent boost to real-time data by 2025 [1].

One of the key data generation sources for the IoT is WSNs in which sensors gather huge volumes of data across diverse applications such as remote health monitoring, weather, surveillance, etc. [2]–[4]. These applications are generally critical and require real-time analysis for effective decision making and reliable network operation(s). By integrating the WSN with the IoT and cloud (as depicted in Fig. 1),
an almost unlimited storage and computation facility can be provided to such real-time applications for efficient data analytics using Artificial Intelligence and Machine Learning based algorithms. However, the intra-WSN data traffic should be minimized; otherwise, the availability of the IoT and the cloud may not be properly utilized due to the limited lifetime of the WSN, as we will discuss later in this article. The gateway node is responsible to bridge the WSN and the IoT.

To manage the data efficiently in the WSNs, various authors advocated the database-oriented methods to be useful for efficient data handling while treating WSNs as distributed database. In such a context, sensor nodes being data sources act as database relations stored at each sensor, constituting a network level distributed database [5], [6]. The distributed database management in WSNs provides two-fold advantage: first, the cost of data collection and analysis in terms of energy is minimized; second, an SQL-like abstractions can be implemented on WSNs that would simplify data collection and query processing [7], [8].

Recently, data being relations of a database, data mining-based approaches have been used in the IoT domain in order to efficiently and effectively analyze the huge volume and high-speed data generated by the WSNs for the intelligent decision making. However, mining the data from the resource constraint sensors is considered to be a challenging task; hence, new AI based techniques are emerging to devise effective analytics upon huge quantity of data generated by WSNs for the IoT infrastructure. Some basic examples of these techniques used for data analytics and producing intelligent applications in the IoT domain are classification, clustering, association analysis, time series analysis, and outlier analysis. Among these techniques, association analysis is concerned with determining interesting patterns from a huge set of data items and to find all the co-occurrence relationships from a dataset, called associations. Association rules mining techniques for WSNs can be classified based on data processing location: Centralized and Distributed. In a centralized method, data from the entire network is stored in a central site for further analysis. In this case, the initial data reduction is performed in the central site. On the other hand, the in-network method considers the limited resource of sensor nodes and performs some extra computation in the nodes to limit the message and communication energy during transferring the data to the central site. Association rule mining is used in various sub-systems of the IoT domain, such as the smart homes, water quality monitoring [9], event detection in traffic management systems, health monitoring, and intrusion detection [10], [11]. However, although effective, a large body of these schemes are not appropriate to be used in the WSNs based IoT due to the fact that they are usually resource-hungry and require a centralized architecture which do not suit the distributed and resource constraint nature of the WSN. It is therefore required to redesign and customize the data mining based approaches to be adapted to the WSN architecture for the overall performance improvement of the IoT. For instance, for a WSN, a viable technique would be the one which is distributed and capable of tackling continuous and rapid streams of data without incurring a lot of processing and communication overhead. Moving along this direction, one of the main factors is energy conservation that is always required in the battery powered IoT-incorporated sensors.

Since, the main cause of energy depletion has been proved to be data exchanges among sensors [12], various data reduction techniques have been proposed to conserve energy [13], [14], such as data compression [15]–[21], packet merging [22], data fusion [23], and approximation-based [24].

Clustering in WSNs serves as a major strategy for energy preserving and extending network lifetime [19], [25], [26]. The clustering algorithm in WSN divided into three phases. The role of phase 1 is to initiate clustering processes and in phase 3, data are collected from cluster members (CMs) and then forward to a BS or sink. While in phase 2, the most important step is done, i.e., Cluster Head (CH) selection then the clusters are formed. CH selection is the essential step in clustering processes and there are various algorithms in the literature that handle this step, for example, DEC [27] algorithm utilizes energy as the factor for CH selection, and LEACH-MF [3] utilizes energy, moving speed, and pause time as the factors for CH selection, while, CREEP [28] and ECH [4] utilize energy and distance as the factors for CH selection. In our work, we considered that the network is clustered using any clustering algorithms for WSNs.

In this paper, WSN is divided into clusters each having a CH leading their CMs see Fig. 1. In the proposed association rule algorithm, CHs have the potential to decompose computations into local ones at their CMs. CHs and CMs exchange minimal statistical summaries in order to perform a partial computation task. Eventually, these partial outcomes from each CH will be aggregated by the BS. The BS may ultimately transfer this data to the cloud for further processing and analysis, via the IoT platform. Moreover, our approach is not only distributed in nature that suits the WSN architecture, but also it considerably reduces the data exchanges among sensors and between sensors and BS thereby saving the
network bandwidth and energy, hence prolonging the lifespan of the network.

The contribution of this paper is as follows: We propose a novel version of the association rule-based data mining approach for the IoT based WSNs platform that finds the association rules of the sensor data without moving the data to BS or CHs. The new proposed scheme considerably reduces the energy depletion of the sensor nodes which is the main constraint. The effectiveness of the proposed scheme is proved via total exchanged messages analysis and via integration with the benchmark algorithms with extensive simulations using various metrics and compared the results with the original benchmarks. The results obtained depict significant performance improvement in terms of energy and network lifetime.

The article is organized into various sections as follows. Section II, presents the related work, Section III presents the integration methodology of the databases, Section IV presents the proposed technique, Section V presents the experimentation and results, and finally, Section VI concludes the paper.

II. RELATED RESEARCH

In the context of IoT, WSNs are the main data generation source offering new prospects for data mining and data analytics research to extract useful information for diverse array of applications. In the literature, a wide range of data mining algorithms have been applied in the IoT domain for discovering different knowledge patterns. These algorithms have been exploited in a variety of ways [29]. Frequent mining, sequential mining, classification, and clustering are the prominent approaches developed to analyze data and to make effective decisions in the IoT systems [30].

Various methods related to our proposed work have been proposed in the literature, which are reviewed as follows. The authors in [31] proposed a centralized technique named as Data Stream Association Rule Mining (DSARM). The DSARM is used to identify the missing values in the data captured by sensors. This technique detects the sensor nodes that recursively transmit the duplicate data and it also estimates the missing values by manipulating the readings reported by other related sensors. Another estimation technique proposed by [32], [33], termed as Closed item-sets-based Association Rule Mining (CARM), is employed to deduce the recent sensor association rules in the sliding window based on the latest closed item-sets. In [34], the authors proposed an online one-pass technique that changes the WSN data-stream to a list form, called Interval List (IL), thereby employing the inter-stream association rule mining from the huge sensor data stream. In [35], the authors proposed a rule-learning based technique that extracts distinct rules from the data reported by sensors to control and coordinate the operations performed by the network. In [36], the authors proposed a tree like data structure called the sensor pattern tree, that is used to derive association rules from sensor data. This technique is advantageous because it scans the database only once.

In the literature, various distributed approaches have also been proposed to solve the application related issues and to optimize the performance of WSN. The authors in [37] address the problem of distributed clustering in the context of WSN. An asynchronous distributed clustering algorithm is proposed which is based on in-network learning. The sensors learn clusters from the data being sensed without communicating the raw data to the BS; thereby minimizing mining time and communication overhead. The k-means and Gaussian Mixture Models were used as main clustering algorithms. This approach requires each node to know other network nodes and communicate data summaries. In [38], the authors proposed an environmental monitoring system and investigated the effect of environmental parameters on the Gross Primary Productivity (GPP) level by using and evaluating six classification models, i.e., naïve Bayes, support vector machine, multilevel perceptron, decision tree, and k-nearest-neighbor. After selecting the best classifier, it is deployed on sensors for predicting the effect of sensed data on the GPP level. For communication overhead reduction, sensors communicate the outcomes to the BS only if the GPP is affected by the sensor readings.

In [39] authors analyzed large datasets derived from WSN based real-time air pollution monitoring system. Different decision-making strategies were devised after applying business intelligence and data mining techniques on the data. K-means was used for clustering. The approach proposed in [40] integrates the WSN with the Artificial Neural Network (ANN) to detect the forest fire. Data related to fire, such as smoke, light, and temperature, is collected by sensors deployed at different regions and is transmitted to a pre-trained ANN installed at the BS. The ANN then detects fire and generate alarms. In [41], the authors proposed a forest fire detection and monitoring framework for clustered WSN. They also presented the inter and intra-cluster protocols. To reduce communication overhead, the fire detection task is done only by cluster heads.

In [42], data mining technique is used in the clustered WSN to detect fire in the forest. Each individual node is responsible for fire detection using data mining-based classifier. Upon detection, a sensor node transmits the alarm message to the BS routed via different CHs and CMs. To reduce the communication overhead, each sensor node only sends the abnormal values to the BS using the mining technique. All the duplicate and normal readings are discarded. Data mining approaches have also been applied for the advancement of agriculture using WSN integrated with IoT. For example, [43]–[45] employed association rules and linear regression to mine the sensor readings for efficient and accurate decision making.

In [46], the authors proposed an approach for energy reduction in the WSNs, called EK-means. The scheme works in two steps. In the first step, similar data is eliminated at each sensor using Euclidean distance. In the second step, the communication overhead is reduced by applying an enhanced version of k-means clustering at data aggregator nodes to
TABLE 1. Summary of data mining techniques used in WSN.

| Approach                      | Objectives                        | Node Data       | Limitations                                      | Evaluation         | Architecture         | Data Movement |
|-------------------------------|-----------------------------------|-----------------|--------------------------------------------------|--------------------|-----------------------|---------------|
| DSARM [32]                    | Estimation of missing data        | Homogeneous     | Ignore sensor that report different values       | Simulation         | Centralized           | Yes           |
| CARM [33], [34]               | Estimation of missing data        | Heterogeneous   | Cannot handle high speed data                    | Simulation         | Centralized           | Yes           |
| Online one-pass algorithm [35]| Represent WSNs data as interval list | Homogeneous     | Duplication in data                              | Analytical model   | Centralized           | Yes           |
| Lightweight rule learning [36]| Discover highly correlated rules  | Heterogeneous   | No validation provided on real data              | Simulation         | Centralized           | Yes           |
| SP tree [37]                  | Discover event patterns          | Homogeneous     | Overhead in terms of tree construction           | Analytical model   | Centralized           | Yes           |
| Distributed clustering algorithm [38] | Clustering                    | Heterogeneous   | Complete map to be provided for nodes            | Simulation         | Distributed           | No            |
| Environmental monitoring [39] | Environmental events detection    | Heterogeneous   | Classifier selection not specified               | Simulation         | Distributed           | No            |
| Fire detector [42]            | Pure detection in forest          | Heterogeneous   | Delayed detection if cluster head is far from fire | Simulation         | Distributed           | Yes           |
| Naive Bayes [43]              | Detection of events              | Heterogeneous   | Limited results                                  | Simulation         | Centralized           | No            |
| EK-means [47]                 | Reduce data transmission         | Heterogeneous   | Not validated for real scenarios                 | Simulation         | Centralized           | Yes           |
| DKC [48]                      | Reduce data redundancy           | Homogeneous     | Not validated for real scenario                  | Prototype           | Distributed           | Yes           |
| K-means and one-way ANOVA model [49] | Eliminate redundancy         | Homogeneous     | Not validated for real scenarios                 | Simulation         | Distributed           | Yes           |
| ASP-tree and SWASP-tree [50]  | Patterns extraction              | Homogeneous     | Complex and requires huge processing             | Simulation         | Centralized           | Yes           |

group duplicate datasets produced by the neighbors into one cluster before sending the data to the BS. In [47], the authors proposed a Distributed k-means Clustering (KDC) mechanism. The authors also proposed an efficient data aggregation technique for WSN using adaptive weighted allocation which is based on their proposed KDC. The aim of the KDC is to reduce the data duplication at the sensor nodes. A closely related approach is used by [48] in underwater WSN. In [49], the authors proposed a data mining algorithm and a compact tree structure named as Associated Sensor Pattern tree (ASP-tree) for WSN. Both of these techniques are used to capture the associated sensor patterns. For overhead reduction, all associated patterns are produced by scanning the whole dataset only once using a pattern growth-based technique.

The above mentioned schemes are summarized in Table 1. The existing techniques produce considerable communication overhead to achieve improved real-time decision making with enhanced precision for the WSN applications. This usually become one of the main causes of energy depletion in the WSN.

Our proposed technique is distributed and different from the ones mentioned above. The main idea of our proposed technique is that we consider the data at sensor nodes in WSN.
to be a distributed database \((D)\). The data stored at each sensor node is stored in the form of rows and columns (i.e., columns denote sensor attributes). In our proposed method, the sensor nodes are grouped into different clusters, each having a CH that manages the cluster and a set of CMs. The CMs will periodically answer the queries generated by their CHs; however, only the statistical summaries would be communicated back to CHs. The CHs after aggregating the summaries will send the accumulated and computed outcome to the BS for final processing. These queries (SQL-like abstractions) are often continuous so that the application is notified continuously regarding the changes recorded by the sensors, unlike traditional queries that mainly focus on the current state of a database [5]. Sensor nodes would not respond to queries until they have new recorded readings in contrast to the previous probing. This approach of data accumulation reduces the huge volume of data communicated across the network, reducing the computation burden incurred on the BS. This would ultimately increase the lifespan of the WSN. Furthermore, for efficient query processing that incurs low energy dissipation and minimal delay, an efficient load balancing policy is employed that takes into account the remaining power and the load of the nodes.

The time synchronization problem to synchronize the local clocks of sensor nodes in the WSN have been extensively studied in literature over the last two decades and yet there is no specific time synchronization scheme available to achieve higher order of accuracy with greater scalability independent of topology and application [50]–[52]. In this paper, a synchronization process is assumed between the sensor nodes, however the synchronization process is not the main study of this paper.

**III. PROBLEM FORMULATION**

In this section, we provide the description of different types of data distribution and the proposed methodology to manage these distributed databases without moving and join at one node such as CH or BS. As mentioned earlier, we assume that each sensor node possesses a component database containing a set of attributes.

**A. IMPLICIT GLOBAL DATABASE**

Each sensor node \(s_i\) stores a component \(D_i\) of the global database \(D\). Data at \(s_i\) in the form of tuples. A database component \(D_k\) residing at node \(s_k\), includes certain attributes shared with \(D_i\) \((k \neq i)\) and some diverse non-shared attributes. The distribution strategy of databases require "Join" operation to construct the implicit global database \(D\) from the components \(D_i\)'s. The proposed methodology utilizes the shared attributes to perform the processing of data. This scheme presents a more realistic approach than applying non-overlapping single key attributes set for the components that are allocated around the nodes in the network. The implicit global database \(D\) exists as fragments (each fragment represents a data at one sensor node) that are distributed over the nodes in the network. By implicit format we mean that each tuple of \(D\) exists in a distributed format at sensor nodes, i.e., tuples do not explicitly exist at BS or end user.

**B. INTEGRATION OF SENSORS DATABASES**

We assume that the global database \(D\) corresponding to the WSN is distributed as local database components over all the sensor nodes in the WSN. The distribution as described above in implicit global database. The global database \(D\) can be generated at the end user or BS through the join of such component relations and can provide remarkable data suitable for performing computation as well as mining activities using association rules. The key focus of the proposed scheme is to mine the implicit global database \(D\) using association rules by maintaining every data fragment \(D_i\) of sensor \(s_i\) and minimizing the data communication between the CMs and their CH. As a result, the local results of each database \(D_i\) at sensor node \(s_i\) will only be transmitted from the CM \(s_i\) to its respective CH for aggregation. Finally, the results from CHs will be sent to the BS for producing final association rules results.

The mathematical formulation of the proposed problem can be described as follows: consider a WSN with \(n\) sensor nodes, each sensor node \(s_i\) has a component of database \(D_i\)'s with \(A_i\) as the set of attributes. Let \(A\) is the union of attributes of the local components at all sensor nodes in the WSN (Equation 1).

\[
A = \bigcup_{i=1}^{n} A_i
\]  

(1)

Let \(S_{ij}\) is the set of attributes shared between \(D_i\) and \(D_j\) as follows:

\[
S_{ij} = \bigcap_{q=i,j} A_q
\]  

(2)

Then, the union of all shared attributes among all the local relation can be defined as the set \(S\) where \(S\) can be computed as follows:

\[
S = \bigcup_{i,j} S_{ij}
\]  

(3)

The proposed scheme emphasizes on determining the association rules of implicit \(D\) through minimal communication of messages among the WSN nodes. Therefore, the global computation task is divided into local computations taking into account the shared attribute constraints. As a result, the aggregation of summaries of local computation results can help to produce the global association rules. This can be formulated mathematically in general way as follows: Consider that a function \(F\) is applied on the explicit database \(D\) to obtain the result \(R\) as given in Equation 4. As stated previously, the required distributed computation is the derivation of association rules corresponding to \(D\). Here, we can denote \(F\) as the algorithmic implementation for the derivation of association rules for \(D\), where \(R\) represents the obtained association rules for \(D\).

\[
R = F(D).
\]  

(4)
If the database $D$ is implicitly defined, the responsibility of every CM to perform local computation on its respective database component. The local results are then exchanged through communication with CH to obtain the global results of computation. The attribute set $S$ shared between components can finally result in generating global $D$ from the components $D_1$ to $D_n$. The corresponding realization of function $F$ as given in Equation 4 can be rewritten as:

$$\text{R}(S) = H[h_1(D_1, S), h_2(D_2, S), \ldots, h_n(D_n, S)]$$  \hspace{1cm} (5)

Here, $h_i(D_i, S)$ represents the $i^{th}$ CM’s local computation implemented on $D_i$ at $s_i$ such that, the operation $H$ represents the aggregation of the results of local computations executed by the CHs. Each problem involves a distinct set of $h$-operators ($h_i$'s) and the features of $H$ and $h_i$ relies on $S$ and the participated $D_i$'s. Finally, the BS will get the discovered associations rules by the CHs.

**FIGURE 2.** Computations on implicit data at $CH_i$ using the explicit data at cluster members.

Fig. 2 shows the process by which the $CH_i$ would compute $R$ from the $D_i$'s. That is, a local computation $h_i(D_i)$ is performed at every sensor $s_j$ using the database $D_j$. The results of these local computations are aggregated using the operation $H$. The component operators of a decomposition ($H$ and $h_i$'s), therefore, need to be dynamically determined by the CH for each instance of F(D), depending on the participating nodes, the attributes contained in their native databases, and the sharing pattern of attributes.

**IV. DISTRIBUTED ASSOCIATION RULES MINING FOR IoT BASED WSN**

The key focus of our newly proposed scheme is to identify the global association rules of the implicit global database $D$. This global computation task is divided and allocated over the sensor nodes in the network such that the computations are executed locally at each CM and only the statistical summaries are gathered and communicated. The BS initiates the global computation task by sending requests to the CHs to perform computations and find the association rules with its CMs. On receiving such requests, each CH starts to create the shared relation with its CMs and asks its CMs to execute computations such as support and confidence locally as we will explain in detail with example in this section. This helps to minimize the size of messages and ensure that only minimal number of messages are communicated between CMs and CH and between BS and CHs, which in turn can minimize the energy utilized and hence enhance the WSN lifetime.

**FIGURE 3.** Flow Diagram of the Proposed Scheme.

The proposed distributed mining technique for deriving association rules from a cluster at IoT based WSN comprises of three main phases: Initialization, Support and Confidence Computing, and Aggregation as depicted in Fig. 3. During the initialization phase, each CH generates the shared relation using attributes of shared set and their values obtained from the CMs. In Support and Confidence Computing phase, each CH queries its members to compute the Support and Confidence. Finally, in the aggregation stage, CH determines the local association rules and sends the aggregated results to the BS.

The whole network is converted into $k$ distinct clusters with the help of a clustering technique, such as DEC [27]. Each CH $CH_i$ has $n$ associated CMs $s_j$, $j = 1 \ldots n$. The CH and its CMs collaboratively implement the mining algorithm as presented in Algorithm 1.

**A. INITIALIZATION PHASE**

In this phase, every CH generates the shared relation as follows: We define the relation $P_{\text{shared}}$ as the cross product
of all distinct values of shared attributes in S, i.e., it includes the records corresponding to all possible combinations of values for attributes in S, mediating the formation of the global D. The records at each node which have zero count are then eliminated from the relation Pshared and hence the resultant relation will be the shared relation, as described in Algorithm 2. This phase will be executed by every CHi.

**Algorithm 2 Shared Relation (will Be Executed by Every CHi With Its CMs)**

1. Find shared Attributes among CMs and the distinct values of each shared attribute.
2. Pshared = Cross Product of all different values of shared attributes
3. Create the relation Shared by removing from Pshared any tuple with zero matching at any CM.
4. Index the Shared relation.

1) Generate relation Pshared as the cross product of the distinct values of the shared attributes of S.
2) Create the relation Shared from the relation Pshared by removing from Pshared all tuples with zero count.

**Example:** For clarification, we consider three database components at the sensor nodes s1, s2 and s3 with three databases D1, D2 and D3 respectively such that these databases jointly determine the global implicit database D. Moreover, we assume that s1, s2 and s3 are CMs of a cluster with CH1. The local databases D1, D2 and D3 from the three nodes are shown in Table 2.

**TABLE 2. Explicit Database Components D1, D2, and D3 at sensor nodes s1, s2 and s3.**

|   | D1 at s1 | D2 at s2 | D3 at s3 |
|---|----------|----------|----------|
| a | b1       | b1       | a2c2     |
| a1| b1       | c1f      | d3       |
| a3| b2       | b2c1     | a2c1     |
| a2| b1c1     | b2c1f1   | d1c1     |
| a5| b2c2     | b2c2     | a2c2     |
| a4| b2c1f1   | b2c2     | a2c2d4   |
| a6| b1       | b1       | a2c1     |
| a1| b2c2     | b2c1f1   | a2c2d4   |

The shared attributes are a, b and c with distinct values for a = {a1,a2}, b = {b1, b2, b3} and c = {c1,c2}. The Pshared relation will be cross products of the values of shared attributes as in Table 3:

**TABLE 3. Pshared Relation.**

| a  | b  | c  |
|----|----|----|
| a1 | b1 | c1 |
| a1 | b1 | c2 |
| a1 | b2 | c1 |
| a1 | b2 | c2 |
| a1 | b3 | c1 |
| a1 | b3 | c2 |
| a1 | b1 | c1 |
| a1 | b1 | c2 |
| a2 | b1 | c1 |
| a2 | b2 | c1 |
| a2 | b2 | c2 |
| a2 | b3 | c1 |
| a2 | b3 | c2 |

Considering only the tuples in Pshared that have non zero count at s1, s2 and s3, the record a1, b1, c2 has zero value at s3 and the relation Shared will be as in Table 4.

**TABLE 4. Indexed Shared relation.**

| Index | a  | b  | c  |
|-------|----|----|----|
| 0     | a1 | b1 | c1 |
| 1     | a2 | b1 | c1 |
| 2     | a2 | b1 | c2 |
| 3     | a2 | b2 | c1 |

**B. SUPPORT AND CONFIDENCE COMPUTING PHASE**

This phase is implemented by each CHi to perform the following two tasks:

1) To enumerate the candidate-sets of the upcoming level using the frequent item-sets of the prior level.
2) To compute support as well as confidence values.

Every CHi will initiate the execution of the Support and Confidence Computing phase. CHi is responsible for performing the crucial control operations such as determining and handling both the active and candidate item-sets, communicating with its CMs to determine the support and confidence. The computation operation is therefore decomposed and iteratively executed and managed by every CHi.

The support of an item-set can be defined as the ratio between the count of transactions which include the item-set and the total count of transactions in the implicit D. Hence, the major computational primitive required is the determination of total tuples count in D.

1) TUPLES COUNT IN IMPLICIT DATABASE

The Tuples Count in Implicit Database can be computed only after obtaining the local outcomes of computation from each sensor node. However, such computations that satisfy particular attribute-value conditions (shared tuple) are challenging and are detailed below. We decompose this process of finding tuples, requesting feedback from the CMs of Di’s regarding the local counts. The corresponding replies from the CMs are then used to determine Ntotal(D), i.e., the total tuples in D. This can be expressed as follows:

\[ N_{total}(D) = \sum_{j} \prod_{i}(N_{Di})_{cond_j} \]  

(6)
where, cond$_j$ denotes the attribute-value condition for tuple $j$th belonging to $Shared$ relation, N(D$_i$)$_{cond_j}$ denotes the tuples count in $D_i$ at CM $s_i$ that satisfies cond$_j$. Based on Equation 6, we can write:

$$h_i(D_i, S) = N(D_i)_{cond_j}$$ (7)

Such that, $j$ refers to the $j$th tuple of relation $Shared$. It is required to have such summary for each tuple in $Shared$ from each CM. The role of the function $H$ is to calculate the sum-of-products from the deduced summaries according to the Equation 6, in which each product term represents the count of tuples satisfying cond$_j$ in a $D_i$ and the resultant gives the number of distinct tuples fulfilling cond$_j$, needed for the implicit Join of all the $D_i$’s. Then the summation operation is performed on the product terms computed for each tuple. This operation simulates a Join operation executed on all the databases without explicit enumeration of the tuples. The most favorable aspect of decomposing $N_{total}(D)$ is that it is possible to translate each product term N(D$_i$)$_{cond_j}$ into an SQL query; select count (*), such that cond$_j$ can be executed using Equation 6 by taking CM ($s_i$).

In the example mentioned above, the count of each shared tuple can be computed using Equation 6 by taking cond$_2$ as the items of each tuple in shared for example the first tuple of shared relation, cond$_2$ will be (a = a1, b = b1, c = c1) which implies N(D$_1$)$_{cond_2}$ = 2, N(D$_2$)$_{cond_2}$ = 2, and N(D$_3$)$_{cond_2}$ = 2 and the sum product will be 8, i.e., the number of tuples corresponding to first shared tuple will be 8. The indexed relation shared and the number of tuples corresponding to each shared are computed in Table 5. The total tuples ($N_{total}$) will be 23.

### Table 5. Indexed $Shared$ relation with the number of tuples corresponding to each shared.

| Index | a | b | c | $SOD_1$ | $SOD_2$ | $SOD_3$ | $N(D_o)_{cond_2}$ |
|-------|---|---|---|---------|---------|---------|--------------------|
| 0     | a1| b1| c1| 2       | 2       | 2       | 8                  |
| 2     | a2| b1| c2| 1       | 2       | 2       | 4                  |
| 3     | a2| b2| c1| 1       | 2       | 2       | 4                  |
| 4     | a2| b2| c2| 1       | 2       | 2       | 4                  |

2) SUPPORT AND CONFIDENCE FOR CANDIDATE SETS

The support of an item-set can be defined as the ratio between the number of transactions which include the item-set and the total number of transactions in $D$, and the confidence regarding a set of transactions is the fraction of transactions which includes the consequent $Y$ given that it includes the antecedent $X$. Hence, the major computational primitive required is the determination of total tuples count in $D$ which can be computed only after obtaining the local outcomes of computation from each sensor node. It is attainable to extend the tuples count decomposition to have the number of tuples which meet a new condition by modifying cond$_j$ of Equation 6 as provided below, which is essential to identify the support measure for a candidate frequent item-set.

$$N_{new-condition} = \sum_{j=1}^{n} N(D_i)_{cond_j \text{ and } new-condition}$$ (8)

The method in which CH$_l$ finds the support measure for a candidate frequent item-set is as described below. In relation $Shared$, CH$_l$ checks the condition specified and identify the tuples matching the attribute-value pairs in the candidate set and then retains those tuples to find the number of tuples resulted from this reduced $Shared$ relation. The support level for a candidate set of attribute-value pairs is given by the ratio of the resultant candidate set count by the total count $N_{total}$.

### Algorithm 3 Frequent Item-Sets Computing

1: $k=1$
2: while $C_k$ is not empty do
3: for every $I_i \in C_k$ do
4: $I_i.count = 0$
5: Construct new $- cond$ on
6: for every database at CM, $D_i$ do
7: $d=0$, $j=1$
8: for every shared tuple $\in$ Indexed $Shared$ relation do
9: $j = j * N(D_i)_{cond_j \text{ and } new-condition}$
10: end for
11: $d=d+j$
12: end for
13: $I_i.count = d$
14: end for
15: $F_k \leftarrow\{I_i \in C_k | I_i.count \geq \text{minsup}\}, k=k+1$
16: $C_k = \text{generateCandidate}(F_{k-1})$
17: end while

Algorithms 3 and 4 provide the frequent item-sets computation at each CH and the candidate item-set generation procedures respectively.

In previous Example, in order to extract frequent item-sets, we assume the following: (1) the minimal threshold value (support value) is 0.30, i.e., the minimum count of occurrence for an item-set in the result list of frequent item-sets is 7 (0.30 * 23), (2) the minimum size of item-set is 1 and (3) the maximum size of an item set is set to the largest item-sets we have found (i.e., size $k$ = 1, 3, etc.).

Using Equation 8 the frequent item-sets at the CH$_1$ cluster head will be as follows:

- In the initial iteration of the algorithm, we consider each item as a member of candidate 1-item-sets ($C_1$), i.e., $C_1 = \{a_1, a_2, b_1, b_2, c_1, c_2, e_1, e_2, e_4, d_3, d_4, d_8, d_9, d_{10}, d_{11}, f_1, j_2, f_6, f_{8}, f_9\}$. The frequency of every item can be computed at CH$_1$ using Equation 8, i.e., find the number of tuples that have each of the attributes $a=a_1, a=a_2, b=b_1, b=b_2, c=c_1, c=c_2, e=e_1, e=e_2, e=e_4, d=d_3$, etc.
Algorithm 4 Generate Candidate (will be Executed by Each \( C_i \))

1: Input: \( F_{k-1} \) [frequent item-sets]
2: Output: \( C_k \) [the set of candidates]
3: \( C_k \leftarrow \phi \)
4: for \( i=1:|F_{k-1}| \) do
5: if \( k < 2 \) then
6: for \( j=i+1:|F_{k-1}| \) do
7: \( C_k \leftarrow F_{k-1}(i) \cup F_{k-1}(j) \)
8: end for
9: else
10: for \( j=i+1:|F_{k-1}| \) do
11: if \( F_{k-1}(i) \cap F_{k-1}(j) \neq \phi \) then
12: for \( r=1:|C_k| \) do
13: if \( (F_{k-1}(i) \cup F_{k-1}(j)) == C_k(r) \) then
14: break
15: end if
16: end for
17: end if
18: end for
19: end if
20: end for
21: end if
22: end for

For example, the frequency of \( a=a_1 \) can be computed as follows: in Equation 8, \( new-condition = \{a=a_1\} \) and \( cond_j=\) shared tuple with index \( j = 0, 1, 2, 3, 4 \). 
\[
N_{(a=a_1)} = 2 \times 2 \times 2 + 0 \times 2 \times 3 + 0 \times 1 \times 2 + 0 \times 1 \times 3 + 0 \times 2 \times 2 = 8,
\]
the frequency of \( b_1 = e_2 \) can be computed as follows: in Equation 8, \( new-condition = \{e_1=e_2\} \) and \( cond_j=\) shared tuple with index \( j = 0, 1, 2, 3, 4 \). 
\[
N_{(e_1=e_2)} = 1 \times 2 \times 2 + 0 \times 2 \times 0 \times 0 \times 1 \times 1 \times 3 + 0 \times 0 \times 2 = 11.
\]
Similarly, the number of tuples containing the other non-shared items are given in Table 6

| Item | Count | Item | Count |
|------|-------|------|-------|
| e1   | 4     | d8   | 3     |
| e2   | 4     | d9   | 3     |
| e4   | 8     | d10  | 4     |
| d3   | 3     | d11  | 5     |
| d4   | 4     | f1   | 2     |
| d2   | 7     | b1   | 2     |
| 18   | 5     | 19   | 3     |

TABLE 6. Non-shared counts.

- According to our threshold (support value), the set of frequent 1-item sets of \( C_1 \) will be \( F_1 = \{a_1, a_2, b_1, b_2, c_1, e_2, e_4, f_1, f_2, f_6\} \). The set of candidate 2-item sets of \( C_2 \) will be the set of 2-combinations of items in \( F_1 \) such that there is no more than one item belongs to the same column, e.g., \( \{a_1, a_2\} \) is invalid combination. As a result the combinations will be as follows:

\[
\{a_2, b_1\}, \{c_1, e_1\}, \{e_2, f_1\}, \{b_2, c_1\}, \{b_2, e_1\}, \{f_1, f_2\}, \{a_1, b_1\}, \{c_1, e_2\}, \{e_1, f_1\}, \{a_2, c_1\}, \{a_1, c_1\}, \{e_4, f_2\}, \{a_2, e_1\}, \{b_1, e_1\}, \{b_2, f_1\}, \{c_1, f_2\}, \{a_1, e_1\}, \{a_2, e_2\}, \{b_1, f_1\}, \{b_2, f_2\}, \{a_1, e_2\}, \{a_2, f_1\}, \{c_1, e_4\}, \{b_1, e_4\}, \{a_2, e_4\}, \{a_1, e_4\}.
\]

As in previous step we can compute the frequency for each combination in Table using Equation 8. for example using Equation 8, the frequency of \( \{e_1, f_1\} \) will be as follows: \( new-condition = \{e=e_1 \times e=f_1\} \) and \( cond_j=\) shared tuple with index \( j = 0, 1, 2, 3, 4 \). And so \( N_{(e_1=e_1 \times e=f_1)} = 1 \times 1 \times 9 \times 1 \times 3 + 0 \times 0 \times 0 \times 2 = 2 \). Similarly, the number of tuples containing the other non-shared items are given in Table 7

| Item  | Count | Item  | Count |
|-------|-------|-------|-------|
| \{a_2, b_1\} | 8 | \{b_1, c_1\} | 4 |
| \{b_2, c_1\} | 7 | \{b_2, e_1\} | 0 |
| \{a_1, b_1\} | 8 | \{b_1, e_2\} | 7 |
| \{a_2, b_2\} | 7 | \{b_1, c_1\} | 4 |
| \{b_1, c_1\} | 14 | \{b_2, e_2\} | 7 |
| \{a_1, b_2\} | 0 | \{c_1, f_1\} | 7 |
| \{a_2, c_1\} | 9 | \{a_2, e_2\} | 0 |
| \{a_1, e_1\} | 8 | \{b_1, e_4\} | 4 |
| \{a_2, e_1\} | 2 | \{a_1, e_2\} | 4 |
| \{e_4, f_2\} | 3 | \{b_1, f_1\} | 7 |
| \{a_1, c_1\} | 8 | \{c_1, e_4\} | 4 |
| \{a_2, c_1\} | 8 | \{a_1, f_1\} | 3 |
| \{a_2, f_1\} | 4 | \{a_1, f_2\} | 3 |

TABLE 7. \( F_2 \) Item sets.

- In order to form the set of candidate 3-item-sets, \( C_3 \), we find the combination from items in \( F_2 \) such that there is no more than one item that belongs to the same column, e.g., \( \{a_1, a_2, a_1\} \) is an invalid combination. Also, any two 2-itemsets from \( F_2 \) that intersect with others, e.g., \( \{a_2, b_1\} \) and \( \{c_1, e_2\} \) cannot form new combination of 3-item-set. As a result, the candidate of 3-itemsets combinations are as listed as follows:

\[
\{a_2, b_1, c_1\}, \{a_2, b_1, e_2\}, \{a_2, b_2, c_1\}, \{a_2, b_2, e_2\}, \{a_2, c_1, f_1\}, \{a_2, c_1, f_2\}, \{a_1, c_1, f_1\}, \{a_1, e_2, b_2\}, \{c_1, e_2, a_2\}, \{b_2, e_2, a_2\}, \{a_2, b_1, e_4\}, \{a_1, b_1, e_4\}, \{a_2, c_1, e_4\}, \{a_1, b_1, c_1\}, \{a_2, c_1, e_2\}, \{a_1, c_1, e_2\}, \{c_1, e_2, f_2\}, \{b_2, f_1, e_4\}, \{a_2, b_2, e_4\}.
\]

- Similarly, we can get the frequency of each item in \( C_3 \). The set of frequent 3-item-sets, \( F_3 \), consisting of those
candidate 3-item-sets in $C_3$ that have minimum support are as shown in Table 8.

**TABLE 8. Frequent 3-item-sets.**

| Item     | Count |
|----------|-------|
| $a_1b_1c_1$ | 8     |
| $a_2b_2c_2$ | 7     |
| $a_2c_1e_2$ | 7     |
| $a_2b_1e_4$ | 7     |

- The combination from $F_3$ is $C_4 = a_2, b_2, c_1, e_2$ and the count of this item-set is 3 and as a result $F_3 = \phi$ which is below support value. Thus, $C_5 = \phi$, and the algorithm execution is terminated and all of the frequent item-sets have been obtained.

**C. EXTRACTION AND INTEGRATION OF RULES**

**Extraction:** During this step, each $CH_i$ extracts the association rules using frequent item-sets $F$. The major steps involved in rules extraction process as presented in Algorithm 5 are given below:

**Algorithm 5** Extract Rules (will Be Executed by Every $CH_i$)

1. Input: $F_i$: large item-sets, support, confidence
2. Output: $R^i$: Association Rules satisfying support and confidence at $CH_i$.
3. $R = \phi$.
4. for each $f \in F_i$ do
5. for each $c \subset f | c \neq \phi, c \neq f$ do
6. if $\frac{\text{support}(c)}{\text{support}(f)} \geq \text{confidence}$ then
7. $R^i = R^i \cup \{c \Rightarrow (f - c)\}$
8. end if
9. end for
10. end for

- for each frequent item-set $f \in F$, using all nonempty subsets $c$ of $f$ and $c \neq f$.
- for every subset $c$, if

$$\frac{\text{support}(f)}{\text{support}(c)} \geq \text{confidence}$$

(where confidence is a threshold).

- $R_i = R_i \cup \{c \Rightarrow (f - c)\}$.

Then, a message is sent to the BS from every $CH_i$ which includes the rules set $R_i$ with confidence of each rule and the total tuples count ($N_i$).

**Integration:** BS integrates the rules by considering the confidence with the help of $N_i$ and the total tuples count of the global WSN database. The major steps involved in the integration process as presented in the Algorithm 6 are given below:

- Input $R_i = \{r_i^j, c_i^j\}, i = 1, \ldots, k, j$ is the rules count in $R_i$, $c_i^j$ is the confidence and $r_i^j$ is the rule.
- $N_i$ is the tuples count obtained from $CH_i$.
- Let $\delta$ be the total tuples count obtained from $k$ clusters.

**Algorithm 6** Rules Integration (will Be Executed by Base Station)

1. $N_{total-j}$ is number of tuples at cluster $j$.
2. Let $\text{confidence}(R^j_i)$ is the confidence of rule $R_i$ at cluster $j$.
3. Let $\delta$ is the total number of tuples obtained from $k$ clusters.
4. $R = R_1 \cup \cdots \cup R_k \{ \text{all rules that received from } k \text{ clusters} \}$
5. $\delta = \sum_{i=1}^{k} \text{num_of_tuples}_i$
6. for each rule $R_i \in R$ do
7. $W_i = \sum_{j} \text{confidence}(R^j_i) \times \frac{\text{num_of_tuples}_j}{\delta}$
8. end for
9. Select rules that satisfy the weight threshold value

- $R = R_1 \cup \cdots \cup R_k$.
- for every $r^n_i \in R$
  - for $i = 1$ to $k$
    - if $r^n_i$ in $R_i$ possess confidence $c_i^j$
      - $c_i^m = c_i^m + \frac{N_i}{\delta} \times c_i^j$

In our example, finally at $CH_1$, the rules generated as shown in Table 9.

**TABLE 9. Association Rules.**

| Rules       |       |
|-------------|-------|
| $a_1 \rightarrow b_1 c_1$ | support: 8, confidence: 1 |
| $a_1 b_1 \rightarrow c_1$ | support: 8, confidence: 1 |
| $a_1 c_1 \rightarrow b_1$ | support: 8, confidence: 1 |
| $b_1 [1] \rightarrow c_1$ | support: 7, confidence: 1 |
| $c_1 [1] \rightarrow b_1$ | support: 7, confidence: 1 |
| $a_2 b_2 \rightarrow c_1$ | support: 7, confidence: 1 |
| $a_2 b_2 \rightarrow [1]$ | support: 7, confidence: 1 |
| $b_1 c_2 \rightarrow [1]$ | support: 7, confidence: 1 |
| $b_1 c_2 \rightarrow e_4$ | support: 8, confidence: 1 |
| $a_2 b_1 \rightarrow e_4$ | support: 8, confidence: 1 |
| $a_2 e_4 \rightarrow [1]$ | support: 8, confidence: 1 |
| $b_2 [1] \rightarrow e_4$ | support: 7, confidence: 1 |
| $a_2 b_2 \rightarrow [2]$ | support: 7, confidence: 1 |
| $a_2 c_4 \rightarrow [2]$ | support: 7, confidence: 1 |

**D. COMPLEXITY COMPUTING AND ANALYSIS OF HYBRID ALGORITHM**

The cost of working with implicitly specified set of tuples can be measured in various ways. One such cost model computes the number of messages that must be exchanged among various sites (sensor nodes). Complexity for distributed query processing in databases has been discussed in [53] and this cost model measures the total data transferred for answering a query. In our case the local computation at every sensor can be ignored by the number of exchanged messages and also the amount of data transferred is very little (statistical summaries) but the number of messages exchanged may grow rapidly with the number of iterations of the proposed mining.
algorithm [54]–[58]. At each cluster in WSN, in order to extract the association rules, a number of messages need to be exchanged for the hybrid algorithm. Let us say:

1) $K$ be the number of CHs.
2) $m$ be the average number of CMs in each cluster, and
3) there are $k$-frequent item sets.

We derive below an expression for the number of messages that need to be exchanged for our proposed algorithm dealing with the implicit set of tuples as follows:

1) CREATION OF Shared RELATION (ALGORITHM 2)

the number of exchanged messages during the creation of Shared relation by every CH can be computed as below:

- $m$ messages to enquire the shared attributes between CMs.
- $m$ messages to enquire and to receive the different shared attribute items to create $P_{shared}$ relation.
- $m$ message to compute the count of each tuple in $P_{shared}$ to find the Shared relation.

2) DETERMINING $k$ FREQUENT ITEM SET

Frequent item-sets at each level of the association rule algorithm can be determined by exchanging only $2m$ messages among CMs. If an association rule algorithm needs to run up to $k$ levels, then we need to exchange a total of $2mk$ messages among the CMs to run the association rule algorithm. This number of messages is not dependent on the number of tuples contained in each database and the system, therefore, is easily scalable to large databases. Also, this number of messages is much smaller than the data that may need to be transferred if we were to accumulate all databases at one site and then perform the data mining task. Hence, the total count of exchanged messages will be as given below (Algorithms 3 and 4): Total count of messages for one cluster will be $3m + 2mk$. Therefore, the total number of messages for $K$ clusters will be:

$$\text{Total number of messages} = Km(2k + 3) \quad (9)$$

The above analysis of complexity shows that the number of messages that need to be exchanged between the CHs and their CMs which is not dependent on the size of the database at each sensor. The communication complexity is dependent primarily on the number and manner in which the attributes are shared among the sensor nodes. This is significant because it shows that as the sizes of the individual databases grow, the communication complexity of the algorithm would remain unaffected. Computational cost of local computations would grow with the database size at each individual sensor but our decomposable version has an advantage in this regard also over the transport, join, and then run the traditional Association Rule at BS. There is tremendous saving in the computational cost when the decomposable version is executed instead of moving the data, creating a Join and then running the Association Rule algorithm. Also, for the communication cost, the number of partial results that need to be transmitted is far fewer that the messages that may have to be transmitted if entire databases are collected at some central site such as BS. Another important gain of decomposable version is that it preserves the privacy of the data by not requiring any data tuples to be placed on a communication network. It also preserves the integrity of individual databases because no sensor needs to update or write into any of the participating databases. All the queries are strictly reading queries.

V. SIMULATION RESULTS

In this section, using MATLAB R2016b, we evaluate the performance of our proposed algorithm. During simulation based experimentation, we conducted two types of experiments in order to validate and evaluate the performance of the proposed approach where both of these experiments employed in the DEC [27] as a clustering algorithm in the WSN. In the first experiment, we examine the effect of support value, number of shared attributes and number of CHs upon the number of messages being exchanged. In the second experiment, the network lifetime, the number of alive nodes per round and the average remaining energy are used as metrics before and after the integration of our proposed approach with DEC [27] and CREEP [28] clustering algorithms. We consider heterogeneous multifunctional sensor nodes, where each sensor node has the ability to sense multiple attributes [59], [60]. Each sensor node maintains a flat table as a database and each column in that table represents an attribute. The attributes are assigned randomly to each sensor node from a predefined set of attributes.

A. EXCHANGED MESSAGES PARAMETERS

In this set of experiments, 100 sensor nodes are randomly deployed on a 2D-plane to monitor a region with size $100 \times 100$ m$^2$. All our experiments’ results have been obtained by averaging various topology seeds while using different set of clusters.

Support value: In this experiment, we demonstrate the effect of the selected support values on the number of messages being exchanged. The support value is varied from 0.1 to 1 with increment of 0.1 and at each value the number of messages are calculated. Fig. 4 depicts that when the support value increases, the number of messages exchanged decreases. The percentage of the number of messages being reduced is from 40% to 90% as compared to the centralized approach (or centralized extraction) in which all data is communicated to the CH, i.e., with zero support value. Hence, our proposed mining data scheme reduces the amount of communicated data and as a result, decreases the communication overhead.

Cluster head percentage: In this experiment, we used the same settings as exploited in the previous one. However, this time we vary the number of CHs from 5% to 50% with increment of 5%. We compute the number of messages at each percentage. Fig. 5 depicts the effect of the CH
percentage on the number of messages. It is evident from the figure that the number of messages goes up when the percentage of CHs increases. It is due to the fact that when the percentage of CHs increases, the messages communicated to the BS also increases. Moreover, the increase in the number of clusters induces the rise in the number of extracted rules which leads to augmented communication overhead.

**Percentage of shared attributes:** In this experiment, we employ identical setup as used in our first experiment. However, we vary the percentage of shared attributes from 5% to 50% with increment of 5%. We calculate the number of messages at each percentage. The effect of the number of shared attributes on the number of exchanged messages is depicted by Fig. 6. It is evident from the figure that increase in the percentage of shared attributes induces a decrease in the number of messages. It is due to the fact that the increase in the number of shared attributes decreases the percentage of unshared attributes; as a result, the number of messages required for determining and controlling the unshared attributes decreases.

**B. VALIDATION AND EFFECTIVENESS**

In this set of experiments, the network area is taken as 100 m × 100 m with 100 nodes randomly deployed and the BS is placed in the center of the network area. The clustering algorithms adopted are DEC [27] and Cluster-Head Restricted Energy Efficient Protocol (CREEP) [28] having 10% of nodes acting as CH nodes. The energy model and its parameters adopted are same as prescribed by [27]. The total network energy is assumed to be 102 J. All the experiments conducted use different random topology seeds. For comparison, we integrated our proposed algorithm in DEC, denoted by DEC+Proposed Approach, and also in CREEP, denoted by CREEP+Proposed Approach and we then compare them with the original DEC and the original CREEP algorithms using the following metrics.

1) **First node dies:** it is the time elapsed from the start of the experiment until the first sensor node dies.
2) **Number of alive sensor nodes per round:** it is the number of alive sensor nodes in the network after each round.
3) **Average remaining energy per round:** it is the ratio of total remaining energy of all sensor nodes to the number of nodes.

- **First Node Dies (FND):** In the first experiment, after implementing the original DEC, original CREEP, DEC + Proposed Approach, and CREEP + Proposed Approach, the experiment is conducted for different random topologies. The average of all simulation runs is demonstrated in the simulation results for the original DEC and the original CREEP and the modified versions of both, i.e., DEC+Proposed Approach and CREEP+Proposed Approach. In this experiment, the network lifetime is taken as performance metric until the FND. Fig. 7
A. M. Khedr et al.: Novel Association Rule-Based Data Mining Approach for Internet of Things Based Wireless Sensor Networks

FIGURE 7. Network lifetime in terms of FND in original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach.

Depicts that the integration of our proposed algorithm with DEC and CREEP augments the overall network lifespan until the FND, i.e., the proposed approach considerably reduces the energy depletion across all sensor nodes which is due to the fact that our proposed approach tackles the data accumulation process via mining and hence saves the energy. Moreover, extracting association rules among sensors helps to capture the set of sensors that report same data or predicate data which leads to reduction in the depleted energy.

FIGURE 8. Number of alive sensor nodes per round in the original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach.

- **Number of alive sensor nodes per round:** Fig. 8 demonstrates the number of alive nodes in the network per round in the original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach. It is evident from the figure that the number of alive nodes in the integrated DEC and integrated CREEP are more than those of the original DEC and the original CREEP algorithms. This is due to the fact that our proposed algorithm reduces the energy depletion at each sensor node by reducing the volume of data sent by each node through the distributed mining algorithm which leads to energy conservation.

- **Average remaining energy per round:** Fig. 9 depicts the residual energy after each round in the original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach. The results depict that the integrated DEC and integrated CREEP have the maximum residual energy in each round, i.e., the energy consumed by integrated DEC and integrated CREEP is less than that of the energy consumed by the original DEC and the original CREEP. The reason for the energy conservation is that our proposed approach decreases the transmissions of duplicate information.

FIGURE 9. Average remaining energy per round in the original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach.

VI. CONCLUSION AND FUTURE WORK

WSNs being the integral part of the IoT are the main sources of huge volume of data generation. This huge data, if not managed properly, would cause serious problems of resource management. In this research article, we have proposed a novel approach to manage the data efficiently, i.e., our proposed technique would mine data generated by sensors locally without communicating it to any cluster head or base station. Cluster members would communicate only statistical summaries to the cluster heads. The main idea of our proposed scheme is to confine computations to sensor nodes and to reduce the inter-node communication in order to minimize the energy wastage and overheads during communication. This was how our proposed approach increased the sensor network lifetime. Our proposed idea was backed by extensive simulations where the results obtained depict the efficiency of our scheme in terms of reduced energy depletion and prolonged network lifetime.

Our future work can be illustrated as follows: First, applying the proposed scheme for mobile WSN where the network topology is continually changing would be challenging because these changes may affect the results of the scheme.
One of the ideas to solve this is to employ the concept of fog nodes and moving cluster head operations to the fog node. Another solution is to modify the scheme to be aware of underlying clustering protocol and topology. Second, selecting improper and inflexible support and confidence thresholds could increase computation complexity, so selecting most appropriate and adaptable thresholds need more investigation. Third, the proposed scheme takes the benefits from the heterogeneity in multifunction sensors to reduce traffic size and to preserve the privacy of sensor data, but there is a required preprocessing step at each node which performs data labeling. Improper data labeling could increase the number of messages and computation costs. So, this step should be handled carefully.

REFERENCES

[1] D. Reinsel, J. Gantz, and J. Rydning, The Digitization of the World: From Edge to Core, document US44413318, 2018.
[2] P. A. Neves, J. P. C. Rodrigues, M. Chen, and A. V. Vasilakos, “A multi-channel architecture for IPv6-enabled wireless sensor and actuator networks featuring PnP support,” J. Netw. Comput. Appl., vol. 37, pp. 12–24, Jan. 2014.
[3] J.-S. Lee and C.-L. Teng, “An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems,” IEEE Internet Things J., vol. 4, no. 4, pp. 1095–1103, Aug. 2017, doi: 10.1109/JIOT.2017.2711248.
[4] H. El Alami and A. Najid, “ECH: An enhanced clustering hierarchy approach to maximize lifetime of wireless sensor networks,” in IEEE Access, vol. 7, pp. 107142–107153, 2019, doi: 10.1109/ACCESS.2019.2933052.
[5] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TinyDB: An acquisitional Query processing system for sensor networks,” ACM Trans. Database Syst., vol. 30, no. 1, pp. 122–173, Mar. 2005.
[6] Y. Yao, “Query processing in sensor networks,” in Proc. 1st Biennial Conf. Innov. Data Syst. Res. (CIDR), Asilomar, PG, USA, 2003, pp. 1–7.
[7] M. Umer, L. Kulik, and E. Tanim, “Optimizing Query processing using selectivity-awareness in wireless sensor networks,” Comput., Environ. Urban Syst., vol. 33, no. 2, pp. 79–89, Mar. 2009.
[8] L. Cheng, Y. Chen, C. Chen, J. Ma, L. Shu, and A. V. Vasilakos, “Efficient Query-based data collection for mobile wireless monitoring applications,” Comput. J., vol. 53, no. 10, pp. 1643–1657, 2010.
[9] E. Byttyä, L. Ahmed, and A. Kurti, “Association rule mining with context ontologies: An application to mobile sensing of water quality,” in Research Conference on Metadata and Semantics Research. Cham, Switzerland: Springer, 2016, pp. 78–88.
[10] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, “Data mining for wireless sensor networks,” in Proc. 7th Int’l Conf. Intell. Sensors, Adelaide, SA, USA, 2011, pp. 341–346, doi: 10.1109/ISSNIP.2011.6146592.
[11] S. Dutt, S. Agrawal, and R. Vgi, “Cluster-head restricted energy efficient protocol (CREEP) for routing in heterogeneous wireless sensor networks,” Wireless Pers. Commun., vol. 100, no. 4, pp. 1477–1497, Jun. 2018, doi: 10.1007/s11277-018-5649-x.
[12] P. Patel and M. L. A. Ali, “On using the intelligent edge for IoT analytics,” Intell. Syst., vol. 32, no. 5, pp. 64–69, 2017.
[13] M. M. Gaber, A. Aneiba, S. Basurara, O. Battu, A. M. Elmiseri, Y. Kovalchuk, and M. H. U. Rehman, Internet of Things and Data Mining: From Applications to Techniques and Systems, vol. 9. Hoboken, NJ, USA: Wiley, 2019.
[14] M. Halachev and L. Gruenwald, “Estimating missing values in related sensor data streams,” in Proc. 11th Int. Conf. Manage. Data, 2005, pp. 83–94.
[15] N. Jiang, “Discovering association rules in data streams based on closed pattern mining,” in Proc. SIGMOD Workshop Innov. Database Res., 2007, pp. 1–7.
[16] N. Jiang and L. Gruenwald, “Estimating missing data in data streams,” in Proc. Adv. Databases, Concepts, Syst. Appl., 2007, pp. 981–987.
[17] K. Loo and I. B. Tong Kao, “Online algorithms for mining inter-stream associations from large sensor networks,” in Knowledge Discovery and Data Mining, vol. 3518. Berlin, Germany: Springer, 2005.
[18] S. K. Chong, S. Krishnaswamy, S. W. Loke, and M. Gaber, “Using association rules for energy conservation in wireless sensor networks,” in Proc. 23rd Annu. ACM Symp. Appl. Comput., 2008, pp. 971–975.
[19] S. K. Tanbeer, C. F. Ahmed, and B. S. Y. Jeong Lee, “Efficient mining of association rules from wireless sensor networks,” in Proc. 11th Int. Conf. Adv. Comput. Technol., 2009, pp. 719–724.
[20] C. Qiao and K. N. Brown, “Asynchronous distributed clustering algorithm for wireless sensor networks,” in Proc. 4th Int. Conf. Mach. Learn. Technol., 2019, 76–82.
[21] E. M. Alsukhni and S. Almallahi, “Classifying environmental monitoring data to improve wireless sensor networks management,” Int. J. High Perform. Comput. Netw., vol. 12, no. 3, p. 217, 2018.
[22] M. Fuertes, A. Cadenas, J. Torres, D. Benavent, F. Tapia, and T. Bulkeridis, “Data analytics on real-time air pollution monitoring system derived from a wireless sensor network,” in Advances in Intelligent Systems and Computing, vol. 918. Cham, Switzerland: Springer, 2019.
[23] H. Soliman, K. Sudan, and A. Mishra, “A smart forest-fire early detection sensory system: Another approach of utilizing wireless sensor and neural networks,” in Proc. IFIP TC11 Conf., Nov. 2010, pp. 1–8.
[24] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, “A framework for use of wireless sensor networks in forest fire detection and monitoring,” Comput., Environ. Urban Syst., vol. 36, no. 6, pp. 614–625, Nov. 2012.
[25] M. Saoudi, A. Bounceur, R. Euler, and T. Kechadi, “Data mining techniques applied to wireless sensor networks for early forest fire detection,” in Proc. Int. Conf. Internet things Cloud Comput., New York, NY, USA, 2016, pp. 1–7.
