IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 22, 2020, accepted August 14, 2020, date of publication August 18, 2020, date of current version August 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017488

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

AHMED M. KHEDR[1,4], WALID OSAMY[2], AHMED SALIM[3,4], AND SOHAIL ABBAS[1]

[1]Department of Computer Science, University of Sharjah, Sharjah 27272, UAE
[2]Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
[3]Department of Computer Science, College of Science and Arts, Qassim University, Buridah 51931, Saudi Arabia
[4]Department 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 consummation.

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),

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

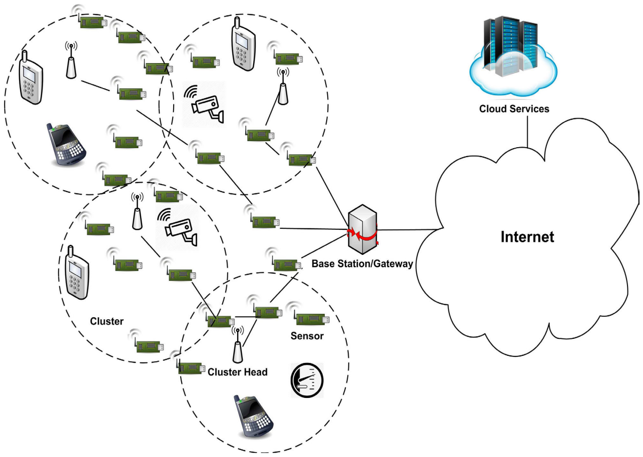

FIGURE 1. WSN integrated IoT and cloud.

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]	Fire 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 continually 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 the 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).

$$A = \bigcup_{i=1}^{n} A_i \tag{1}$$

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

$$S_{ij} = \bigcap_{q=i,j} A_q \tag{2}$$

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:

$$S = \bigcup_{i,j} S_{ij} \tag{3}$$

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.

$$R = F(D). \tag{4}$$

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:

$$R(S) = H[h_1(D_1, S), h_2(D_2, S), \ldots, h_n(D_n, S)]. \qquad (5)$$

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.

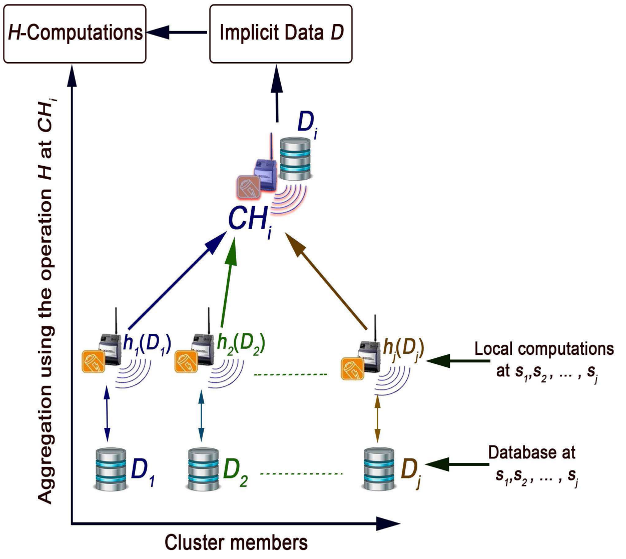
H-Computations
Implicit Data D
D_i
CH_i
$h_1(D_1)$
$h_2(D_2)$
$h_j(D_j)$
Local computations at $s_1, s_2, \ldots, s_j$
D_1
D_2
D_j
Database at $s_1, s_2, \ldots, s_j$
Aggregation using the operation H at CH_i
Cluster members

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_is. 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_is), 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.

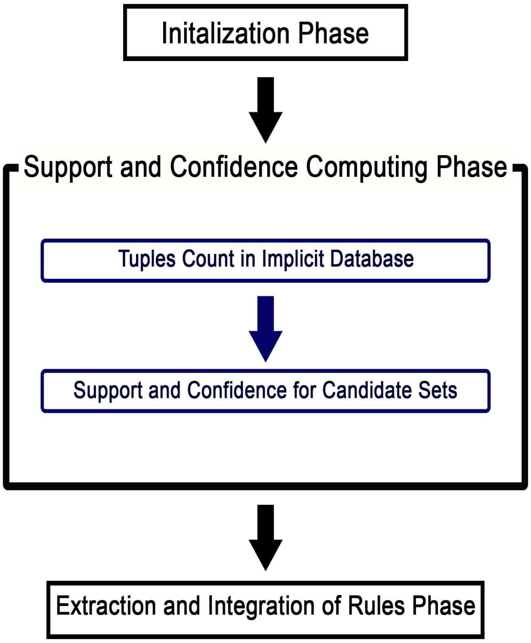
Initalization Phase
Support and Confidence Computing Phase
Tuples Count in Implicit Database
Support and Confidence for Candidate Sets
Extraction and Integration of Rules Phase

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 $s_j^i, j = 1 \ldots n$. 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 *Pshared* as the cross product

Algorithm 1 Mining Algorithm (will Be Executed by Every CH_i)

```
Call Shared Relation procedure.
Find the candidate item-sets $C_i$.
Find frequent item-sets $F_i$ by calling find-frequent item-sets procedure.
Extract Rules $R_i$ by calling Extract Rules($F$, support, confidence) procedure.
Compute the total number of tuples at $CH_i$ ($N_{total-i}$).
Send to BS: $R_i$ with confidence of each rule and $N_{total-i}$.
```

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 CH_i.

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

```
Find shared Attributes among CMs and the distinct values of each shared attribute.
$Pshared$ = Cross Product of all different values of shared attributes
Create the relation $Shared$ by removing from $Pshared$ any tuple with zero matching at any CM.
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 s_1, s_2 and s_3 with three databases D_1, D_2 and D_3 respectively such that these databases jointly determine the global implicit database $\mathcal{D}$. Moreover, we assume that s_1, s_2 and s_3 are CMs of a cluster with CH CH_1. The local databases D_1, D_2 and D_3 from the three nodes are shown in Table 2.

TABLE 2. Explicit Database Components D_1, D_2, and D_3 at sensor nodes s_1, s_2 and s_3.

D_1 at s_1		
a	**b**	**e**
a1	b1	e2
a1	b3	e4
a3	b2	e2
a2	b1	e4
a3	b1	e5
a2	b2	e2
a1	b1	e1

D_2 at s_2		
b	**c**	**f**
b1	c1	f1
b3	c2	f4
b2	c2	f9
b2	c2	f8
b2	c1	f6
b1	c1	f2
b1	c2	f8

D_3 at s_3		
a	**c**	**d**
a2	c2	d3
a1	c1	d9
a2	c1	d8
a1	c1	d10
a2	c1	d11
a2	c2	d4
a2	c1	d4

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
a2	b1	c1
a2	b1	c2
a2	b2	c1
a2	b2	c2
a2	b3	c1
a2	b3	c2

Considering only the tuples in *Pshared* that have non zero count at s_1, s_2 and s_3, the record $a1$, $b1$, $c2$ has zero value at s_3 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
4	a2	b2	c2

B. SUPPORT AND CONFIDENCE COMPUTING PHASE

This phase is implemented by each CH_i 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 CH_i will initiate the execution of the Support and Confidence Computing phase. CH_i 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 CH_i.

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 D_i's regarding the local counts. The corresponding replies from the CMs are then used to determine $N_{total}(D)$, i.e., the total tuples in D. This can be expressed as follows:

$$N_{total}(D) = \sum_j \prod_t (N_{D_t})_{cond_j} \tag{6}$$

where, $cond_j$ denotes the attribute-value condition for tuple j^{th} belonging to *Shared* relation, $N(D_t)_{cond_j}$ denotes the tuples count in D_t at CM s_t^i that satisfies $cond_j$. Based on Equation 6, we can write:

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

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 $N(D_t)_{cond_J}$ into an SQL query; select count (*), such that $cond_j$ can be executed by CM (s_t^i).

In the example mentioned above, the count of each shared tuple can be computed using Equation 6 by taking $cond_j$ as the items of each tuple in shared for example the first tuple of shared relation, $cond_j$ will be (a = a1, b = b1, c = c1) which implies $N(D_1)_{cond_J} = 2$, $N(D_2)_{cond_J} = 2$, and $N(D_3)_{cond_J} = 2$ 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**	N(D1)	N(D2)	N(D3)	$\prod_t (N_{D_t})^{cond_j}$
0	a1	b1	c1	2	2	2	8
1	a2	b1	c1	1	2	3	6
2	a2	b1	c2	1	1	2	2
3	a2	b2	c1	1	1	3	3
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.

$$N_{new-condition} = \sum_j \prod_{t=1}^{n} N(D_t)_{cond_j \ and \ new-condtion} \tag{8}$$

The method in which CH_i finds the support measure for a candidate frequent item-set is as described below. In relation *Shared*, CH_i 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

```
k=1
while $C_k$ is not empty do
    for every $I_i \in C_k$ do
        $I_i.count = 0$
        Construct $new - condtion_i$
        for every database at $CM_t$ $D_t$ do
            d=0, j=1
            for every shared tuple$j$ $\in$ Indexed Shared relation do
                $j = j * N(D_t)_{cond_j}$ and $new{-}condtion_i$
            end for
            d=d+j
        end for
        $I_i.count = d$
    end for
    $F_k \leftarrow \{I_i \in C_k | I_i.count \geq minsup\}$, k=k+1
    $C_k$ =generateCandidate($F_{k-1}$)
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., $C_1 = \{a1, a2, b1, b2, c1, c2, e1, e2, e4, d3, d4, d8, d9, d10, d11, f1, f2, f6, f8, f9\}$.

  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 items a=a1, a=a2, b=b1, b=b2, c=c1, c=c2, e=e1,e=e2,e=e4, d=d3,etc.

Algorithm 4 Generate Candidate (will Be Executed by Each CH_i)

```
Input: $F_{k-1}$ {frequent item-sets}
Output: $C_k$ {the set of candidates}
$C_k \leftarrow \phi$
for i=1:$|F_{k-1}|$ do
    if $k < 2$ then
        for j= i+1:$|F_{k-1}|$ do
            $C_k \leftarrow F_{k-1}(i) \bigcup F_{k-1}(j)$
        end for
    else
        for j=i+1:$|F_{k-1}|$ do
            if $F_{k-1}(i) \bigcap F_{k-1}(j) \neq \phi$ then
                for r=1:$|C_k|$ do
                    if $(F_{k-1}(i) \bigcup F_{k-1}(j)) == C_k(r)$ then
                        break
                    else
                        $C_k \leftarrow F_{k-1}(i) \bigcup F_{k-1}(j)$
                    end if
                end for
            end if
        end for
    end if
end for
```

For example, the frequency of a=a1 can be computed as follows: in Equation 8, $new - condition$ = {a=a1} and $cond_j$=shared tuple with index $j, j = 0, 1, 2, 3, 4$.
$N_{\{a=a1\}} = 2 \times 2 \times 2 + 0 \times 2 \times 3 + 0 \times 1 \times 2 + 0 \times 1 \times 3 + 0 \times 2 \times 2 = 8$.
the frequency of e = e2 can be computed as follows: in Equation 8, new-condition = {e=e2} and $cond_j$ = shared tuple with index $j, j = 0, 1, 2, 3, 4$.
$N_{\{e=e2\}} = 1 \times 2 \times 2 + 0 \times 2 \times 2 + 0 \times 1 \times 2 + 1 \times 1 \times 2 + 1 \times 2 \times 2 = 11$. Similarly, the number of tuples containing the other non-shared items are given in Table 6

TABLE 6. Non-shared counts.

Item	Count	Item	Count
e1	4	d8	3
e2	11	d9	4
e4	8	d10	4
d3	3	d11	3
d4	4	f1	7
f2	7	f6	7
f8	3	f9	3

- According to our threshold (support value), the set of frequent 1-item sets of C_1 will be $F_1 = \{a1, a2, b1, b2, c1, e2, e4, f1, f2, f6\}$. The set of candidate 2-item-sets (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., (a1, a2) is invalid combination. As a result the combinations will be as follows:
{a2,b1}, {c1,e1}, {e2,f1}, {b2,c1 },
{b2,e1 }, {f1,f2 }, {a1,b1 }, {c1,e2 },
{e1,f1 }, {a2,b2 }, {b1,e1 }, {e2,f2 },
{a1,c1 }, {b2,e2 }, {e1,f2 }, {a1,b2 },
{c1,f1 }, {e4,f1 }, {a2,c1 }, {a1,c1 },
{e4,f2 }, {a2,e1 }, {b1,e1 }, {b2,f1 },
{c1,f2 }, {a1,e1 }, {a2,e2 }, {b1,f1 },
{b2,f2 }, {a1,e2 }, {a2,f1 }, {b1,f2 },
{a1,f1 }, {a2,f2 }, {a1,f2 }, {c1,e4 },
{b1,e4 }, {a2,e4 }, {a1,e4 }.
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 {e1,f1} will be as follows: $new - condition$ = {e=e1 and f=f1} and $cond_j$=shared tuple with index $j, j = 0, 1, 2, 3, 4$. And so $N_{\{e=e1\ and\ f=f1\}} = 1 \times 1 \times 2 + 0 \times 1 \times 3 + 0 \times 0 \times 2 + 0 \times 0 \times 3 + 0 \times 0 \times 2 = 2$. Similarly, the number of tuples containing the other non-shared items are given in Table 7

TABLE 7. F_2 itemsets.

Item	Count	Item	Count
{a2,b1}	8	{c1,e1}	4
{b2,c1}	7	{b2,e1}	0
{a1,b1}	8	{c1,e2}	7
{a2,b2}	7	{b1,e1}	4
{b1,c1}	14	{b2,e2}	7
{a1,b2}	0	{c1,f1}	7
{a2,c1}	9	{a2,e1}	0
{a1,c1}	8	{b1,e2}	4
{e2,f1}	2	{b2,f1}	0
{e1,f1}	2	{c1,f2}	7
{e2,f2}	2	{a1,e1}	4
{e1,f2}	2	{a2,e2}	7
{e4,f1}	3	{b1,f1}	7
{e4,f2}	3	{b2,f2}	0
{a1,e4}	0	{a1,e2}	4
{a2,e4}	8	{a2,f1}	3
{b1,e4}	8	{b1,f2}	7
{c1,e4}	6	{a1,f1}	4
{a1,f2}	4	{a2,f2}	3

- 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., {a1, a2, a1} is an invalid combination. Also, any two 2-itemsets from F_2 that intersect with others, e.g., {a2, b1} and {c1, e2} cannot form new combination of 3-item-set. As a result, the candidate of 3-itemsets combinations are as listed as follows:
{a2,b1,c1 }, {a2,b1,e2 }
{a2,b2,c1 }, {a2,b2,e2 }
{a2,c1,f1 }, {a2,c1,f2 }
{a1,c1,f1 }, {c1,e2,b2 }
{c1,e2,a2 }, {b2,e2,a2 }
{a2,b1,e4 }, {a1,b1,e4 }
{a2,c1,e4 }, {a1,b1,c1 }
{a2,c1,e2 }, {a1,c1,e2 }
{c1,e2,f2 }, {b2,f1,e4 }
{a2,b2,e4 }
- 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
a1,b1,c1	8
a2,b2,e2	7
a2,c1,e2	7
a2,b1,e4	7

- The combination from F_3 is $C_4 = a2, b2, c1, e2$ and the count of this item-set is 3 and as a result $F3 = \phi$ 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)

```
Input: $F_i$: large item-sets, support, confidence
Output: $R^i$: Association Rules satisfying support and confidence at $CH_i$.
$R = \phi$.
for each $f \in F_i$ do
    for each $c \subset f | c \neq \phi, c \neq f$ do
        if $\frac{support(f)}{support(c)} \geq confidence$ then
            $R^i = R^i \bigcup \{c \Rightarrow (f - c)\}$
        end if
    end for
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

$$\frac{support(f)}{support(c)} \geq confidence$$
$$(where\ confidence\ is\ a\ threshold).$$

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

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 δ be the total tuples count obtained from k clusters.

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

```
$N_{total-j}$ is number of tuples at cluster j.
Let $confidence(R_i^j)$ is the confidence of rule $R_i$ at cluster $j$.
Let $\delta$ is the total number of tuples obtained from $k$ clusters.
$R = R^1 \bigcup \cdots \bigcup R^k$ { all rules that received from k clusters}
$\delta = \Sigma_{i=1}^{k} num\_of\_tuples_i$
for each rule $R_i \in R$ do
    $W_i = \sum_j^k confidence(R_i^j) \times \frac{num\_of\_tuples_j}{\delta}$
end for
Select rules that satisfy the weight threshold value
```

- $R = R_1 \bigcup \cdots \bigcup R_k$.
- for every $r_l^m \in$ R
  - for $i = 1$ to k
    - ∗ if r_l^m in R_i possess confidence c_i^t
    - ∗ $c_l^m = c_l^m + (N_i/\delta) * c_i^t$

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

TABLE 9. **Association Rules.**

Rules
[a1] → [b1 c1]; support: 8, confidence: 1
[a1 b1] → [c1]; support: 8, confidence: 1
[a1 c1] → [b1]; support: 8, confidence: 1
[f1] → [b1 c1]; support: 7, confidence: 1
[b1 f1] → [c1]; support: 7, confidence: 1
[c1 f1] → [b1]; support: 7, confidence: 1
[f2] → [b1 c1]; support: 7, confidence: 1
[b1 f2] → [c1]; support: 7, confidence: 1
[c1 f2] → [b1]; support: 7, confidence: 1
[e4] → [a2 b1]; support: 8, confidence: 1
[a2 b1] → [e4]; support: 8, confidence: 1
[a2 e4] → [b1]; support: 8, confidence: 1
[b1 e4] → [a2]; support: 8, confidence: 1
[b2] → [a2 e2]; support: 7, confidence: 1
[a2 b2] → [e2]; support: 7, confidence: 1
[a2 e2] → [b2]; support: 7, confidence: 1
[b2 e2] → [a2]; 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 messagees 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 hybird 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 *Pshared* relation.
- m message to compute the count of each tuple in *Pshared* 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 $3 * m + 2 * m * k$. Therefore, the total number of messages for K clusters will be:

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

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

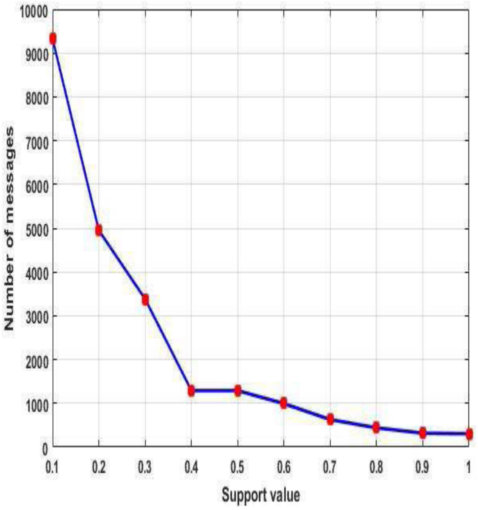

FIGURE 4. Number of messages versus support values.

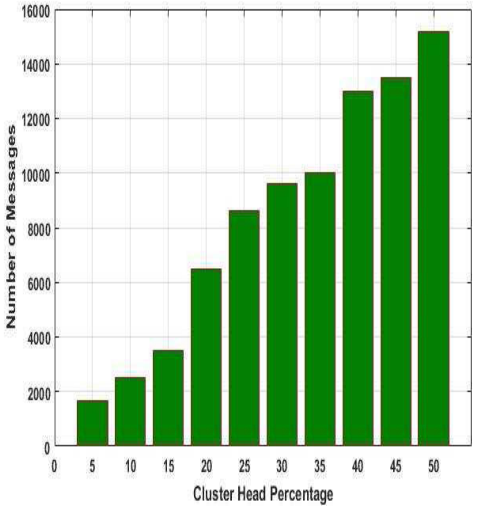

FIGURE 5. Percentage of CHs versus number of messages.

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.

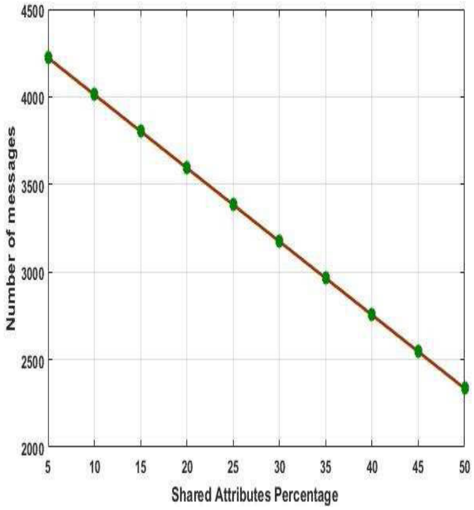

FIGURE 6. Number of messages versus percentage of shared attributes.

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 $102J$. 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, orginal 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

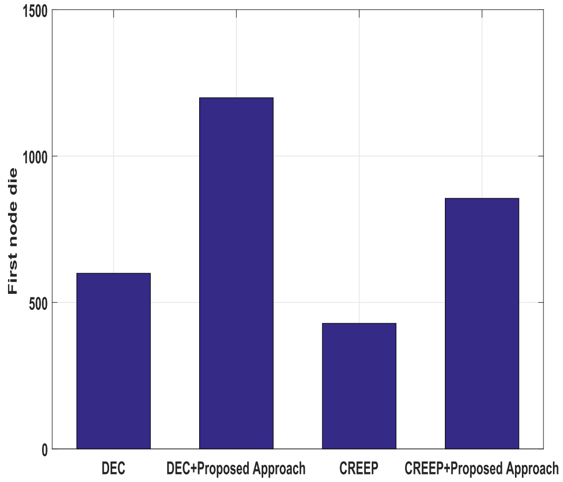

FIGURE 7. Network lifetime in terms of FND in original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach.

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.

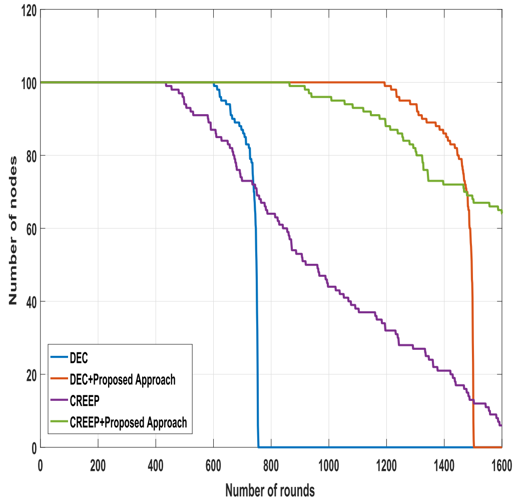

FIGURE 8. Number of alive nodes per round in the original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach.

- **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.

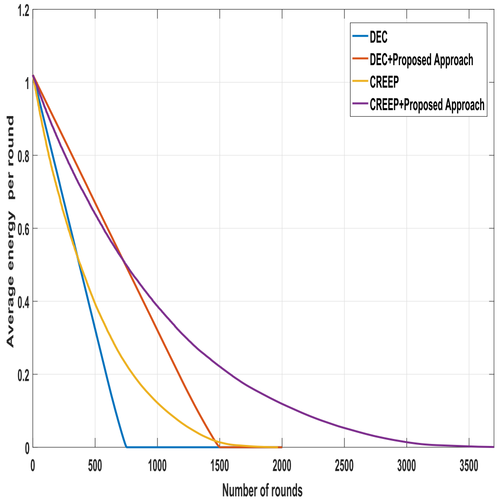

FIGURE 9. Average remaining energy per round in the original DEC, original CREEP, DEC+Proposed Approach, CREEP+Proposed Approach.

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. 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. Tanin, "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. Bytyãi, L. Ahmedi, 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. 67–78.

[10] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, and L. T. Yang, "Data mining for Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 77–97, 4th Quart., 2014.

[11] M. Z. Ge, H. Bangui, and B. Buhnova, "Big data for Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 87, pp. 601–614, Oct. 2018.

[12] J. G. Pottie and J. W. Kaiser, "Wireless integrated network sensor," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[13] R. Kacimi, "Energy conservation techniques for wireless sensor networks," Ph.D. dissertation, Dept. Math. Inf. Telecom., INPT Univ., Toulouse, France, 2009.

[14] C. Dini, "Les re seaux capteurs sans fil avec access sporadique au noeud-puits," Ph.D. dissertation, Inf. Eng., Haute Alsace Univ., Mulhouse Cedex, France, 2010.

[15] C. Dini and P. Lorenz, "Primitive operations for prioritized data reduction in wireless sensor network nodes," in *Proc. 4th Int. Conf. Syst. Netw. Commun.*, Sep. 2009, pp. 274–280.

[16] A. Aziz, K. Singh, W. Osamy, and A. M. Khedr, "Effective algorithm for optimizing compressive sensing in IoT and periodic monitoring applications," *J. Netw. Comput. Appl.*, vol. 126, pp. 12–28, Jan. 2019.

[17] D. M. Omar and A. M. Khedr, "ERPLBC: Energy efficient routing protocol for load balanced clustering in wireless sensor networks," *Ad Hoc Sensor Wireless Netw.* vol. 42, pp. 145–169, Oct. 2018.

[18] Omar D M, Khedr A. M., Agrawal, "Optimized clustering protocol for balancing energy in wireless sensor networks," *Int. J. Commun. Netw. Inf. Security.*, vol. 9, no. 3, pp. 367–375, 2017.

[19] W. Osamy and A. M. Khedr, "An algorithm for enhancing coverage and network lifetime in cluster-based wireless sensor networks," *Int. J. Commun. Netw. Inf. Secur.*, vol. 10, no. 1, pp. 1–9, 2018.

[20] N.-N. Qin, L. Zhang, and B.-G. Xu, "The coverage force algorithm for heterogeneous wireless sensor networks," *J. Electron. Inf. Technol.*, vol. 2010, no. 1, pp. 189–194, Feb. 2010.

[21] A. M. Khedr and D. M. Omar, "SEP-CS: Effective routing protocol for heterogeneous wireless sensor networks," *Ad Hoc Sensor Wireless Netw.* vol. 26, pp. 211–232, Oct. 2015.

[22] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *IEEE Wireless Commun.*, vol. 14, no. 2, pp. 70–87, Apr. 2007.

[23] L. Shu, J. Lloret, J. C. Rodrigues, and M. Chen, "Distributed intelligence and data fusion for sensor system," *IET Commun.* vol. 5, no. 12, pp. 1633–1636, 2011.

[24] C. J. Debono and N. P. Borg, "The implementation of an adaptive data reduction technique for wireless sensor networks," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Dec. 2008, pp. 1–7.

[25] W. Osamy, A. Salim, and A. M. Khedr, "An information entropy based-clustering algorithm for heterogeneous wireless sensor networks," *Wireless Netw.*, vol. 26, no. 3, pp. 1869–1886, Apr. 2020, doi: 10.1007/s11276-018-1877-y.

[26] A. Salim, W. Osamy, and M. Ahmed Khedr, "IBLEACH: Effective LEACH protocol for wireless sensor networks, wireless networks," *Wireless Netw.*, vol. 20, pp. 1515–1525, 2014.

[27] F. A. Aderohunmu, J. D. Deng, and M. K. Purvis, "A deterministic energy-efficient clustering protocol for wireless sensor networks," in *Proc. 7th Int. Conf. Intell. Sensors*, Adelaide, SA, USA, 2011, pp. 341–346, doi: 10.1109/ISSNIP.2011.6146592.

[28] S. Dutt, S. Agrawal, and R. Vig, "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.

[29] P. Patel and M. I. A. Ali Sheth, "On using the intelligent edge for IoT analytics," *Intell. Syst.*, vol. 32, no. 5, pp. 64–69, 2017.

[30] M. M. Gaber, A. Aneiba, S. Basurra, O. Batty, A. M. Elmisery, 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.

[31] M. Halatchev and L. Gruenwald, "Estimating missing values in related sensor data streams," in *Proc. 11th Int. Conf. Manage. Data*, 2005, pp. 83–94.

[32] N. Jiang, "Discovering association rules in data streams based on closed pattern mining," in *Proc. SIGMOD Workshop Innov. Database Res.*, 2007, pp. 1–7

[33] N. Jiang and L. Gruenwald, "Estimating missing data in data streams," in *Proc. Adv. Databases, Concepts, Syst. Appl.*, 2007, pp. 981–987.

[34] 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.

[35] 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.

[36] 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. Commun. Technol.*, 2009, pp. 719–724.

[37] 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.

[38] 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.

[39] W. Fuertes1, A. Cadena1, J. Torres, D. Benátez, F. Tapia, and T. Toulkeridis, "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.

[40] 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. IEEE Sensors*, Nov. 2010, pp. 1–8.

[41] 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.

[42] 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.

[43] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, "IoT and agriculture data analysis for smart farm," *Comput. Electron. Agricult.*, vol. 156, pp. 467–474, Jan. 2019.

[44] A. K. Tripathy, J. Adinarayana, K. Vijayalakshmi, S. N. Merchant, U. B. Desai, S. Ninomiya, M. Hirafuji, and T. Kiura, "Knowledge discovery and leaf spot dynamics of groundnut crop through wireless sensor network and data mining techniques," *Comput. Electron. Agricult.*, vol. 107, pp. 104–114, Sep. 2014.

[45] A. Kamilaris, A. Kartakoullis, and F. X. Prenafeta-Boldá, "A review on the practice of big data analysis in agriculture," *Comput. Electron. Agricult.*, vol. 143, pp. 23–37, Dec. 2017.

[46] M. Rida, A. Makhoul, H. Harb, D. Laiymani, and M. Barhamgi, "EK-means: A new clustering approach for datasets classification in sensor networks," *Ad Hoc Netw.* vol. 84, pp. 158–169, Dec. 2019.

[47] P. Zou and Y. Liu, "A data-aggregation scheme for WSN based on optimal weight allocation," *J. Netw.*, vol. 9, no. 1, pp. 100–107, Jan. 2014.

[48] H. Harb, A. Makhoul, and R. Couturier, "An enhanced K-means and ANOVA-based clustering approach for similarity aggregation in underwater wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 10, pp. 5483–5493, Oct. 2015.

[49] M. M. Rashid, I. Gondal, and J. Kamruzzaman, "Mining associated patterns from wireless sensor networks," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 1998–2011, Jul. 2015.

[50] D. Yong-wen, X. Ning, L. Hui-ling, L. Wei, and F. Ke, "A hierarchical time synchronization algorithm for WSN," *Procedia Comput. Sci.*, vol. 131, pp. 1064–1073, Oct. 2018, doi: 10.1016/j.procs.2018.04.260.

[51] X. Huan and K. S. Kim, "On the practical implementation of propagation delay and clock skew compensated high-precision time synchronization schemes with resource-constrained sensor nodes in multi-hop wireless sensor networks," *Comput. Netw.*, vol. 166, Jan. 2020, Art. no. 106959, doi: 10.1016/j.comnet.2019.106959.

[52] B. Wang and Y.-P. Tian, "Time synchronization in WSNs with random communication delays: A constant gain design 1 1This work is supported by the national natural science foundation of China (under grants 61573105, 61273110)," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 657–662, Jul. 2017.

[53] C. Wang and M.-S. Chen, "On the complexity of distributed Query optimization," *IEEE Trans. Knowl. Data Eng.*, vol. 8, no. 4, pp. 650–662, Oct. 1996.

[54] A. M. Khedr, "Decomposable algorithm for computingk-nearest neighbours across partitioned data," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 31, no. 4, pp. 334–353, Jul. 2016.

[55] A. M. Khedr, "New algorithm for clustering distributed data using k-means," *Comput. Inform.* vol. 33, pp. 1001–1022, Apr. 2014.

[56] A. M. Khedr and R. Mahmoud, "Agents for integrating distributed data for function computations," *Comput. Informat.*, vol. 31, pp. 1101–1125, Oct. 2012.

[57] A. M. Khedr, "Nearest neighbor clustering over partitioned data," *Comput. Informat.*, vol. 30, pp. 1001–1026, Oct. 2011.

[58] A. Khedr and M. Salim, "Decomposable algorithms for finding the nearest pair," *J. Parallel Distrib. Comput.*, vol. 68, pp. 902–912, Oct. 2008.

[59] M. Xie, K. Hisano, M. Zhu, T. Toyoshi, M. Pan, S. Okada, O. Tsutsumi, S. Kawamura, and C. Bowen, "Flexible multifunctional sensors for wearable and robotic applications," *Adv. Mater. Technol.*, vol. 4, no. 3, Mar. 2019, Art. no. 1800626.

[60] N. Pallaro, F. Visintainer, M. Darin, E. Balocco, E. Borello, M. Gottardi, and N. Massari, "Multifunctional sensor to detect environmental parameters," in *Proc. Sensors Microsyst.* Berlin, Germany: Springer, Feb. 2004, pp. 247–265.

AHMED M. KHEDR received the B.Sc. degree in mathematics and the M.Sc. degree in optimal control from Zagazig University, Egypt, in June 1989 and July 1995, respectively, and the M.Sc. and Ph.D. degrees in computer science and engineering from the University of Cincinnati, Cincinnati, OH, USA, in July 1999 and March 2003, respectively. From March 2003 to January 2004, he was a Research Assistant Professor with the Department of ECECS, University of Cincinnati. From January 2004 to May 2009, he worked as an Assistant Professor with Zagazig University. From September 2009 to September 2010, he worked as an Associate Professor with the Department of Computer Science, College of Computers and Information Systems, Taif University, Saudi Arabia. Since December 2014, he has been a Professor with Zagazig University. From September 2010 till December 2019, he worked as an Associate Professor, and since January 2020, he has also been a Professor with the Department of Computer Science, College of Computing and Informatics, University of Sharjah, UAE. He was awarded the State Prize of distinction in advanced technology, the Sharjah Islamic Bank prize of distinction in research, and the University of Sharjah prize of distinction in research, in June 2009, May 2013, and April 2014, respectively. His research interests include wireless sensor networks, the Internet of Things, and distributing computing.

WALID OSAMY received the B.Sc. (Hons.), M.S., and Ph.D. degrees in computer science from the Faculty of Science, Zagazig University, Egypt. He has been involved in the projects of network infrastructure, and management information systems with the Communication Information Technology Center (CITC), Zagazig University. From 2010 to 2019, he had been an Assistant Professor with the Department of Computer Science, Faculty of Computers and Informatics, Benha University, Egypt. Since 2015, he has been an Assistant Professor with Qassim University, Buridah, Saudi Arabia. Since 2019, he has also been an Associate Professor with the Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University. His research interests include computational intelligence and in the field of IoT (mobile computing and wireless sensor networks (WSNs)).

AHMED SALIM received the B.Sc. degree in computer science and the M.Sc. degree in distributed computing from Zagazig University, Egypt, in 2001 and 2006, respectively, and the Ph.D. degree in systems, network, and telecommunication devices from the Bonch-Bruvich University of Telecommunication, Saint-Petersburg, Russia, in 2010. In 2012, he was a Consultant of the Information and Communication Technology Project (ICTP), Zagazig University. From 2011 to 2019, he was an Assistant Professor with the Department of Mathematics, Faculty of Science, Zagazig University, where he has been an Associate Professor, since 2019. Since 2014, he has also been an Assistant Professor with the Department of Mathematics, Faculty of Science and Art, Qassim University, Saudi Arabia. His research interests include decomposable algorithms, computing, the IoT, and wireless sensor networks.

SOHAIL ABBAS received the Ph.D. degree in wireless network security from Liverpool John Moores University, U.K., in 2011. He is currently working as an Assistant Professor with the Department of Computer Science, College of Sciences, University of Sharjah, UAE. He has been involved in academia for more than 14 years and in research for more than ten years. His research interests include security issues, such as intrusion detection, identity-based attacks, and trust in wireless networks, such as mobile ad hoc networks, wireless sensor networks, and the Internet of Things. He is a member of various technical program committees, including IEEE CCNC, IEEE VTC, IEEE ISCI, IEEE ISWTA, and so on. He is also serving various prestigious journals as a Reviewer, such as *Security and Communication Networks*, *IET Wireless Sensor Systems*, *Mobile Networks and Applications*, the *International Journal of Electronics and Communications*, and the *International Journal of Distributed Sensor Networks*.