Indian Journal of Science and Technology, *Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102104, October 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Application of Data Fusion and Sensor Head Algorithm for Data Extraction from Wireless Sensor Networks

Bhavya Sharma*, Adit A. Pillai and P. Saranya

Computer Science and Engineering department, SRM University, Kattankulathur, Kanchipuram - 603 203, Tamil Nadu, India; bhavya.rayasthaila@gmail.com, aditapillai@gmail.com, saranyatkp@gmail.com

Abstract

Objectives: The primary objective of the paper is to propose an optimized data extraction technique to obtain data from the Wireless Sensor Networks (WSNs)using 'Sensor Head Algorithm' and Data Fusion. **Methods/Statistical Analysis:** The proposed idea works in a three-layered architecture which includes the deployed sensors, the selected Sensor Head and the Static Data Sink. The Sensor Head Algorithm is used to select a sensor head that collects data from all sensors. The Data Fusion algorithm results in generation of unified data which is used to derive the output. Finally, the fused data is transmitted from the Sensor Head to the Static Data sink. **Findings:** A sample data was used to verify the working of the model using simulations. On the application of Sensor Head Algorithm, the sensors are assigned priorities on the basis of their residual energies. The sensor with highest residual energy becomes the Sensor Head. All the sensors transmit their data to the Sensor Head where the Data Fusion takes place later. Data Fusion initially removes redundancy in from the collected data making the processing simpler and saving the energy. The resultant data is then used to generate fused inference using Fuzzy Logic Controller (FLC). These inferences are recorded and later used to generate final output. The proposed idea ensures reduction of load on the system, simplifies the previously proposed methods, reduces latency in data collection and improves network lifetime. This data, finally transmitted to the static data sink, can be used for finding inferences or reaching a conclusion about the target area in the desired way. **Application/Improvements:** In the current paper, we have proposed an idea which we shall use to develop a fully functional model. Since, the proposed technique is concerned with data extraction, it can be applied to every situation where the Wireless Sensor Network is deployed.

Keywords: Data Fusion, Fuzzy Logic Controller, Residual Energy, Sensor Head Algorithm, Static Data Sink, Wireless Sensor Networks (WSNs)

1. Introduction

The Sensors have been used since many years for collecting data in various domains of applications such as the military, environment, health, home, improvised surveillance cameras[1] and other commercial domains. It is possible to expand this classification with more categories such as space exploration, chemical processing, and disaster relief[2]. Although various methods of data extraction using sensors have been proposed, there still is a scope of improvement in terms of various factors that affect data extraction. The current paper deals specifically with the Wireless Sensors deployed in a uniform region to collect physical parameters and proposes algorithms to collect and then fuse the data so as to produce a proper inference that can be used for desired purposes. The proposed model shall reduce latency in data collection, reduce the redundant data, reduce energy consumption and improve the network lifetime. Going by the definition, "A Sensor is a tiny device that is capable of gathering physical information like heat, light or motion of an object or environment. Sensors are deployed in an ad-hoc manner

**Author for correspondence*

in the area of interest to monitor events and gather data about the environment"[3]. Wireless Sensor Networks is the network of such autonomous wireless sensors distributed in a target area. The information collected by these sensors is sent to a static data sink where it can be used for various purposes. The proposed idea involves using algorithms that effectively improve the complete procedure from collecting the data at the sensors to sending all the information to the sink. The Sensor Head Algorithm is the first algorithm that selects a sensor from amongst the deployed sensors to receive data. It prioritizes all the sensors on the basis of residual energy and selects the sensors with maximum priority as the Sensor Head. The selected head has the maximum residual energy so as to support the incoming traffic. The next algorithm is of Data Fusion. 'Data Fusion is a process dealing with the association, correlation, and the combination of data and information from single and multiple sources to achieve refined position and identity estimates, complete and timely assessment of situations and threats, as well as their significance'. In the proposed idea, the Fusion takes place at the Sensor Head after the data has been collected. Now, in terms of Sensor Networks, Data Fusion can be explained as 'Techniques that combine data from multiple sensors, and related information from associated databases, to achieve improved accuracies and more specific inferences than could be achieved by the use of a single sensor alone'[4]. Basically, the aim of Data Fusion is to generate small set of information that can be used as inference from large data collected[5]. The use of Data Fusion firstly removes the redundancy in the collected dataset, hence preventing overloading and energy loss. It then uses Fuzzy Logic Controller to generate Inference Factor from the input data. This Inference Factor reading, taken over a period of time, is used to find the average of all the factors. This average is the final fused data in an integrated form that is sent to the data sink. The next section explains the theoretical working of the model proposed, followed by an explanation of Sensor Head algorithm and Data Fusion technology.

2. Explanation of Proposed Model

The proposed plan describes an architecture that includes the Sensor Layer, Sensor Head Layer, and Data Sink Layer. Figure 1 It is inspired by the framework explained in[6]. The bottom-most layer or the Sensor layer contains all the sensors distributed densely and arbitrarily over the target area. All the deployed sensors work on a specific energy level. The energy level will vary from sensor to sensor since generally there is no uniformity in placing the sensors. Hence, amongst the network of sensors, the energy is not balanced and is haphazard in nature causing the coverage to be affected. In order to ensure the balance of energy of sensors and the data being collected, a sensor needs to be selected to collect data from all the deployed sensors. This selection of Sensor Head is facilitated by the Sensor Head Algorithm. The selection of Sensor Head provises network lifetime and leads to reduction of energy usage[7]. As explained before, the Sensor Head is selected on the basis of residual energy priority. All the sensors have their individual residual energy, and the priority is defined by the formula given.

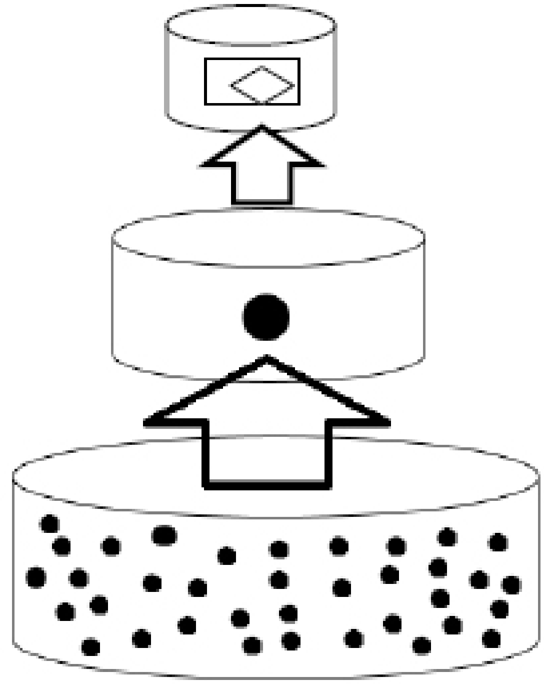


Figure 1. The Sensor Layer depicting sensors in clusters sending data to Sensor Head Layer which finally sends it to Static Data Sink.

$$\text{Sensor priority} = \frac{\text{Residual Energy of sensor}}{\text{Total Energy of the system}} \quad (1)$$

When the Sensor Head depletes its energy, the next sensor with the highest priority is selected as the Sensor Head and assumes all the on-going operations being carried out by previous Sensor Head. Upon selection of Sensor Head, a possibly large number of sensors transmit their information to the Head. Hence, there are high chances of collisions amongst the sensors when they try to approach the Sensor Head to transmit data. These collisions can be avoided using Time-division Multiple-Access (TDMA) based method and by synchronization of the clocks of the sensor nodes. The second virtual layer is called the Sensor Head layer which contains the Sensor Head. After the application of the Sensor Head Algorithm, the Sensor Head has been selected which collects the data from all the sensors and fuses it into a unified data using the fusion technology while removing the redundant data items. Figure 2 The third layer is the Data Sink layer that contains the Static data sink where all the fused information is finally collected. Since the data now being transmitted to the sink has been reduced, the cost of transmission in terms of energy is also reduced significantly. Thus, the data retrieved would be easy to study and process.

3. The Sensor Head Algorithm

As explained before, the Sensor Head Algorithm is responsible for selecting a Sensor Head from the deployed Sensors. Inspired by the work in, the algorithm's working primarily depends on the selection of the Sensor Head at a better cost. Since the residual energy of each sensor has been taken as the basis for assigned priorities, the sensor with the highest residual energy is assigned the first priority and is made the Sensor Head. In the case of a clash in priority, the Sensor Head is chosen randomly from the sensors with clashing energy values. The complete algorithm has been broken down into four phases, which have been explained.

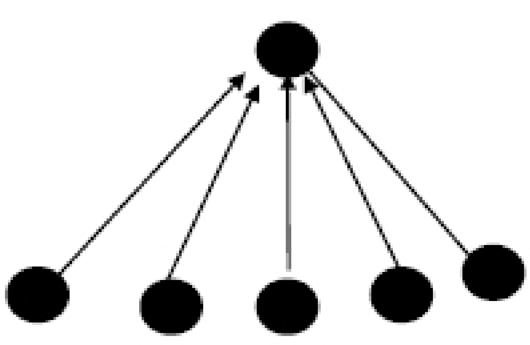


Figure 2. Aggregation of data from various sensors into one.

3.1 Phase I

The first phase of the algorithm sets up the sensors with their residual energy and calculates their initial priorities. Initially, all the sensors are marked with an "uncertain" status, which is finally changed to "member" or "head" depending on the fact that the sensor is a sensor head or a normal member.

Phase I
1: Find the set of neighbors(s_n) of a particular sensor (s);
2: **if** s_n is **NULL then**
3: Set s.status as sensor_head
4: Set sensor_head.id as s;
5: **else**
6: Set initial priority of s as ($\mathbf{E_r/E_t}$) /* *Residual energy / Total Energy* */
7: cluster_head = 0;
8: Set s.status as uncertain;
9: Collect s.peers;
10: Set initial priority of s as s.Init_priority + summation of priorities of all peers;
11: Set all peers as **NULL** and all cluster members as **NULL;**
12: Iterate till 0;

Pseudocode for Phase I

3.2 Phase II

In the second phase of the algorithm, the sensors update their status in an iterative manner. The total number of iterations is in turn governed by the node degree of each sensor and the question of making a sensor the cluster head depends on its initial priority. With each iteration, the sensors exchange their priorities through packet transfer and inter-sensor communication. Finally, the sensor with the highest initial priority is marked as the "head" and rest of the sensors are marked as "members". At the start of phase II, all the sensors are by default marked as "members";

```
Phase II

Loop
    Select sensor i;
    Send(i.Init_prio,Set(sensors)-i);
    Set=Receive(Set(sensors)-i);
Loop
    If(i.initPrio>Set)
    i.status = "head";
End loop
```

Pseudocode for Phase II

3.3 Phase III

The next phase in the algorithm assigns address to the selected sensor head and associates the sensors with it. The address, in the form of Sensor Head ID, is sent to all the deployed sensors. This allows all the messages to be sent to this particular location. Whenever a Sensor Head is low on energy, it transfers its control to the sensor with priority next to the selected Sensor Head. This sensor, hence, becomes the new Sensor Head.

```
Phase III

if s.status = sensor_head then
   s.sensor_head = s.id;
else
    recv();
    if sensor_head!= NULL then
       s.status= sensor;
    send(3,s.id,s.sensor_head,sensor,s.Initial_prior-
    ity);
else
    s.status = sensor_head;
    s.sensor_head = s.id;
    send(2,s.id,sensors,sensor_head,s.priority);
```

Pseudocode for Phase III

3.4 Phase IV

In the last phase, the algorithm works towards preventing collision between the sensors in approaching the Sensor Head. This is achieved by synchronizing the local clocks of each sensor in order to perform TDMA activities. Each sensor sends a beacon message to the Sensor Head with its priority and waits for the response from the Head. The Head receives the message and synchronizes its local clock with the time stamp on the message from the sensor with higher priority.

```
Phase IV

if s.status = sensor_head then
          send beacon message with s.Initial_priority,
My_clock etc.;
          receive beacon message m from other sen-
sors;
if m.Initial_priority> s.Initial_priority then
          s.clock = m.clock
```

Pseudocode for Phase IV.

Hence, now we have the selected Sensor Head which receives information from the deployed sensors. The next step is to apply Data Fusion to send the integrated data to the Static Data Sink.

4. Data Fusion

Data Fusion, as explained above, is used to combine data from various sensors to generate an integrated data that can be transmitted to the sink for further processing. After the completion of Sensor Head Algorithm, we have the selected Sensor Head where all the data is received. Data Fusion initially combines all the data items into one data set while effectively removing the redundant items collected from the deployed sensors. Since the sensors are present in the uniform geographical location, the nature of data collected is generally similar. This means that redundancy will prevail in the collected data. Since the final result is in the form of an inference derived from the sensor data, redundant data does not have a role to play. Hence, a significant amount of load can be reduced on the system by taking only unique data items. In order to derive a unified fused output, we use Fuzzy Logic Controller (FLC). It works on the principles of Fuzzy Logic which involves generation of approximate models that match the human decision-making ability. "The Fuzzy Logic Controller provides a means of converting a linguistic control strategy based on expert knowledge into an automatic control strategy"[8]. Essentially, Fuzzy Logic models are based on a set of rules that are defined for a given system. The models derived are not precise but do match the human ability to make rational decisions in an uncertain environment[9]. The method used in the proposed model's approach of Fuzzy Logic Controller and has been inspired by the work done in[10]. While the defuzzification method used is called 'Centre of Gravity'. The unique data parameters collected at the Sensor Head are fed to FLC as input. Upon applying a set of rules designed for a given system, the output is generated by FLC in the form of Inference Factor (IF). This denotes the fused data of that particular set of input. The proposed system involves taking the reading of the Inference Factor (IF) for multiple data input to later take an average of all factors as the final data being sent to the data sink. Another major aspect of the proposed model is flushing of memory upon Data Fusion. Since we are removing the redundant data, it is very much necessary that we flush the memory after fusing individual data set in order to derive proper inference from the incoming sensor data. Not doing so may reject the incoming data that matches the previous data and shall eventually lead to an improper conclusion, defeating the complete purpose of the proposed method. Hence, the complete procedure of Data Fusion from removing the redundancy to flushing of memory has been explained using the algorithm.

Data Fusion

1: Collect data from (N-1) sensors to the S_Head;
2: Set = Set - (Redundant data from sensors);
3: Initialize (FLC);//fuzzy logic controller
4: Inference_Factor (IF)= FLC(Set); //test the input data with the defined rules to give result.
5: log Inference_Factor in a file;
6: Flush system memory;
7: Avg = Sum(Inference_Factor)/Set.size();
8: transmit the Avg() to data sink;
9: continue to the next cycle;

Pseudocode for Data Fusion.

5. Sample Output

The following section explains the working of proposed model using a set of random input values to derive the average of Inference Factor, which is sent to the data sink. In the given sample, we have deployed sensors to take the reading of humidity and temperature. The Inference Factor decides the possibility of precipitation on the basis of humidity and temperature values by using the defined rules in Fuzzy Logic Controller. It should be noted here as a disclaimer that the mentioned example is just a depiction of how the proposed model can be used to fuse the sensor data. The example mentioned can be modified to suit multiple real parameters to find actual results. Although precipitation is governed by a large number of factors, we have just used humidity and temperature to simulate the working of the model. Consider the following values of humidity and temperature collected from the deployed sensors. Serial number 1 – 20 depicts High humidity. 21 – 25 depicts Medium humidity and 26 – 30 depicts Low Humidity. Table 1.The data collected by the sensors now reaches the elected sensor head where the Data Fusion takes place. The primary step at the sensor head is the removal of the redundant data to create a unique set of data. This data is then taken as the input for Fuzzy Logic Controller which generates Inference Factor as the output. The non-redundant data with their respective inference factor values have been depicted for high, medium and low humidity data Table 2. The graphical output of the Fuzzy Logic Controller for the input values Humidity =

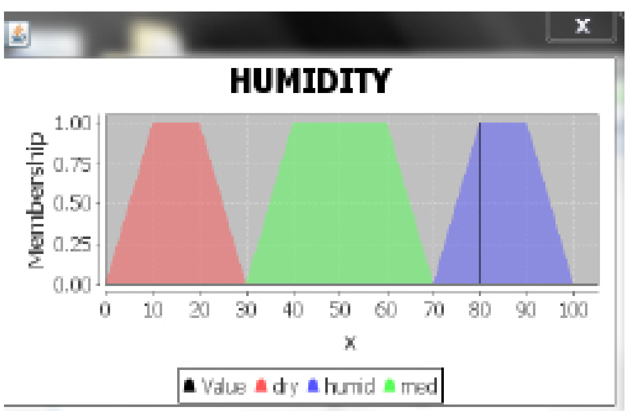
Table 1. Input values recorded by sensors

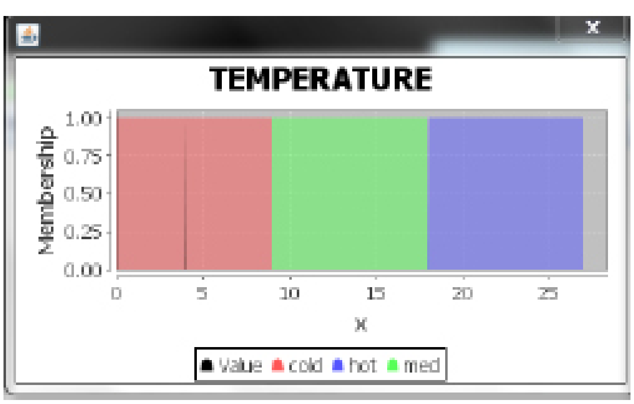
Serial no.	Humidity (%)	Temperature (°C)
1	80	23
2	81	22
3	82	20
4	82	20
5	82	20
6	78	23
7	78	23
8	78	23
9	78	23
10	78	23
11	81	22
12	82	20
13	82	20
14	80	23
15	81	22
16	79	20
17	80	20
18	79	21
19	80	20
20	79	21
23	50	29
24	52	28
25	51	29.2
26	52	27.5
27	20	20
28	23	20
29	19	22
30	21	21

Table 2. Output values for high humidity

Serial no.	Humidity (%)	Temperature(°C)	Inference Factor
1	80	23	7.499988
2	81	22	7.499990
3	82	20	7.499992
4	78	23	7.499996
5	79	20	7.499992
6	80	20	7.499988
7	79	21	7.499992

80%, Temperature = 23°C has been depicted Figure 3. Note that the temperature reading is depicted after subtracting 19 from the original output to fit the value in the graph. Therefore, temperature value for 23 °C is 23 – 19 = 4. The resultant value is hence 7.499988, which depicts high possibility of precipitation. The value finally sent to the data sink is the average of all the Inference Factors, which is equal to 7.4999114285 . Now, for the Medium humidity input values, the inference factors have been depicted as Table 3. The average resultant is obviously 4.5 in this case. The graphical representation for Humidity = 50%, Temperature = 29°C (29 – 19 = 10) has been given Figure 4. For the Low humidity input values, the inference factors have been depicted as Table 4. The average resultant value is 1.500009. The graphical representation for Humidity = 20 % and Temperature = 20°C (20 – 19 = 1) has been given Figure 5. The resultant value is 1.500012, as has been depicted in the graph. Hence, in the similar way the Wireless Sensor Networks can be used to derive a unified fused output for various other real-life applications by changing the number of parameters that affect the output. Fused data inference can give an insight on the possibility of occurrence of an event that is affected by the input values from the sensors.





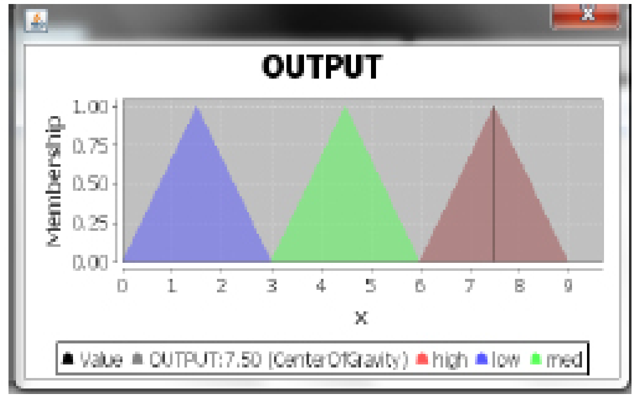
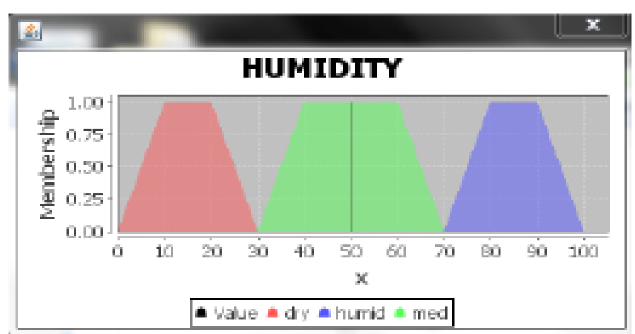
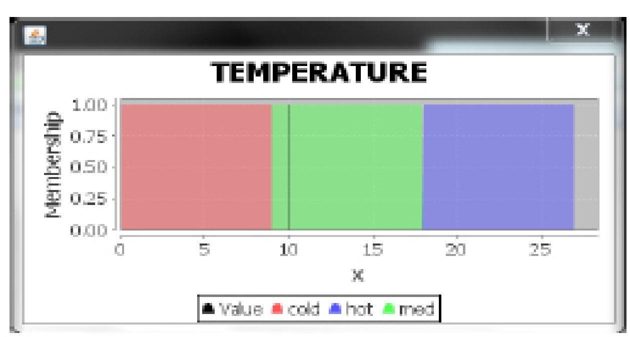


Figure 3. Graphical depiction of result with High humidity input.

Table 3. Output values for Medium Humidity

Serial no.	Humidity (%)	Temperature(°C)	Inference Factor
1	50	29	4.5
2	52	28	4.5
3	51	29.2	4.5
4	52	27.5	4.5





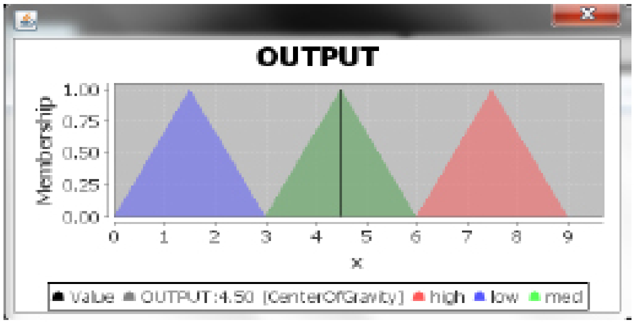


Figure 4. Graphical depiction of result with Medium humidity input.

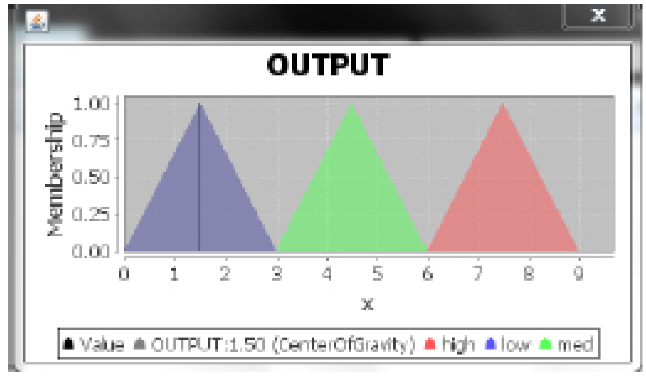
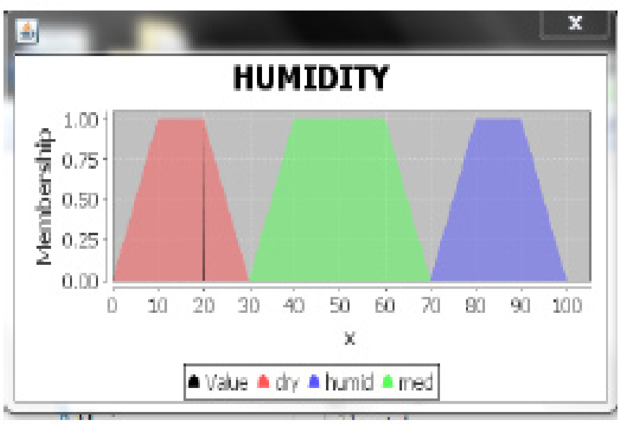
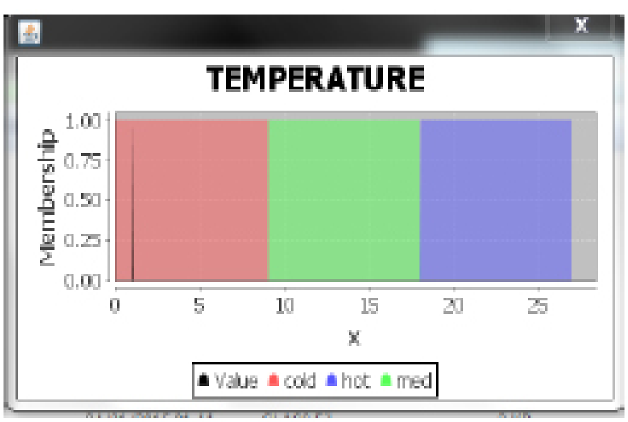


Figure 5. Graphical depiction of result with Low humidity input.

Table 4. Output values for Low Humidity

Serial no.	Humidity (%)	Temperature(°C)	Inference Factor
1	20	20	1.500012
2	23	20	1.50006
3	19	22	1.500010
4	21	21	1.500008





6. Future Work and Conclusion

As described, the work explained in the paper is currently under process. Although we worked out a sample output using simulation tools, we are yet to perform real-time testing using a working model of a network of wireless sensors to test the hypotheses and study the results. Upon complete implementation, we will be able to produce solid results to back our proposed theory. The paper can be concluded as the following. The sensors of a Wireless Sensor Network have great utility and have numerous applications wherever data recording is needed. The paper proposed Sensor Head Algorithm to evaluate the residual energies of the sensors and then prioritize them accordingly. This results in the selection of the Sensor Head, i.e., the sensor with the highest priority. The information collected by the sensors are sent to the Sensor Head, where it is converted to unified form using Data Fusion which essentially removes redundant data and finds the Inference Factor using Fuzzy Logic Controller. The resultant (IF) is then taken for multiple inputs, using which, an average is derived. This average of Inference Factors is the final result that is being transmitted to the static data sink.

7. References

1. Tavakkolai H, Yadollahi N, Yadollahi M, Rahmani Hosseinabadi AA, Kardgar M. Sensor Selection Wireless Multimedia Sensor Network using Gravitational Search Algorithm. Indian Journal of Science and Technology. 2015 Jul; 8(14):1–6.
2. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Asurvey on sensor networks. IEEE Communications Magazine. 2002 Aug; 40(8):102–14.

3. Load Balanced Clustering in Wireless Sensor Networks. Available from: http://ieeexplore.ieee.org/document/1203919/. Date Accessed: 20/06/2003.
4. Hall DL, Llinas J. An Introduction to Multisensor Data Fusion. Proceedings of the IEEE. 1997 Jan; 85(1):6–23.
5. Thiriveni GV, Ramakrishnan M. Distributed Clustering based Energy Efficient Routing Algorithm for Heterogeneous Wireless Sensor Networks. Indian Journal of Science and Technology. 2016 Jan; 9(3):1–6.
6. Zhao M, Yang Y, Wong C. Mobile Data Gathering with Load Balanced Clustering and Dual Data Uploading in Wireless Sensor Networks. IEEE Transactions on Mobile Computing. 2015 Apr; 14(4):770–85.
7. Thenmozhi E, Audithan S. Energy Efficinet Cluster Head Selection and Data Convening in Wireless Sensor Networks. Indian Journal of Science and Technology. 2016 Apr; 9(16):1–6.
8. Lee CC. Fuzzy Logic in Control Systems: Fuzzy Logic Controller- Part I. IEEE transactions on systems, man and, cybernetics. 1990 Mar-Apr; 20(2):1–15.
9. Zadeh LA. Knowledge Representation in Fuzzy Logic. IEEE Transactions on Knowledge and Data Engineering. 1989 Mar; 1(1):89–100.
10. Izadi D, Abawajy JH, Ghanavati S, Herawan T. Data Fusion method in Wireless Sesnsor Networks. Sensor. 2015 Jan; 15(2):2964–79.