

An Energy Efficient Intelligent Method for Sensor Node Selection to Improve the Data Reliability in Internet of Things Networks

Remesh Babu K R^{1*}, Preetha K G², Saritha S² and Rinil K R³

¹Government Engineering College, Sreekrishnapuram,
Kerala, India
[remeshbabu@yahoo.com]

²Rajagiri School of Engineering & Technology,
Kerala, India
[preetha_kg@rajagiritech.edu.in]
[saritha_s@rajagiritech.edu.in]

³College of Engineering, Thalassery
Kerala, India
[rinilkr2000@gmail.com]

^{1*}Corresponding author: Remesh Babu K R

*Received March 15, 2021; revised May 26, 2021; revised July 16, 2021; accepted August 9, 2021;
published September 30, 2021*

Abstract

Internet of Things (IoT) connects several objects with embedded sensors and they are capable of exchanging information between devices to create a smart environment. IoT smart devices have limited resources, such as batteries, computing power, and bandwidth, but comprehensive sensing causes severe energy restrictions, lowering data quality. The main objective of the proposal is to build a hybrid protocol which provides high data quality and reduced energy consumption in IoT sensor network. The hybrid protocol gives a flexible and complete solution for sensor selection problem. It selects a subset of active sensor nodes in the network which will increase the data quality and optimize the energy consumption. Since the unused sensor nodes switch off during the sensing phase, the energy consumption is greatly reduced. The hybrid protocol uses Dijkstra's algorithm for determining the shortest path for sensing data and Ant colony inspired variable path selection algorithm for selecting active nodes in the network. The missing data due to inactive sensor nodes is reconstructed using enhanced belief propagation algorithm. The proposed hybrid method is evaluated using real sensor data and the demonstrated results show significant improvement in energy consumption, data utility and data reconstruction rate compared to other existing methods.

Keywords: Internet of Things, hybrid protocol, data quality, energy consumption.

1. Introduction

Any physical object with embedded sensors can now share information very quickly, thanks to the Internet of Things (IoT). The world is moving towards smarter Cyber-Physical Systems (CPS) with more sophisticated intelligent devices, tremendous of information is available from these devices and it is very difficult to handle and remember these information in our day to day life [1][2]. The development of IoT sensor network starts in military application and now it is extended to home, industry, healthcare and agricultural applications. In many of the applications, tiny sensor nodes are densely deployed in various unsupervised locations [3][4]. The IoT sensor networks are data centric and there is instantaneous generation of huge data from IoT devices, which is difficult to handle and store, and hence it is desirable to make a framework to address those issues and storage of continuously generating data. Homogeneous data from thousands of sensors are not structured always, so that storage, retrieval, management, and extracting reliable information from the available data set are prominent issues in IoT networks. A large number of sensor nodes must be installed in order to improve the precision of the sensed data. Densely deployed sensor nodes produces massive amount of data. It is difficult to collect data from all of the sensor nodes [5], but there are a number of methods available in the literature for obtaining data from a subset of nodes. Since the data quality and reliability depends on energy level of the sensor nodes, it is necessary to find a solution for improving the energy of the sensor nodes.

The accuracy of the data may not be same for all the sensors deployed in the same area. Selection of reliable data among the available set is a herculean task. The selection of the reliable sensor nodes always depends on the user application. This paper proposes an intelligent approach for selecting sensor nodes in order to save energy and money by deactivating the sensors with the least amount of activity. Sensor nodes usually collect a large amount of data, which is then stored in the cloud. If only accurate and reliable data is stored, the amount of cloud storage required is significantly reduced, and it provides better computing resource management and improve the quality of service (QoS) in cloud computing [6].

The proposed hybrid protocol makes an effort to bring down the energy consumption and to ensure the data quality in the sensor data. The protocol works in two stages (i) Sensor Selection (SS) and (ii) Data Reconstruction (DR). SS stage starts with the deployment of large number of sensor nodes in the specific area and a network is created which collects the required information from the sensor nodes. The active sensor nodes are selected using Ant Colony Optimization (ACO) inspired variable path sensor selection algorithm and the data dissemination is optimized by using Dijkstra's shortest path algorithm. ACO is a good probabilistic method to find the optimal path in a network [7]. During the SS stage, only subset of sensor nodes, which are reliable, are turned to be on and others are turned off to preserve the energy in the sensor nodes. In the DR stage, Enhanced Belief Propagation (EBP) Algorithm is used to compute the belief on each sensor nodes. To boost data efficiency, EBP collects missing data due to inactive nodes. The proposed hybrid protocol is tested on the real sensor data collected from 54 sensors installed in the Intel Berkeley Research Laboratory [8] and result shows that the new method perform much better in terms of data utility, overall energy consumption and error in prediction.

The rest of this paper is organized as follows. The Section II gives the related research reported in the area of energy conservation and sensor selection methods in IoT sensor networks. The proposed hybrid method to improve the energy efficiency of sensor nodes is

described in Section III. Section IV present the experimental setup and validation of results and a conclusion is given in Section V.

2. Related Works

Over the last decade, advances in sensor nodes and hand-held computing devices have brought revolution in the area of IoT networks. IoT has grown in diversity of applications and the areas pertaining to energy management, data security, data reliability and data delivery have always attracted the attention of researchers all over the world. The authors in [9] proposed a balanced energy consumption architecture for IoT which uses relay nodes to reduce the computational difficulty of the sensors and avoid quick battery drain problems. Sensor nodes are used for sensing the data and relay nodes for data communication. This approach has less complexity with reduced energy consumption using node placement and a routing mechanism. To attain high connectivity in the network, routing relays are placed between single hop neighbours and the number of relay nodes depends on the network traffic. Here optimal path discovery is based on the residual energy of sensor nodes. If the residual energy is low, there is a chance of disconnecting the link and thus leads to data loss. The usage of relay mechanism is energy efficient, but does not guarantee the data quality.

Compressive sensing is another way of reducing the volume of sensed data. A large-scale data aggregation mechanism which uses a compressed sensing technique to achieve low network traffic and longer lifetime is proposed in [10]. It uses the compressed sensing for the intra-source correlation among sensor readings. It also includes a data gathering and recovery design which uses both intra and inter-source data correlations among the different sensor readings. The system is robust against measurement and communication noises without introducing excessive computation on sensor nodes but has high energy utilization. For the proper reconstruction of lost data, the paper [11] proposed an algorithm, which is integrated with space and time features, while the compressive sensing approach in [12] uses a complete environmental matrix to perform the inference function.

The caching mechanism proposed in [13] is used for energy harvesting for IoT sensing process. The system earns energy from the environment which is used for sensing and transmitting the data. Energy consumption of sensor nodes can be reduced by the use of sensor cache. Wireless gateway would be the perfect place for implementing sensing cache, which helps to avoid frequent activation of sensors. The threshold adaptation algorithm allows the sensing cache to adjust the parameter to increase the combined hit rate of the sensing services from different sensors. Caching is used for filtering the requests to sensors which also helps to preserve energy. A timer is used to keep the fresh data in the cache. For the optimal performance, the threshold adaptation algorithm is designed as a learning algorithm, but it is complex.

Adaptive sensing algorithm with belief propagation protocol (ASBP), presented in paper [14], is used to reduce energy consumption and provide high data quality for IoT sensors. In this method, the base station calculates the coefficient correlation matrix from environment matrix to find link quality, correlation and shortest path. The base station uses these data as inputs to solve the sensor selection optimization problem formulated in phase one. The sensor selection optimization problem is solved using constraint programming. Similar idea is also used in HOLA [15] and this is a heuristic and opportunistic algorithm for link selection, which reduces and balances energy consumption.

Bluetooth Low Energy (BLE) architecture reported in [16] uses ultra-low power wireless standard and a hybrid topology which reduces battery usage, and thus increases the lifespan of

a sensor node. BLE performs faster connection establishment process, but it is suitable only in low power applications. The Bayesian rule-based Expectation Maximization (EM) algorithm [17] finds reliable sensors based on the environmental features. It has higher accuracy, but is time consuming. In the distributed compression method [18], the sensor data is compressed based on the past data and data from other sensor nodes. Here adaptive prediction is used to find the data correlation and to estimate the number of bits required for the data compression source coding. The limitation of this method is that the compression is done independently at different levels without considering what is being sensed by the other sensor nodes.

The Constrained Application Protocol (CoAP) [19] addresses the issues regarding packet sizes, time-to-live (TTL) and multicasting challenges in IoT. A layered structure of IoT networks in which the topmost layers are handled by virtual machines is proposed in [20]. The main attraction of this cloud assisted method is the reduced power consumption and traffic. Here each network device can act as virtual machine and the controlling and aggregation of IoT traffic is the responsibility of the virtual machines. The total power consumption is reduced in this method.

The multi-objective particle swarm optimization framework proposed in [21] mainly balances the hiding failure, missing cost, artificial cost and database dissimilarity. The method provides a secure platform for data sharing but does not make an attempt for energy reduction. Authors in [22] formulate a secure outage probability (SOP) and the strictly positive secure capacity (SPSC) to examine the secrecy performance and various controlling parameters.

In spite of the rising interest and wide acceptability, the IoT networks throws serious challenges and inadequacies in data selection and communication. The review of literature section brings out clearly the requirement of well-designed and effective method to reduce the energy consumption and proper data reconstruction in IoT networks.

3. Proposed Hybrid Protocol

While reviewing the literature, it was observed that there are many methods developed for improving the efficiency of IoT networks, but none are optimal and requires further explorative study and development.

Sensors in the same domain can communicate with each other easily and heterogeneous sensors in the different domain can also be able to communicate by using suitable protocol. When thousands of sensors are deployed for different purposes on a wide area, it is not feasible to rely on the huge data collected from all these sensors. IoT devices independently sense environment parameters like temperature, humidity, and wind speed, etc., which may vary with sensors. Single sensor sense only one parameter whereas multi-sensors sense multiple environment conditions. But, for predicting some specific conditions, such as weather, multiple parameters are needed, including temperature and humidity.

In this work, an IoT network is established with multi-sensors which are distributed over a large geographical area and these sensors are capable of sensing m different environmental parameters $\{p_1, p_2, \dots, p_m\}$. Let $D_{ij}(t)$ denotes the value of j^{th} parameter from the i^{th} sensor at time t , then the data from the sensor network is represented as a set shown in equation (1).

$$D_{ij}(t) = \{D_{i1}(t), D_{i2}(t), \dots, D_{in}(t)\}; \text{ where } i = 1, 2, \dots, m. \quad (1)$$

The following subsections describe the details of network model, heuristic function, sensor selection and data reconstruction of the hybrid protocol.

3.1 Network Model of Hybrid Protocol

The network model of the proposed approach is depicted in Fig. 1. In this, the heterogeneous sensor nodes are randomly deployed at deterministic locations on a two-dimensional grid. There are two types of nodes deployed viz., sensor nodes and relay nodes. Sensor nodes are capable of collecting heterogeneous data, while relay nodes gather all data received from the sensors. The relay node interacts directly with the sensor nodes, servers, and datacenters in the IoT system. The servers collect information from the relay nodes and datacenter is responsible for data storage.

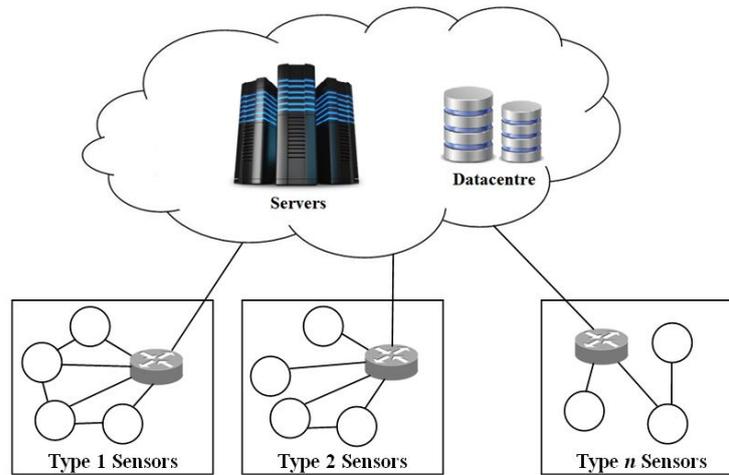


Fig. 1. Network Model for Hybrid Protocol

The proposed hybrid protocol works in two stages, Sensor Selection (SS) and Data Reconstruction (DR). In the SS stage, required data from different sensors are collected and in DR phase, the same data is used to reduce the energy consumption and improve the data quality. Both SS stage and DR stage works parallel for improving the data quality and reducing energy consumption in IoT sensors. In the DR stage, a belief is computed on each sensor nodes based on the history and captures the missing data based on the belief accumulated. The detailed design of the hybrid protocol is shown in Fig. 2.

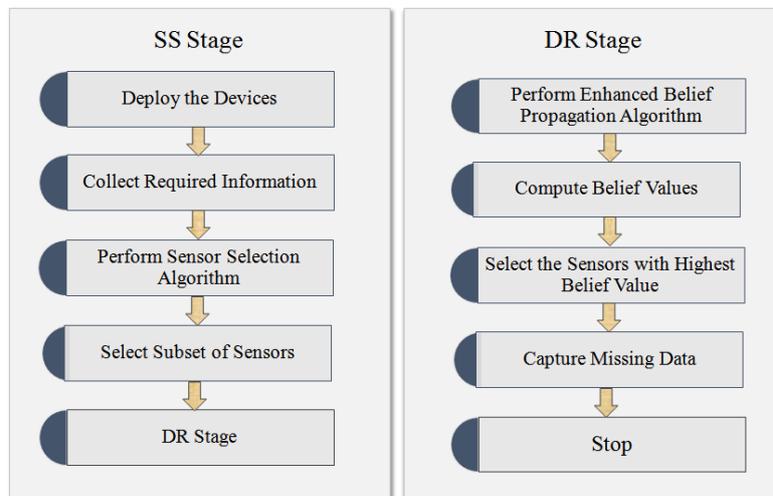


Fig. 2. Structure of the Hybrid Protocol

3.2 Sensor Selection (SS) Stage

Sensor nodes always communicate with the base stations (BS) according to the commands generated by the BS. In the SS stage, protocol estimates the link quality of the shortest path between sensor nodes and the base station, which is running on BS and can be used for sensor selection. There are two methods used for the sensor selection process. Dijkstra's algorithm is used for estimating the shortest path and ACO Inspired Variable Path algorithm is used for sensor selection.

Dijkstra's algorithm [23] is one of the best algorithms for calculating the shortest path from source to destination and is used to select the subset of sensor nodes which are active during the second stage. It generates the shortest path tree with the source node as root. It maintains two arrays, the first array contains sensor nodes which are included in the shortest path tree and second array stores sensor nodes which are not included in the shortest path tree. The algorithm takes a set of sensor nodes and link quality between these sensor nodes as inputs. At each iteration, the algorithm finds a sensor node with maximum link quality from a source node. It then creates a maximum link quality path set, S which keeps track of sensor nodes. The sensor nodes are included in the maximum link quality paths, i.e., link quality of these nodes from the source is calculated and stored. Initially the algorithm assumes the link quality of each node as infinity and link quality value from the source node to each node is assigned as zero. In the next iteration, the link quality is estimated by sending probe packet and selects the sensor node j which is not in S and has maximum link quality. Then update link quality value of all neighbouring sensor nodes of j . For every neighbouring sensor node n , if the sum of the link quality value of j (from source) and weight of edge j - n , is greater than the link quality value of n , then algorithm updates link quality of n . The formal algorithm for sensor selection is illustrated in Fig. 3. The algorithm initially set the link quality (link_quality) of source to vertex to 0 and other nodes except visited node is set to infinity. When there is no path available in the queue, the algorithms selects a node Q with a maximum link quality and add this node to the list of visited nodes. The procedure is repeated for all the neighbour nodes (n) in the link. Once all the nodes in the selected nodes are completed, the algorithm repeats the same process with another link with high link quality.

```

1. linkquality[s] ← 0          //(link quality to source vertex is zero)
2. for all  $n \in N - \{s\}$ 
3. do linkquality[n] ← ∞ // (set all other link quality to infinity)
4.  $S \leftarrow \phi$            //(S, the set of visited nodes is initially empty)
5.  $Q \leftarrow N$            //(Q, the queue initially contains all the nodes)
6. while  $Q \neq \phi$ 
7. do  $j \leftarrow \text{maxlinkquality}(Q, \text{linkquality})$  //(select the element of Q with max. link
   quality)
8.  $S \leftarrow S \cup \{j\}$  // (add j to list of visited nodes)
9. for all  $n \in \text{neighbours}[j]$ 
10. do if  $\text{linkquality}[n] < \text{linkquality}[j] + w(j, n)$  //(if new high link quality path
   found)
11. then  $\text{linkquality}[n] \leftarrow \text{linkquality}[j] + w(j, n)$  //(set new value)
12. return linkquality

```

Fig. 3. Dijkstra's Sensor Selection Algorithm

The second method for the sensor selection is ACO Inspired Variable Path Sensor Selection. The working of the ACO algorithm to solve an optimization problem is shown in the Fig. 4. The ants leave a certain amount of pheromone in its trail while travelling from its nest to the food source. While returning to the nest, ants track the same trail marked by the pheromone and deposit the pheromone on its path on their return journey. The follower ants choose a path which has a higher concentration of pheromone deposit. In the ACO, pheromone acts for exchanging the information between ants about the quality of the food source. If more number of ants follows a path, the concentration of pheromone will be high, i.e., the food source is good enough and that path leads to an optimum food source with maximum quality. This information also helps to find a number of alternative paths and shortest path from the nest to the food source.

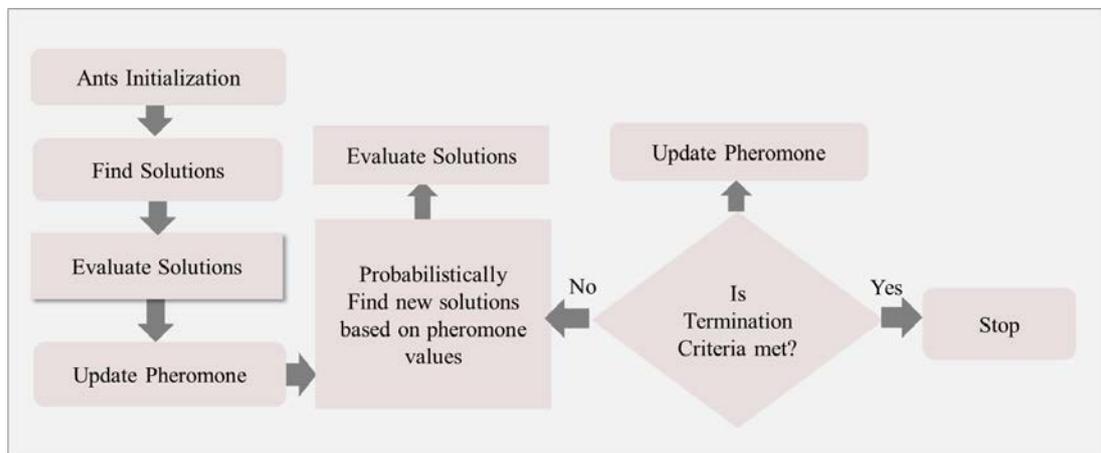


Fig. 4. Working Model of an ACO Algorithm

ACO algorithm is an extended version of ACO which is used to find the variable path selection dynamically from a source to destination in the hybrid protocol. In this algorithm, a forward ant is placed in every network node to find a path to reach the destination. The ants move and select the next node based on the probabilistic transition function d . The probability transition function of an ant which will select a sensor node j from sensor node i is

$$d(i, j) = \frac{[\eta_{ij}]^\beta}{[\eta_{is}]^\beta} \quad (2)$$

Where η_{ij} is the visibility of sensor node j from node i and η_{is} is the visibility of the sensor node i from source node s . Equation 3 and 4 gives the representation of η_{ij} and η_{is}

$$[\eta_{ij}] = 1/l_{ij} \quad (3)$$

$$[\eta_{is}] = 1/l_{is} \quad (4)$$

Where β is the weighting factor to regulate the influence of η_{ij} and η_{is} and its values is in the range of $[0, 1]$, l_{ij} is the link quality between sensor nodes j and i , l_{is} is the total link quality between sensor node j to node s .

The algorithm generates different paths from source to the destination by using the probabilistic transition function in equation (2). The detailed algorithm is given in Fig. 5. The function $getpath[s, d]$ maintains the set of different paths from the source node to a destination node. It selects only active nodes in each path. For finding different paths, the algorithm first checks whether the current node is the destination node or not. If it is a destination node, then it saves the path in the $getpath[s, d]$ and updates the routing table. If it is false, the algorithm

selects the next sensor node based on d value. If the d values of all next sensor nodes are same then set all these nodes as current node, else select the node with maximum d value. This process is repeated until the destination node is reached. As a result, different paths are obtained and stored in the set $paths[s, d]$. The algorithm ends with sorting these paths in the increasing order of their cost of the path. The estimated path is used in the second stage for reconstructing the missing data.

```

1.  getpaths(s, d)           //get the different path from source s to destination d
2.  if (c==d)               //if (current node=destination node) then save the path
3.  update the routing table
4.  Else                     //else get the next node with the probabilistic transition value
5.    For all paths from c to  $N_i$  //  $N_i$  all next available nodes
6.     $Prob[N_j] \leftarrow ProbTrans=[c, N_j]$  //get all probabilistic transition value, all available
    nodes from current node
7.    For all  $Prob[N_j]$ 
8.    If ( $Prob[N_j] == Prob[N_{ik}]; j \neq k$  //if (nodes have same probability
    transition values)
9.    select_route(c,  $N_i$ ) //select all the nodes of same value
10. update_route(c,  $N_i$ ) // update the routing table
11. set  $c=N_i$  //send to next node
12. get path(c, d) //gets the path
13. Else //else if (node available)
14. get  $\max\{prob[N_i]\}$ 
15. select_route(c,  $N_i$ )
16. update_route(c,  $N_i$ ) //update the routing table
17. set  $c=N_i$  //send it to next node
18. get path(c, d) //repeat the steps until it reach the destination node
19. Sort paths(s, d) in ascending order //sort the variable paths in ascending order
20. Return paths(s, d)

```

Fig. 5. ACO Inspired Variable Path Sensor Selection Algorithm

3.3 Data Reconstruction (DS) Stage

After the SS stage, only a subset of active sensor nodes is turned ON and other nodes are kept in an inactive mode. As a result, the total energy consumption of the network is significantly reduced. Due to these inactive nodes some data may not be available. DR stage reconstruct the missing data by using Enhanced Belief Propagation (EBP) algorithm. In this algorithm, a belief is computed with respect to the history of data sensing by each node [24]. According to the computed belief, sensor node passes a message to the neighbouring nodes. At each stage of the algorithm the decision to propagate the messages is based on the belief computed, so that there is pronounced improvement in the energy consumption. Fig. 6 shows an example of a message passing from y_1 to y_2 . In the figure, the sensor node y_1 waits for the arrival of messages from nodes P, Q, R, S, and T before sending message to y_2 . The message is transmitted similar to the Bayesian network [25].

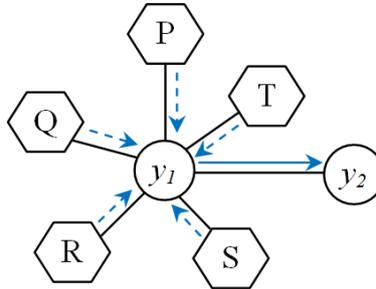


Fig. 6. Representation of message passing from y_1 to y_2

The detailed working of the EBP algorithm is shown in **Fig. 7**. The algorithm uses a BP function to reconstruct the missing data. Different variable paths and weight of each path obtained from the sensor selection are used as input to the algorithm. Initially user enters the sensor nodes to be activated. Each sensor node computes the belief values and returns a set BN_i which consists of all the calculated belief values corresponding to the particular required node. The sensor node which has minimum belief value ($\min [BN_i]$) is selected. Finally the least weighted path with minimum belief is selected and keep the sensor nodes in the path is in active mode. In this way the missing sensor data is recomputed to improve data quality as well as reduce the energy consumption in the IoT sensor network.

```

1. Belief Propagation [Paths from s to d, weight of each path]
   // Different variable paths and weight of each path obtained from the sensor
   selection are used as the input.
2. do
3.   begin
4.    $n_{rj} \leftarrow$  Required Node; Here  $r=1$  to  $n$ ;  $n>0$ ,  $j=1$  to  $m$ ;  $m>0$ 
   // the needed sensor node [ $n_{rj}$ ] readings are the missing sensor data where  $r$  is the node
   number, e.g., 6th node, 7th node etc.  $j$  is the context value: Example: temperature,
   voltage, pressure etc
5. for each available nodes  $N_i$ 
6.    $BN_i = \sqrt{(N_{ij} - n_{rj})^2}$  //performs belief function and computes the belief
   values of each sensor nodes. Set  $BN_i$  which consists of all calculated belief values
   corresponding to the needed node.
7. get min [ $BN_i$ ]
8. get  $N_i$  (get min [ $BN_i$ ])
9. get  $W[s, d]$  //Where  $W[s, N_i, d] = \min(s, N_i, d)$ 
   // From the calculated belief values of different sensor nodes, takes the sensor node
   which have minimum belief value ( $\min [BN_i]$ ). Instead of taking the sensor node
   having maximum belief value the algorithm takes the minimum belief valued
   sensor node because the belief value computation is done in inverse manner//
10. end begin

```

Fig. 7. Enhanced Belief Propagation Algorithm

The SS stage and DS stage of the hybrid protocol is illustrated with the following example. Consider a sensor network with 20 sensor nodes as given in **Fig. 8**. Different links are available through these nodes to collect data from the user-specified sensor.

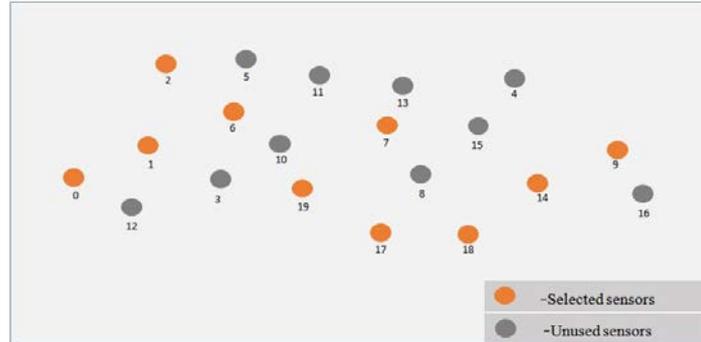


Fig. 8. IoT sensor Network with 20 nodes

The possible sensor selection paths and its weights during the SS stage using ACO inspired variable path sensor selection are shown in the [Table 1](#).

Table 1. Sensor Selection Path and Weight

Path	Weight
0- 1- 2- 6- 11- 13- 14- 18- 17- 19	406
0- 1- 2- 6- 11- 13- 15- 14- 18- 17- 19	527
0- 1- 2- 4- 6- 7- 9- 14- 18- 17- 19	475
0- 1- 2- 4- 6- 11- 13- 14- 18- 17- 19	457
0- 1- 2- 4- 6- 11- 13- 15- 14- 18- 17- 19	578
0- 1- 2- 4- 9- 14- 18- 17- 19	399
0- 1- 2- 9- 14- 18- 17- 19	359
0- 1- 6- 7- 9- 14- 18- 17- 19	348
0- 1- 6- 11- 13- 14- 18- 17- 19	330
0- 1- 6- 11- 13- 15- 14- 18- 17- 19	451
0- 1- 6- 11- 13- 2- 4- 9- 14- 18- 17- 19	559
0- 1- 6- 11- 13- 2- 9- 14- 18- 17- 19	519

Suppose the reading from 5th sensor node is required and data from that node is missing. In order to reconstruct the data from 5th node, the algorithm uses historical data from this node. For that, a belief is computed by using Euclidean distance comparison of current sensor readings of each node, such as temperature, humidity, light, voltage etc., with the past data. Then BN_i is calculated as:

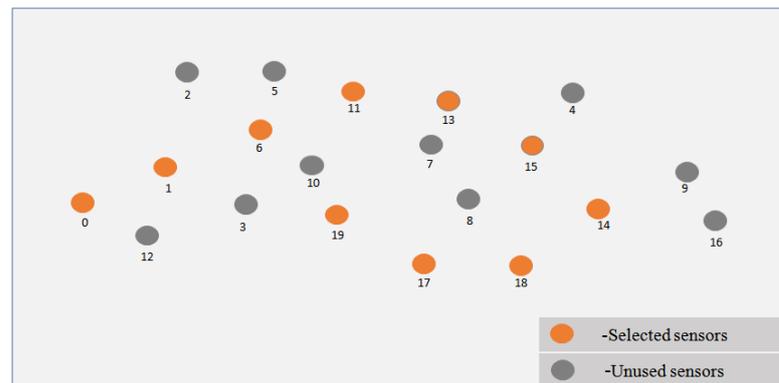
$$BN_i = \sqrt{((N_{1t} - N_{5t})^2 + (N_{1h} - N_{5h})^2 + (N_{1l} - N_{5l})^2 + (N_{1v} - N_{5v})^2)} \quad (5)$$

Where N_{1t} , N_{1h} , N_{1l} , and N_{1v} represent respectively temperature, humidity, light, and voltage readings from first node. Similarly, N_{5t} , N_{5h} , N_{5l} , and N_{5v} are the available past sensor readings available from the 5th node. Similarly, the system calculates belief propagation weight for all selected sensors. The belief propagation weight for each selected node in the example is listed in [Table 2](#).

Table 2. Belief Propagation Weight

Node	BP Weight
1	1.4429425005869
2	2.1917309534019
4	1.7340053898705
6	2.7327039612077
7	1.4460751190723
9	1.3928700655840
11	5.3872996964713
13	5.0418438802089
14	4.8184143034820
17	2.4712567571986
18	1.0515410179827
19	1.9482419654909
15	0.85920338808689

In the next step the sensor node with minimum belief value ($\min [BN_i]$) is selected as the candidate node. In this example 15th node is selected. The optimal sensors in the example network are in the path 0- 1- 6- 11- 13- 15- 14- 18- 17- 19 with BP weight value 451. The optimal nodes are shown in orange colour in the Fig. 9.

**Fig. 9.** Optimal Nodes are selected according to the BP weight computed

4. Results and Discussions

The proposed system collected real sensor data set from 54 sensors installed in the Intel Berkeley Research Laboratory [8]. The structure of the deployment of sensor is shown in Fig. 10. The sensor nodes are represented in hexagon shape and the number written on the shape is sensor id. Sensor data are collected by means of TinyDB in-network query handling system built on the TinyOS platform. The dataset consists of sensor readings and aggregate data over time. Aggregate data represents the value of the link quality aggregated between sensor nodes. It consists of sender id, receiver id, and the probability of receiving a message successfully from sender to receiver.

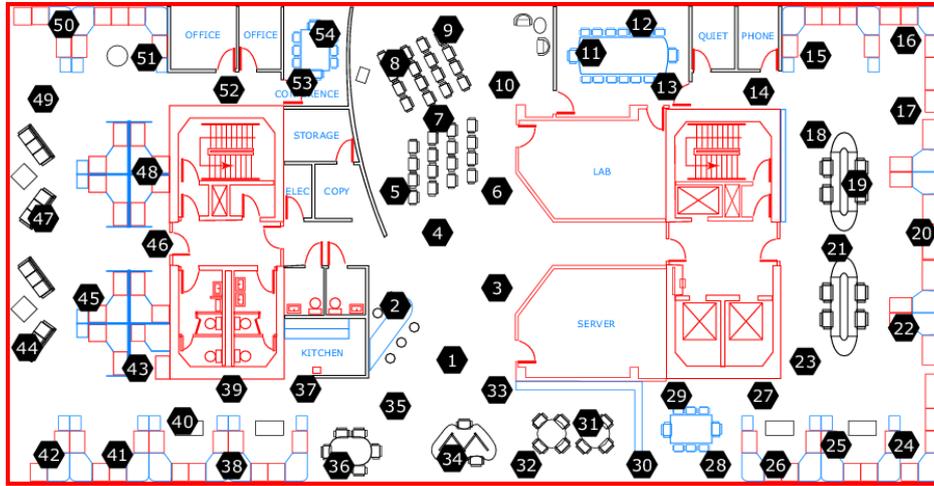


Fig. 10. Map of Intel Berkeley Research Laboratory

Fig. 11 represent the network map based on connectivity information obtained from the dataset from selected 20 sensors. If there is no direct connectivity between nodes, then the value in the respective cell is 1000 in the network map. Other values in the network map represent the measured inverse link quality values between different sensor nodes. The link quality measurement in an inverse manner is described in the EBP algorithm. The diagonal cell values are represented as 0. Different types of sensors are used to collect required heterogeneous data. The sensor nodes are responsible to collect various time stamped topology statistics, along with temperature, humidity, light and voltage values once every 30 seconds, during a period of 36 days. The base station collects the data from the selected sensors. The Dijkstra’s sensor selection algorithm and ACO Inspired variable path sensor selection algorithm are used for sensor selection.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	0	21	33	32	1000	1000	1000	1000	53	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
1	21	0	75	22	1000	1000	54	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
2	33	75	0	1000	40	1000	59	1000	1000	19	1000	1000	1000	96	1000	1000	1000	1000	1000	1000	1000
3	32	22	1000	0	1000	1000	83	1000	1000	1000	1000	92	1000	1000	1000	1000	1000	1000	1000	1000	1000
4	1000	1000	40	1000	0	1000	70	1000	85	79	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
5	1000	1000	1000	1000	1000	0	1000	1000	1000	1000	1000	68	99	1000	1000	1000	1000	1000	1000	1000	1000
6	1000	54	59	1000	70	1000	0	39	78	1000	46	58	1000	1000	1000	1000	1000	1000	1000	1000	1000
7	1000	1000	1000	83	1000	1000	39	0	59	46	52	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
8	53	1000	1000	1000	85	1000	78	59	0	51	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
9	1000	1000	19	1000	79	1000	46	51	0	1000	1000	82	1000	60	1000	1000	1000	1000	1000	1000	1000
10	1000	1000	1000	1000	1000	1000	46	52	1000	1000	0	35	82	55	70	1000	1000	1000	1000	1000	1000
11	1000	1000	1000	1000	1000	68	58	1000	1000	1000	35	0	1000	23	25	1000	89	1000	97	1000	1000
12	1000	1000	1000	92	1000	99	1000	1000	1000	82	82	1000	0	19	27	1000	1000	1000	1000	1000	1000
13	1000	1000	96	1000	1000	1000	1000	1000	1000	1000	55	23	19	0	46	96	1000	1000	1000	1000	1000
14	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	60	70	25	27	46	0	71	60	1000	63	1000
15	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	96	71	0	74	1000	1000	1000	1000
16	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	89	1000	1000	60	74	0	1000	32
17	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	22	50
18	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	97	1000	1000	63	1000	32	22	0	1000
19	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	49	50	1000	0
20	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

Fig. 11. Network Map based Representation on Aggregate connectivity data

The performance of the new system is evaluated in terms of overall energy consumption, data utility, accuracy, and data reconstruction error. To evaluate the energy utilization of the system, considered 14-mA as transmission cost for a sensor as reported in Mica2Dot mote, which is used in the deployment of Intel Berkeley Research laboratory.

4.1 Overall Energy Consumption

The low value of energy consumption (Fig. 12) highlights the relative improvement of the proposed Dijkstra's sensor selection and ACO Inspired variable path sensor selection as against Constraint Programming (CP) and Random sensor selection algorithms with varying number of sensor nodes. Since the selected sensor nodes are only involved in the data sensing phase in the proposed algorithms, the energy consumption is substantially reduced, as shown in figure. It is interesting to observe that the proposed ACO inspired variable path sensor selection has less energy consumption than other algorithms and energy consumption increases linearly with respect to number of nodes.

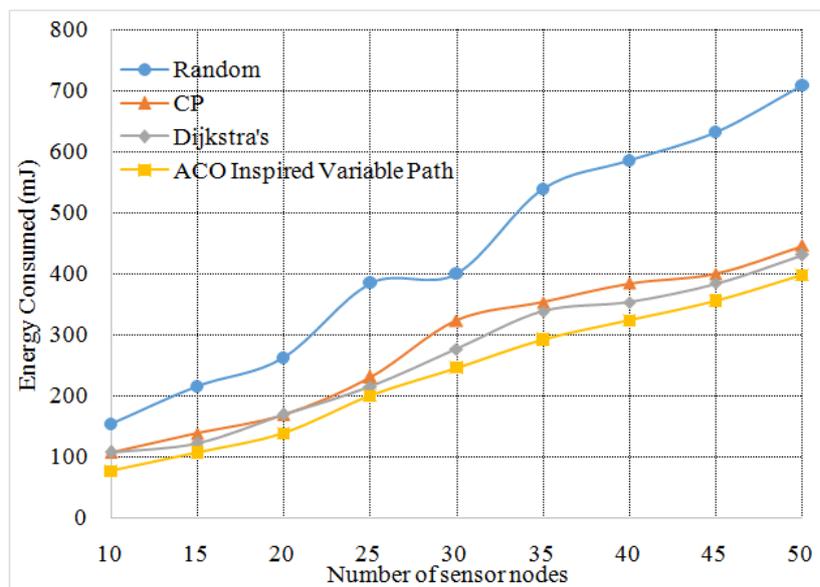


Fig. 12. Energy consumption of Dijkstra's and proposed ACO inspired variable path sensor selection is relatively low, compared to Random and CP algorithms, when the number of node increases

4.2 Data Utility

Data utility of a node depends on link quality and number of nodes present in a path. The data utility rate is increased with respect to the number of paths a node participates. The data utility is measured using the equation (6).

$$d[s] = \frac{y[s].q[s]}{n[s]} \quad (6)$$

where 's' represents sensor nodes, $d[s]$ denotes data utility of the sensors in the network and $y[s]$ is a Boolean variable. If a sensor is selected then y value is 1 otherwise 0. The $q[s]$ is the total link quality achieved from the selected sensor nodes to a base station in a path and $n[s]$ is the total number of nodes used for data collection.

Fig. 13. Shows the comparison of the performance of Dijkstra's and ACO Enhanced BP (ACO-EBP) with CP in terms of packet data utility for different number of nodes. Referring to **Fig. 13**, there is a pronounced improvement in data utility in ACO-EBP compared to other algorithms.

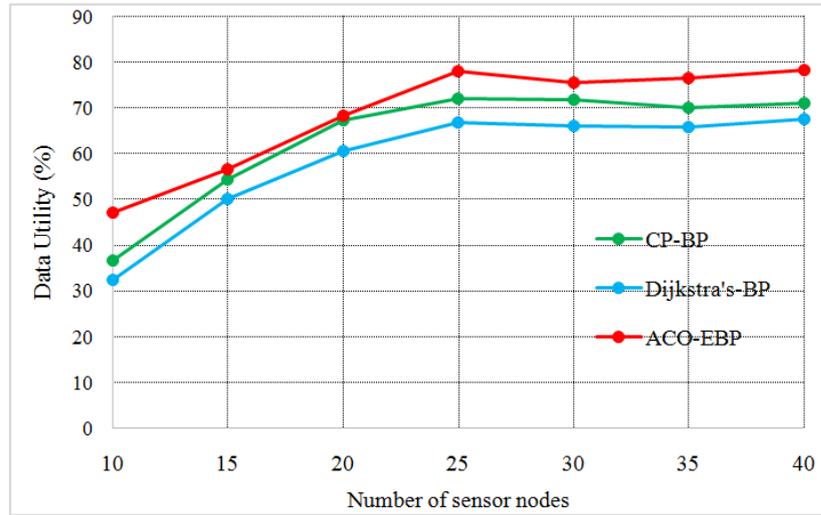


Fig. 13. Comparison of Data Utility; Data Utility is highest for ACO Inspired EBP with large number of nodes.

4.3 Data Reconstruction Error

The mean square error (MSE) of the reconstructed data is computed using BP algorithm and results are represented in **Fig. 14**. The data reconstruction error value is low in the case of ACO-EBP and Dijkstra's-BP algorithms with the increase in the number of nodes, as against the abrupt increase in CP and Random algorithms.

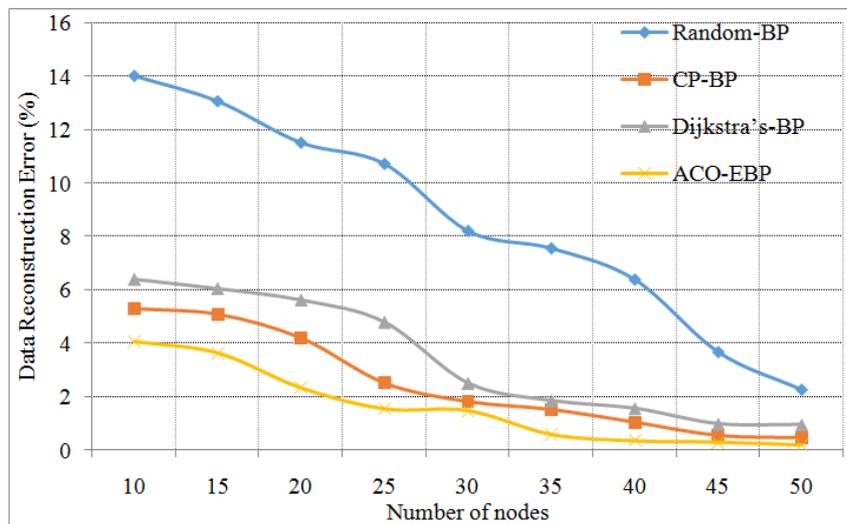


Fig. 14. MSE is low for ACO-EBP compared to other three protocols, when the number of nodes increases.

4.4 Data Prediction Error

Comparison of data prediction error among ACO inspired EBP, Dijkstra’s BP with CS-BP and Random BP is given in Fig. 15. CS-BP approach uses the matrix completion techniques for predicting the missing data in the Environmental Matrix (EM). Even though CS-BP is an efficient method for missing data prediction, the major limitation is the data must be collected from the different sensors regularly. The low value of the data prediction error indicates the remarkable confidence in ACO inspired BP.

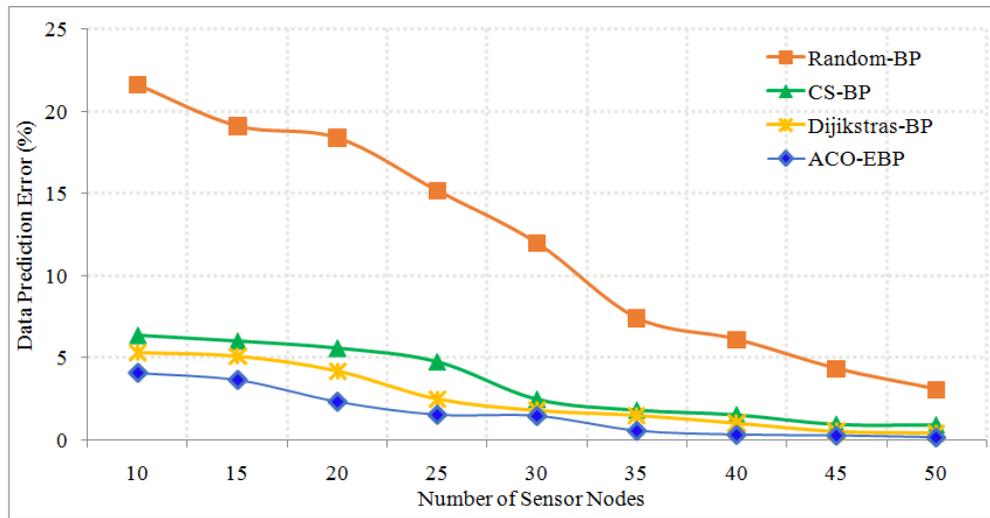


Fig. 15. Data prediction error comparison –ACO-EBP and Compressive Sensing

4.5 Inference Accuracy Analysis of Enhanced BP Algorithm

In order to reconstruct the lost or incorrect data from a sensor, the proposed system uses Enhanced Belief Propagation (EBP) algorithm. The missing data is inferred from the existing data of the nearby sensors. Inference accuracy is an important parameter to test the efficiency of the proposed method. Fig. 16 shows the comparison of inference accuracy between EBP algorithm and the belief propagation. Relatively high value of inference accuracy shows improvement in the performance of EBP algorithm.

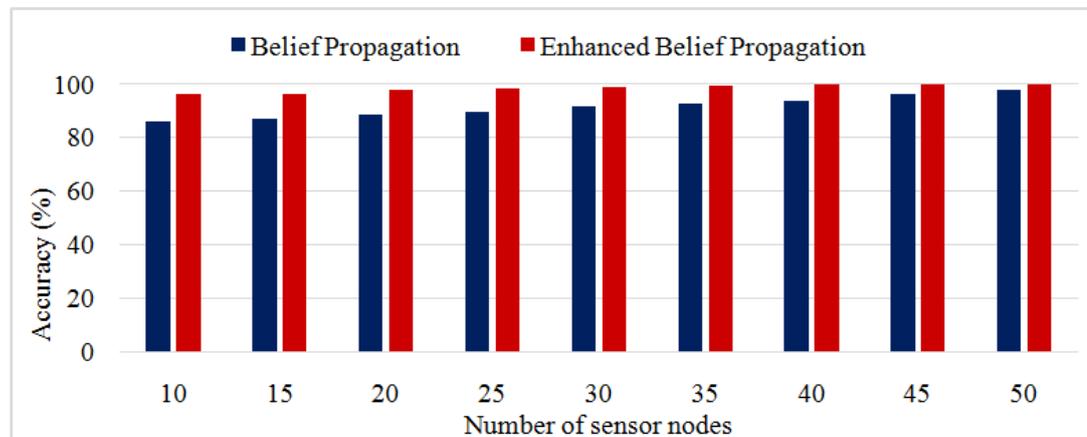


Fig. 16. Inference accuracy of Enhanced BP &BP as against the number of sensor nodes

5. Conclusion

The paper has established and evaluated a hybrid protocol which consists of two algorithms for improving the energy efficiency and data quality in IoT sensor networks. The proposed hybrid protocol selects an optimal set of active sensor nodes that maximizes the data quality and minimizes energy consumption in IoT networks, thereby reducing the storage space in the cloud platform. The first method is formulated using Dijkstra's sensor selection algorithm and the second method using ACO inspired variable path sensor selection algorithm, which further strengthens the reconstruction of missing data by computing belief using enhanced belief propagation algorithm. Belief is computed on the basis of the past data from the nearby sensor nodes, and thus resulting in the most accurate data.

The performance of the proposed algorithms are evaluated in respect of energy consumption, data utility, data reconstruction error, data prediction error and inference accuracy. The results are compared against the existing Random sensor selection method, constrained programming and Compressive sensing method. The Dijkstra's algorithm shows 37.5% improvement and ACO-BP method shows 45.7% than random method in energy consumption. ACO is more efficient than Dijkstra's algorithm in energy efficiency as well as time complexity. As the number of nodes increases, mean square error is reduced to 0.16% in ACO-EBP as compared to other existing algorithms. The experimental study shows the pronounced improvement in the performance parameters. The energy consumption is significantly reduced in the suggested algorithms since the selected sensor nodes are only participating in the data sensing phase. It's worth noting that the suggested ACO-inspired variable path sensor selection consumes less energy than existing algorithms, and that energy consumption increases linearly as the number of nodes increases.

The main task of this BP-based algorithm is to update messages between pairs of nodes. The worst case computational complexity of proposed BP algorithm is $O(N^2)$. But on closer analysis, the method considers fewer nodes for the optimal transfer, thus the computational load is lesser than worst case analysis of Dijkstra's algorithm, i.e., $O(N^2)$. For constraint programming, the complexity is $O(d^3e)$, which depends on the upper bound of the domain size (d) and the number of constraints e and it is much higher than the proposed method.

References

- [1] Zheng, D.Simplot-Ryl, C. Bisdikian, and H. Mouftah, "The Internet of Things," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 30–31, Nov. 2011. [Article \(CrossRef Link\)](#)
- [2] A.B. Pawar, Shashikant Ghumbre, "A survey on IoT applications, security challenges and counter measures," in *Proc. of International Conference on Computing, Analytics and Security Trends (CAST)*, pp. 294 – 299, 2016. [Article \(CrossRef Link\)](#)
- [3] N Bari, Ganapathy Mani, Simon Berkovich, "Internet of Things as a Methodological Concept," in *Proc. of IEEE International Conference on Computing for Geospatial Research and Application*, pp. 48 - 55, 2013. [Article \(CrossRef Link\)](#)
- [4] L. Kong, D. Jiang, M.Y. Wu, "Optimizing the spatio-temporal distribution of cyber-physical systems for environment abstraction," in *Proc. of IEEE 30th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, pp. 179–188, Jun. 2010. [Article \(CrossRef Link\)](#)
- [5] L.Palopoli, R. Passerone, T.Rizano, "Scalable offline optimization of industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 328–339, 2011. [Article \(CrossRef Link\)](#)
- [6] Remesh Babu K.R., Philip Samuel, "Enhanced Bee Colony Algorithm for Efficient Load Balancing and Scheduling in Cloud," *Advances in Intelligent Systems and Computing*, Vol. 424, pp. 67-78, 2015. [Article \(CrossRef Link\)](#)

- [7] Shufen Liu, Huang Leng, Lu Han, "Pheromone Model Selection in Ant Colony Optimization for the Travelling Salesman Problem," *Chinese Journal of Electronics*, vol. 26, no. 2, pp. 223 – 229, 2017. [Article \(CrossRef Link\)](#)
- [8] S. Madden, "Intel Lab data 2004," [Online]. Available: <http://www.select.cs.cmu.edu/data/labapp3/index.html>.
- [9] Sarwesh P, N Shekar V Shet, Chandrasekaran K, "Energy Efficient Network Architecture for IoT Applications," in *Proc. of IEEE International Conference on Green Computing and Internet of Things*, pp. 784-789, 2015. [Article \(CrossRef Link\)](#)
- [10] Evangelos Zimos, Joao F. C. Mota, Miguel R. D. Rodrigues, and Nikos Deligiannis, "Internet-of-Things Data Aggregation Using Compressed Sensing with Side Information," in *Proc. of International Conference on Telecommunications*, pp.1-5, June 2016. [Article \(CrossRef Link\)](#)
- [11] Shancang Li, Li Da Xu, Xinheng Wang, "Compressed Sensing Signal and Data Acquisition in Wireless Sensor Networks and Internet of Things," *IEEE Transactions On Industrial Informatics*, Vol. 9, No. 4, pp. 2177 - 2186, November 2013. [Article \(CrossRef Link\)](#)
- [12] L. Kong, M. Xia, Xiao-Yang Liu, Min-You Wu, Xue Liu, "Data loss and reconstruction in sensor networks," *IEEE INFOCOM*, pp. 1654–1662, 2013. [Article \(CrossRef Link\)](#)
- [13] Dusit Niyato, Dong In Kim, Ping Wang, and Lingyang Song, "A Novel Caching Mechanism for Internet of Things (IoT) Sensing Service with Energy Harvesting," in *Proc. of IEEE International conference on Communications*, pp.1-6, 2016. [Article \(CrossRef Link\)](#)
- [14] Farshid Hassani Bijarbooneh, Wei Du, Edith C.-H. Ngai, Xiaoming Fu, J. Liu, "Cloud-Assisted Data Fusion and Sensor Selection for Internet of Things," *IEEE Internet of Things Journal*, Vol. 3, No. 3, pp. 257-268, June 2016. [Article \(CrossRef Link\)](#)
- [15] Kaustubh Dhondge, Rajeev Shorey, Jeffrey Tew, "HOLA: Heuristic and Opportunistic Link Selection Algorithm for Energy Efficiency in Industrial Internet of Things (IIoT) Systems," in *Proc. of IEEE 8th International Conference on Communications Systems and Networks*, pp.1-6, 2016. [Article \(CrossRef Link\)](#)
- [16] K. Nair, J. Kulkarni, M. Warde, Z. Dave, V. Rawalgaonkar, Ganesh Gore, J. Joshi, "Optimizing Power Consumption in IoT based Wireless Sensor Networks using Bluetooth Low Energy," in *Proc. of International Conference on Green Computing and Internet of Things*, pp. 589 – 593, 2015. [Article \(CrossRef Link\)](#)
- [17] Yihong Zhang, Claudia Szabo, Quan Z. Sheng, "An Estimation Maximization Based Approach for Finding Reliable Sensors in Environmental Sensing," in *Proc. of IEEE International conference on parallel and distributed systems*, pp.190-197, Mar. 2015. [Article \(CrossRef Link\)](#)
- [18] J. Chou, D. Petrovic, K. Ramachandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," *IEEE INFOCOM*, vol. 2, pp. 1054–1062, 2003. [Article \(CrossRef Link\)](#)
- [19] Shang W, Yu Y, Droms R., "Challenges in IoT networking via TCP/IP architecture," NDN Technical Report NDN-0038, 2016. [Online]. Available: <http://named-data.net/techreports.html>.
- [20] Zaineb T. Al-Azez, A.Q. Lawey, Taisir E.H. El-Gorashi, J.M.H. Elmirghani, "Virtualization Framework for Energy Efficient IoT Networks," in *Proc. of IEEE 4th International Conference on Cloud Networking*, pp. 74 - 77, March 2015. [Article \(CrossRef Link\)](#)
- [21] Hasan Ali Khattak, Haleem Farman, Bilal Jan, and Ikram Ud Din, "Toward Integrating Vehicular Clouds with IoT for Smart City Services," *IEEE Network*, vol. 33, no. 2, pp.65-71, 2019. [Article \(CrossRef Link\)](#)
- [22] Do, Dinh-Thuan, Minh-Sang Van Nguyen, Thi-Anh Hoang, and Miroslav Voznak, "NOMA-assisted multiple access scheme for IoT deployment: Relay selection model and secrecy performance improvement," *Sensors*, vol. 19, no. 3, p. 736, 2019. [Article \(CrossRef Link\)](#)
- [23] B. Li, N. Wu and Y. Wu, "Distributed Verification of Belief Precisions Convergence in Gaussian Belief Propagation," in *Proc. of ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 9115-9119, 2020. [Article \(CrossRef Link\)](#).

- [24] J. S. Yedidia, W. T. Freeman, Y. Weiss, "Understanding belief propagation and its generalizations," in Exploring artificial intelligence in the new millennium, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 239 – 269. [Online]. Available: <https://www.merl.com/publications/docs/TR2001-22.pdf>
- [25] Saqaeyan, Sasan, and Hossein Amirkhani, "Anomaly detection in smart homes using bayesian networks," *KSI Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 4, 1796-1816, 2020. [Article \(CrossRef Link\)](#).
- [26] G. Anastasi, A. Falchi, A. Passarella, M. Conti, E. Gregori, "Performance measurements of motes sensor networks," in *Proc. of ACM International symposium on Modeling, analysis and simulation of wireless and mobile systems*, pp. 174–181, 2004. [Article \(CrossRef Link\)](#)



K R Remesh Babu received PhD Degree from Cochin University of Science and Technology (CUSAT) in 2019. He also holds bachelor degree in Mathematics, Information Technology and master degree in Computer Science. Currently he is working as Associate Professor and Head in the department of Information Technology, Government Engineering College, Palakkad, India. He is author of more than 50 publications in the leading research journals and conferences. He is interested in Distributed and Cloud Computing, Internet of Things, Wireless Sensor Networks, Machine Learning, and Big Data Analytics.



Preetha K G has completed her PhD in Mobile Ad hoc Networks from Cochin University of Science and Technology in 2018. She has completed her M Tech and B Tech Degree in Computer Science and Engineering. She has been associated with Rajagiri School of Engineering & Technology since 2004 and is now working as an Associate Professor in the Department of Computer Science & Engineering. She has around 20 years of academic experience. Her research interests include Mobile Computing, Wireless Networks, Ad-hoc Networks, Data Analytics etc. She has around 25 national and international conference papers (IEEE, Springer, ACM) and international journal papers (SCI/Scopus Indexed). She is a member of ISTE and CSI. She is the recipient of the Research Excellence award in 2018. She is also a reviewer of international conferences and journals and a registered research guide in APJ Abdul Kalam Technological University. She has also a National Patent published to her credit.



Rinil K R earned his M. Tech (Information Technology) degree from APJ Abdul Kalam Technological University, Kerala, India. He is currently working as Assistant Professor in Computer Science and Engineering department at College of Engineering Thalassery, Kerala, India. His current research interests include Cloud Computing and Big data, Internet of Things, Data Mining, Networking and Cyber-Physical Systems.



Saritha S earned her PhD degree in Computer Science and Engineering from Cochin University of Science and Technology, India. She is currently working as an Associate Professor in the Department of Computer Science, Rajagiri School of Engineering & Technology, Kochi, Kerala, India. She has around 18 years' of teaching experience to her credit. She also possess several years of industrial experience in IT industry. Her research interest includes data mining, machine learning, cyber physical systems, computer networking and cloud computing. She has also authored several publications in international conferences and journals. She is a reviewer of well known international journals.