Journal of AI and Data Mining
Vol 3, No 1, 2015, 47-57.
doi:10.5829/idosi.JAIDM.2015.03.01.06

FDMG: Fault detection method by using genetic algorithm in clustered wireless sensor networks

A. Ghaffari* and S. Nobahary

Department of Computer Engineering, Tabriz branch, Islamic Azad University, Tabriz, Iran.

Received 3 December 2014; Accepted 5 April 2015
**Corresponding author: a.ghaffari@iaut.ac.ir (A. Ghaffari).*

Abstract

Wireless sensor networks (WSNs) consist of a large number of sensor nodes which are capable of sensing different environmental phenomena and sending the collected data to the base station or Sink. Since sensor nodes are made of cheap components and are deployed in remote and uncontrolled environments, they are prone to failure**.** Thus, maintaining a network with its proper functions even when undesired events occur is necessary and is called fault tolerance. Hence, fault management is essential in these networks. In this paper, a new method has been proposed with particular attention to fault tolerance and fault detection in WSN. The performance of the proposed method was simulated in MATLAB. The proposed method was based on majority vote, which can permanently detect faulty sensor nodes accurately. High accuracy and low false alarm rate helped exclude them from the network. To investigate the efficiency of the new method, the researchers compared it with Chen, Lee, and hybrid algorithms. Simulation results indicated that the proposed method has better performance in parameters such as detection accuracy (*DA*) and a false alarm rate (*FAR*) even with a large set of faulty sensor nodes.

Keywords: *Wireless Sensor Networks, Fault Detection, Genetic Algorithm, Fault Diagnosis, Clustering Algorithm.*

1. Introduction

Recent advancements in Micro-Electro-Mechanical Systems (MEMS) technology and wireless communication have promoted the emergence of a new generation technology which is called WSN. It consists of tiny, inexpensive sensors with limited processing and computing resources. These sensor nodes can sense, measure, and gather information from the environment. Hence, they have been used in many applications such as environmental monitoring, object tracking, agricultural lands, office buildings, industrial plants and military systems [1-3].

It is obvious that sensor networks are prone to failure which is mainly due to the fact that many applications require deploying sensors in harsh and contaminated environments such as battlefield. Fault detection and fault tolerance in wireless sensor networks have been investigated in the literature. Moreover, deployed sensor networks may suffer from many faults because of environmental impacts such as lightning, dust and moisture which can reduce the quality of wireless communications and divert sensors from their desirable operations. Moreover, hardware defects of sensors are related to cheap sensors prices which have low quality electronic components; such sensors are used in the construction of sensors which can negatively affect desirable network operations. Also, software bugs have such negative impacts on network operations[4]. These faults can be the cause of data failure and functional failures[5]. Data faults and failures result in inappropriate response of the network manager and faulty nodes bring about inaccurate routing by directing data through intermediate faulty nodes. Accordingly, it is essential to detect and manage faults in WSNs.

As mentioned above, due to the failure of network, there should be a kind of responsibility for avoiding failure so that network fault tolerance is guaranteed. In general, the first step in enhancing fault tolerance in a system is to try to use fault avoidance techniques so as to avoid damaging factors. To achieve this objective, one should use high-technology electronic devices, advanced equipment for designing, constructing and strict compliance of the design roles and testing stages. It should be noted that the first two cases, in particular, will increase the cost of production and is not operational for such networks. On the other hand, the two other cases only ensure reliability of performance accuracy for each sensor node in the construction stage and there is no guarantee for network operation against environmental factors. Consequently, fault avoidance techniques should be used in a network as well as other mechanisms so that network can continue to function properly. These mechanisms are referred to as fault tolerance techniques. The networks having the above-mentioned capability are known as fault-tolerant networks. In general, four types of redundancies, namely, hardware, software, information and time redundancy are used in the development of fault tolerant systems [6]. Using the first two redundancy types significantly increases the cost of production; hence, they are not appropriate for WSNs. In contrast, the other two redundancy types are used in some protocols which are proposed for these networks.

There are several sophisticated techniques and methods for detecting faults in WSNs. For instance, one highly powerful method, i.e. the majority vote method, is appropriate for detecting faults. This method makes use of genetic algorithms (GAs). GA is aimed at using natural evolution and a fitness value for each possible solution to the problem. The best GA choice and candidate is a representation of candidate solutions to the problem in (genotype).The initial population randomly produces a fitness function. It measures and compares each solution in the population; genetic algorithm operates the crossover and mutation functions to produce new Generation. Finally, the algorithm tunes parameters such as population size either finds the best data or finishes the time of execution, etc. Successful application of GAs in sensor network designs [7] has resulted in the development of several other GA-based application-specific approaches in WSN design mostly by the structure of a single fitness function [5,8,9]. Also, it has led to meditation optimality in the evaluation of fitness values[7].However, in the majority of these methods, very limited network characteristics are considered; hence, several requirements of application cases are not taken into consideration in the performance measure of the algorithm.

However, in this paper, the researchers proposed a new method to solve the problem of majority vote. Moreover, it should be noted that by using GA, the proposed method can detect faulty sensors with high detection accuracy and low false alarm rate. In the proposed method, GA was used in sinking to select the best data and to define the status of each sensor node.

The rest of the paper is organized as follows: Section (2) provides a brief overview of fault detection methods in WSNs and related works; then in section 3, the proposed method was explained and network models are discussed in detail. The results of the simulation are mentioned and evaluated in section 4. Ultimately, section 5 sums up the findings, concludes the study and suggests directions for future works.

2. Related works

In this section, common and related algorithms and methods to fault detection literature are reviewed [10-14]. These methods use majority vote but they can’t detect common failure nodes.

Chen et al.[15]have proposed a new distributed fault detection algorithm for wireless sensor networks. In this algorithm, data of sensors were compared twice to achieve a final decision on the status of sensors; moreover, four steps have to be taken and the improved majority voting was used. Two predetermined threshold values, marked up by θ_1 and θ_2, were used. Each sensor node compared its own sensed data with the data of neighbor nodes in the time stamp t; if the difference between them was greater than θ_1, the comparison would be repeated in the time stamp t+1; in case the difference was greater than θ_2, too, it was interpreted that data of this node was not similar to data of the neighbor nodes. In the next step, each sensor defined its own status as likely good (*LG*) if its own sensed data was similar to at least half of the neighbors’ data. Otherwise, the sensor status would be defined as likely faulty (*LF*). In the next step, each sensor can determine its own final status according to the assumption that the sensor status is GOOD (*GD*) if it determined its status as *LG* in the previous step and more than half of the neighbors are *LG*. Then, sensors whose statuses are *GD* broadcast their status to their neighbors. A sensor node with an undetermined status can determine its status using the status of its neighbors. If a sensor node whose status is defined as *LG* and receives *GD* status from its neighbor node whose own sensed data is similar to the data of the sender of this message;

hence, it changes its status to *GD*. If a sensor whose status is defined as *LF* and receives faulty status from its neighbor whose own sensed data is similar to the data of the sender of this message, then it will change its status to faulty. The complexity of this algorithm is low and the probability of fault detection accuracy is very high. This algorithm only detects permanent faults while transient faults are ignored although these types of faults may occur in most of the sensor nodes.

Lee et al. [16] proposed a distributed fault detection algorithm for wireless sensor networks which is simple and highly accurate in detecting faulty nodes. This approach used time redundancy for increasing the tolerance of transient faults. In this method, two predetermined threshold values marked up by θ_1 and q were used. Every node compared its own sensed data with data from its neighbor nodes q times in order to determine whether its data are similar to the data of neighbors or not. In the next step, the sensor status would be defined as fault-free if its sensed data is similar to at least θ_1 of the data of neighbor nodes. Each sensor whose status is determined will broadcast its status to undetermined sensors so that they define their status. Simulation results in that study indicated that fault detection accuracy of this algorithm would decrease rapidly when the number of neighbor nodes was low but fault detection accuracy would increase when the number of neighbor nodes was high. The disadvantage of this algorithm is that it is not able to detect common mode failures.

As mentioned above, most fault detection algorithms [6, 16-20] in WSNs compare their own sensed data with the data of neighbor nodes. If their data is similar to at least half of the data sensed by neighbors, the cited sensor will be considered as fault-free. Comparison-based fault detection methods suffer from several deficiencies. They are unable to detect faulty nodes in remote areas where sensors do not have any availability to data of neighbors' nodes in their transceiver boards. The poor performance of algorithms in detecting common mode failures is another problem for these techniques.

With respect to the research gap highlighted above, in this paper, the researchers proposed a distributed method which is able to detect faulty nodes. To increase load balancing and lifetime of WSNs, different clustering algorithms are used. NHEEP[3](anew hybrid energy efficient partitioning approach for WSN clustering) is a clustering approach based on a partitioning technique in which the number of partitions are determined by the sink. After partitioning, each node can determine which partition is present. For electing a CH (cluster head) inside a partition, different parameters are considered such as position, distance of the nodes and the residual energy of nodes. NHEEP takes two important parameters into account for selecting cluster head node as follows:

Energy: Due to the lack of energy sources needed for regulating the lifetime of WSNs, energy is one of the most important parameters in research on WSNs. The residual energy is very important for the cluster head. Cluster head is negatively affected by high energy consumption of cluster members. Inasmuch as cluster head is responsible not only for gathering data from cluster members but also for processing data aggregation and data transmission to the sink; hence, its energy runs out very quickly. A qualified node for the cluster head is selected to ensure uninterrupted accomplishment of tasks. This node has more residual energy compared with others nodes.

Centrality: Sometimes the density of a node is high but the nodes which are around that node are only in one side of the mentioned node. When the nodes are in the central part of the area, they play an important role in network structure because the central nodes have an important role in transmitting data to the next step. Thus, it is preferable to have cluster heads in central neighborhood to maintain load balance.

In this algorithm, firstly, nodes identify their own clusters and then they try to select the best node in terms of high centrality and high remaining energy in each cluster as cluster head. In this algorithm, the cluster head collects the data of the cluster and sends it to the other cluster heads to send to the sink through multi-hop approach.

In the next section, the proposed fault detection algorithm is described. In this method, the network is clustered by using NHEEP [3] algorithm.

Then, in the stability phase, before the transmitting data to the cluster head, faulty sensors will be detected according to the fault detection algorithm. This fault detection phase is repeated in proportion to the existing noise in the operational area. In the time slot, fault detection process takes place, and data sensing and transmission will stop. After the mentioned time slot, the network continues its operation again.

3. Proposed method

In this section, fault model, variables and assumptions used in the proposed method are described.

Fault model:

In detecting WSN faults, nodes with faulty state and permanent communication faults are spotted. Since selfish sensor nodes with malfunctioning behavior are still capable of routing information, they could participate in the network operation. However, the sensor nodes with a permanent communication fault (including lack of power) are eliminated from the network [23-24].

Definitions:

The notations used in proposed fault detection algorithm are listed as follows:

- n: total number of sensor nodes distributed throughout the environment;
- M: number of clusters in the network;
- $L=\left[{}^{N}/_{M}\right]$: length of chromosomes;
- x_i: data of i-nodes;
- $|x_i - x_j|$: fitness function;
- θ_1: predetermined threshold value;
- R_r: the best selected gen among all gens

The proposed fault detection algorithm includes three phases: (a) setup and clustering process, (b) fault detection phase, and (c) Data transmission and updating phase. The above-mentioned three phases are described below.

3.1. Set up and clustering processes

In this phase, the deployed sensor nodes identify their neighbor nodes, create neighbor table and create clustering process. For creating the neighbor table, each sensor node sends a hello message at the beginning to identify its neighbors. Hello message includes the identification number of sensor nodes, node coordination and residual energy level of nodes. Figure 1 shows the structure of the hello packets as follows:

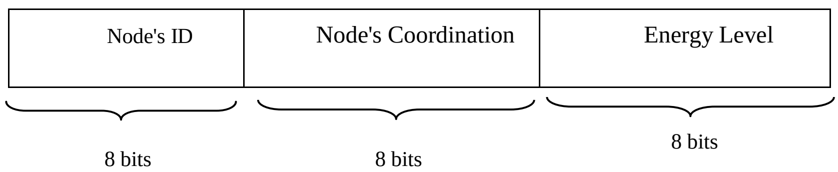

Figure 1. Structure of HELLO message.

The neighbor nodes receiving the hello message respond to the sender node by sending echo message. This echo message includes the node's *ID* (Identifications), its distance to sink, and energy level (residual energy) of neighbor nodes. Figure 2 represents the format and number of bits of response package message (echo packet) as follows:

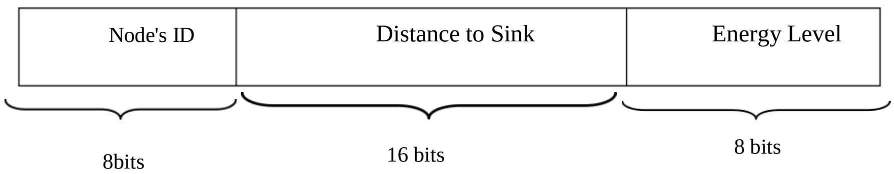

Figure 2. Structure of response (echo) message.

All nodes receive the parameters of their neighbor nodes and store them in the neighborhood table. At this stage, nodes automatically try to select a cluster and their cluster heads. Clustering operation in the proposed method was carried out with the partitioning algorithm which was described in the previous section. For selecting cluster heads, the proposed scheme took the following parameters into account: centrality and residual energy. After choosing cluster heads, cluster member nodes introduce themselves to the cluster head and practically justify their membership in the cluster. In the next step, the cluster members gather the information from the occurred events and send the data packets to the sink node after data aggregation processing. Figure 3 depicts clustered WSN model with faulty sensor nodes, faulty cluster head, fault-free sensor nodes and fault-free cluster head nodes.

3.2. Fault detection phase

During the normal operation of network, nodes send their data to the cluster head. However, in the fault diagnosis and fault detection phase, each member node of the cluster sends data to nodes of each cluster head. Thus, cluster heads will select the best data of the clusters by applying genetic algorithm and sending them to the sink. Also, sink applies GA whose properties are determined in the following section. The best data is selected between cluster head data previously sent to sink and the status of each cluster head is determined by the sink.

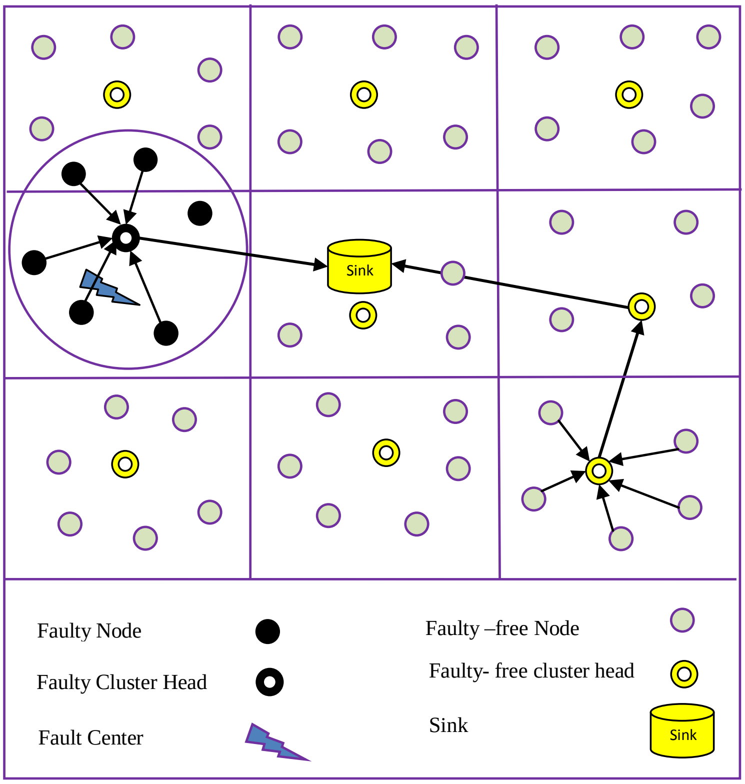

Figure 3. Wireless sensor network with fault-free and faulty nodes and cluster head.

The selected data of sink is broadcast as a message to all the cluster heads. Fault-free cluster heads can determine the status of the cluster nodes but faulty cluster heads should be changed and given to a healthy cluster head. Figure 4 depicts the structure and number of bits of a broadcast message. Cluster head with a faulty plate is detected by broadcasting messages to all members of the cluster to choose their cluster head node again. It is obvious that the choice of a new cluster head node is healthy. If 30% or less nodes on the cluster nodes as well-known nodes are allowed to join, the cluster nearby nodes is proportional to their distance.

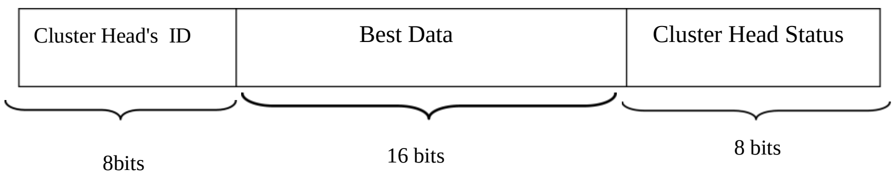

Figure 4. Structure of broadcast message.

Cluster head with a faulty plate is detected by broadcasting messages to all members of the cluster to choose their cluster head node again. It is obvious that the choice of a new cluster head node is healthy. If 30% or less nodes on the cluster nodes as well-known nodes are allowed to join, the cluster nearby nodes is proportional to their distance.

Before determining $\theta_1$parameter of member nodes, a definite threshold fitting data network is determined. If the dispute is declared as the best dispute in a particular node, it is greater than θ_1. Nodes will be regarded as faulty nodes; otherwise, the node is known to be healthy. The proposed method reveals that the decision about nodes' status and cluster heads are determined in the sink. Indeed, the method introduced in this paper is a combined approach for selecting the best data based on genetic algorithms.

In the first generation, a chromosome whose genes are real numbers and whose chromosome length is equal to the number of nodes in each cluster. Number of nodes in each cluster is

denoted by the symbol L. Values of each gene is equal to the amounts of data sensed by each node. In the proposed algorithm, the number of generations is one thing and the fitness function is used between each gene and other genes of chromosome according to (1). Hence, the last gene will continue using the same gene L and each of the obtained data is gathered and placed in the W and generalizes the corresponding gene in an array according to (2).
Finally, the fitness of a gene (least amount of conflict with other genes) is the most as the genes in the chromosome (R_r) is selected.

$$F = |x_i - x_j| \tag{1}$$

$$W_1 = |x_1 - x_1| + |x_1 - x_2| + \ldots + |x_1 - x_{L-1}| + |x_1 - x_L|$$
$$W_2 = |x_2 - x_1| + |x_2 - x_2| + \ldots + |x_2 - x_{L-1}| + |x_2 - x_L|$$
$$\vdots$$
$$W_L = |x_L - x_1| + |x_L - x_2| + \ldots + |x_L - x_{L-1}| + |x_L - x_L| \tag{2}$$

Figure 5 shows the chromosome of the best gene.

W_1	W_2	…	W_L

Figure 5. The chromosome of the best gene.

For ensuring the accuracy of selecting correct data in each cluster, the algorithm repeats it 10 times to set chromosomes R_0, R_1,...,R_9.
Then, the highest fitness chromosome is selected. Repeating the fitness function diminishes the probability of transient fault occurrence in the network.

3.3. Data transmission and updating phase
Data transmission phase corresponds to network application based on event occurrence; alternatively, the sensed data collected to the cluster head node and it forwards the aggregated data to the sink.
Since faulty node detection consumes considerable energy, it is not used in all the stages of data collection. In each cluster, sensor nodes send data to the cluster head.
Cluster node controls, aggregates and sends data to the sink.
In the proposed algorithm, new cluster head will be selected if cluster head node is faulty or its battery is low. In this case, cluster members select a new cluster head. Figure 6 illustrates the pseudo code of the first and third stages.

Algorithm phase 1(layer 1)

```
Step 1: Each node Si sets its status to H, send hello message to each node, and each node is neighbor sends reply message.
   Establish clusters and definite cluster head.
   Step 2: Fault detection done for first time (and in each Rr round)
   Each node Si sends its data to CHi(for example r=10)
        for  k=1  to rsi  sends data to CHi
          for j=1 to L do
Wk= |s_i^k − s_j^k|
        Best(i)=Min(Wi)
   Send Best(i) for each cluster to sink
for j=1 to ido
Wi= |Best_i − Best_j|
   Total Best=Min(W)
Each CHi determines its  status
        if |X_CHi − TotalBest| ≤ θ1   then      TCHi=H
             else  TCHi=F
        if  TCHi=F   then  elect another CHi
Each Si determines its  status
        if |X_si − TotalBest| ≤ θ1   then      TSi=H
             else  TSi=F
Step 3: Send data to sink and update network status
```

Figure 6. Pseudo code of the first to third stages.

4. Simulation results
4.1. Network model
The proposed method was simulated in MATLAB software. n sensors were randomly deployed in $A*A$(m^2) square area which was aimed at collecting data during each round. It was assumed that the sink was in the middle of the area with the coordinate of (A/2, A/2).The simulation was repeated in 1000 cycles and the simulation parameters were indicated in Table 1.The following two metrics, detection accuracy (DA) and false alarm rate (FAR) are used to evaluate the performance, where DA is defined as the ratio of the number of faulty sensor nodes detected to the total number of faulty nodes and FAR is the ratio of the number of fault-free sensor nodes diagnosed as faulty to the total number of fault-free nodes [23]. In this simulation and performance evaluation, nodes with some transient faults are treated as fault-free nodes [23].A simple model for radio hardware energy dissipation was used where the transmitter dissipates energy to run radio electronics, and power amplifier and the receiver dissipates energy to run the radio electronics.
Based on the model, the network had the following features:

- All nodes were uniformly distributed within a square area.
- Each node has a unique *ID*.
- Each node has a fixed location.

- All nodes can perform data aggregation.
- Transmission energy consumption was proportional to the distance of the nodes.

Both free space and multi-path fading channel models were used for the experiments of this study based on the distance between the transmitter and receiver. Thus, energy consumption for transmitting a packet was calculated for l bits over distance d by (3) as follows [21]:

$$E_{tx}(l,d) = E_{tx-elec}(l) + E_{tx-amp}(l)$$

$$= \begin{cases} E_{tx}(l,d) = l.E_{elec} + l.\varepsilon_{fs}d^2 & d < d_0 \\ l.E_{elec} + l.\varepsilon_{amp}.d^4 & d > d_0 d_0 \end{cases} \quad (3)$$

According to the above-mentioned energy consumption model, if the distance between sensor node and base station (BS)is less than a threshold d_0, as calculated by (4), the free space (f_s) model will be used; otherwise, the multi-path (*mp*) model will be used [21].The $d_0$parameter can be calculated as follows [21]:

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{amp}}} \quad (4)$$

Table 1 shows the values of ε_{fs} and ε_{amp}. Energy consumption for receiving a packet of l bits is calculated according to (5) [21] as follows:

$$E_{RX}(l) = E_{RX-elec}(l) = l.E_{elec} \quad (5)$$

The probability of faulty sensor nodes was assumed to be 0.10, 0.2, 0.3, 0.4 and 0.5. The number of included nodes was assumed to be 100 and 150, respectively.

Table 1. Simulation parameters.

Parameter	Value
Number of sensors	100, 150
Area	400×400 (m^2)
Sink position	(200, 200)
d_0	87 m
Radio range	70 m
E_{elec}	50nj/bit
ε_{fs} (if destination to BS<=d_0)	10pj/bit/m^2
ε_{amp} (if destination to BS >=d_0)	0.0013 pj/bit/m^4
Initial energy	1j
E_{da} (Data aggregation energy)	10 nj/bit/packet
Packet size	4000 bits
Simulation repeate	1000 cycles

4.2. Simulation results and performance evaluation

The efficiency of the proposed method was evaluated and compared with Lee [16] and Chen [15] algorithms in terms of detection accuracy and false alarm rate parameters. Whereas *DA* was defined as the ratio of the number of detected faulty nodes to the total number of faulty nodes, *FAR* was defined as the ratio of the number of fault-free nodes that are detected as faulty node to the total number of fault-free nodes [22]. Table 2 compares the fault detection accuracy in the proposed scheme, Chen [15], Lee [16] and Hybrid [6] algorithms.

Table 2. Fault detection accuracy in the proposed method, Chen [15], Lee [16] and hybrid [6] algorithms.

Algorithms								
Chen [15]	**Lee [16]**	**Hybrid [6]**	**Proposed**	**Chen [15]**	**Lee [16]**	**Hybrid [6]**	**Proposed**	**P**
0.984	0.986	0.988	1	0.984	0.986	0.988	1	0.1
0.982	0.984	0.985	1	0.982	0.984	0.985	1	0.2
0.96	0.97	0.977	1	0.96	0.97	0.977	1	0.3
0.97	0.50	0.6	1	0.95	0.5	0.6	1	0.4
0	0	0	0.42	0	0	0.1	0.34	0.5
n=150				*n*=100				

Figures 7 and 8 show the comparison of the proposed algorithm with the algorithms of Chen [15], Lee [16], and hybrid [6] respectively, in terms of detection accuracy and false alarm rate with 100 nodes in network.

When the probability of sensor failure was 0.1, the detection accuracies of Chen [15], Lee [16], and hybrid [6] algorithms were 0.986, 0.984, 0.988 and 0.986 respectively. However, the detection accuracy of the proposed algorithm was equal to

1.When the probability of the sensor failure was 0.25, the detection accuracies of Chen [15], Lee [16], and hybrid [6] algorithms were 0.975, 0.97, and 0.977, respectively. However, it should be noted that the detection accuracy of the proposed algorithm was equal to 0.981. When the probability of sensor failure was 0.25, the false alarm rate in Lee [16] and Chen [15] algorithms was 0.0018 and 0.0021, respectively. In contrast, the false alarm rate of the proposed algorithm was equal to 0.0013.

Figures 9 and 10 compare the proposed algorithm with that of Chen [15] and Lee [16] in terms of detection accuracy and false alarm rate when there were 150 nodes in the network.

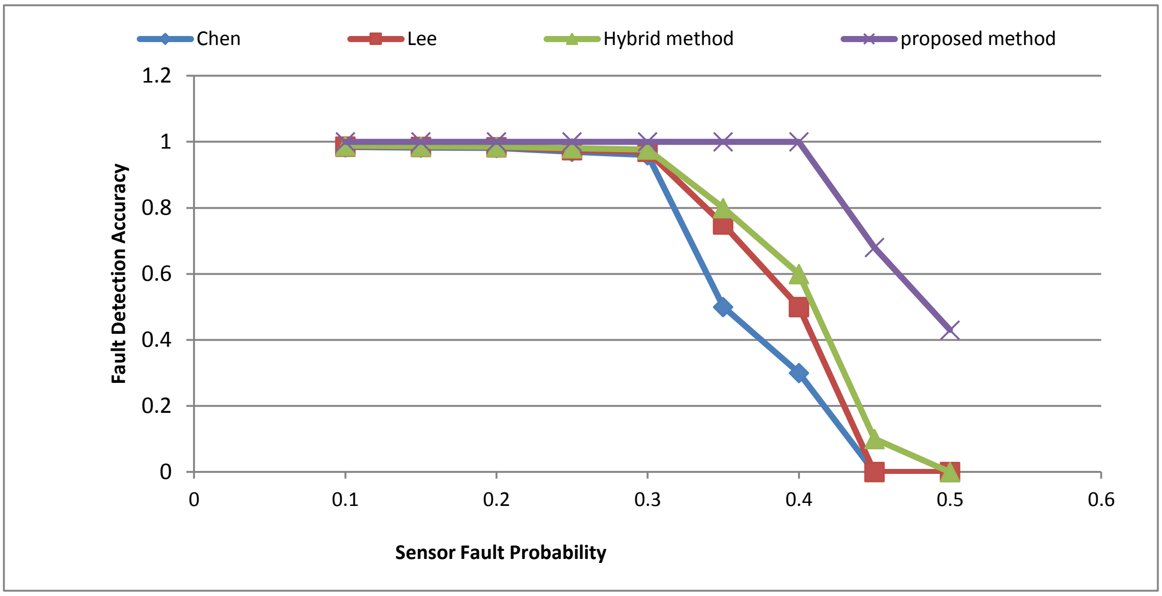

Figure 7. Fault detection accuracy when *N*=100.

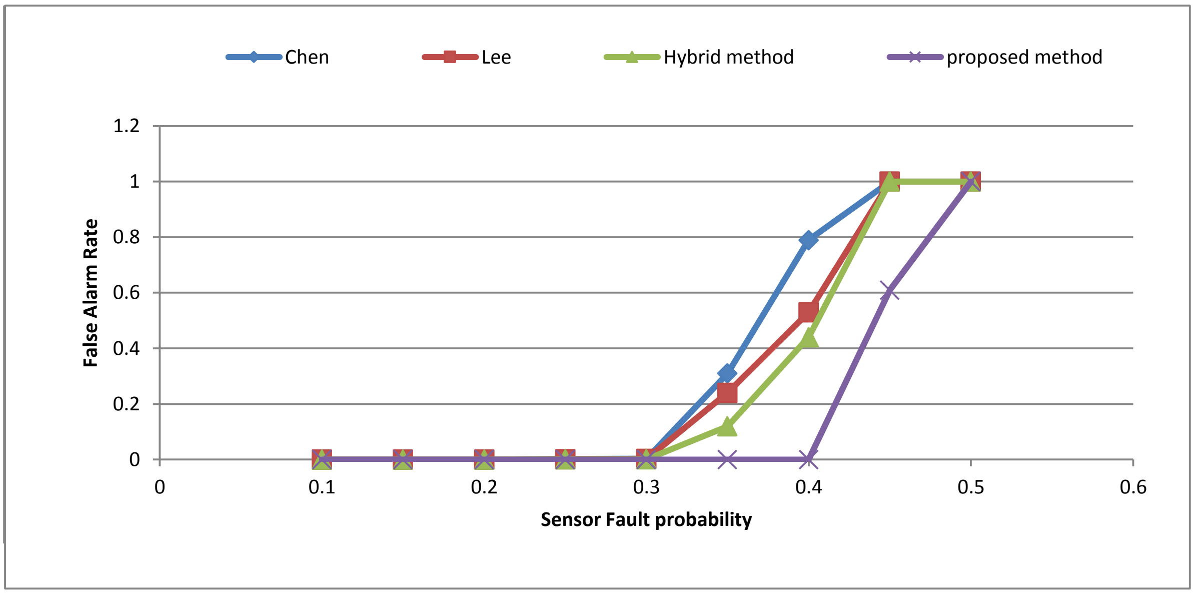

Figure 8. False alarm rate when *N*=100.

When the probability of sensor failure was 0.1, the detection accuracy in both Lee [16] and Chen [15] algorithms was 0.999. However, the detection accuracy of the proposed algorithm was equal to 1.

When the probability of the sensor failure was 0.25, the detection accuracy of Lee [16], Chen [15], and hybrid [6] algorithms were 0.993, 0.991, 0.991 and 0.994, respectively. Nevertheless, the detection accuracy of the proposed algorithm was equal to 0.994.

Similarly, when the probability of sensor failure was 0.15, then, the false alarm rate of Chen [15] algorithm was 0.0001. In contrast, the false alarm rate of Lee [16], hybrid [6] and the proposed algorithm was equal to zero.

When the probability of sensor failure was 0.25, the false alarm rate of Lee [16], Chen [15], and hybrid [6] algorithms were 0.0012, 0.0014, 0.0007and 0.0004, respectively but that of the proposed algorithm was equal to 0.0006. In other words, as the probability of sensor failure increases, the false alarm rate of the proposed algorithm was less than those of Lee [16] and Chen [15] algorithms. Based on figures 7 and 8, the researchers can draw the conclusion that the detection accuracy increases as the number of neighbors' increases but false alarm rate decreases. Furthermore, as the probability of sensor failure increases, detection accuracy of the proposed algorithm will be higher than those of Lee [16] and Chen [15] algorithms.

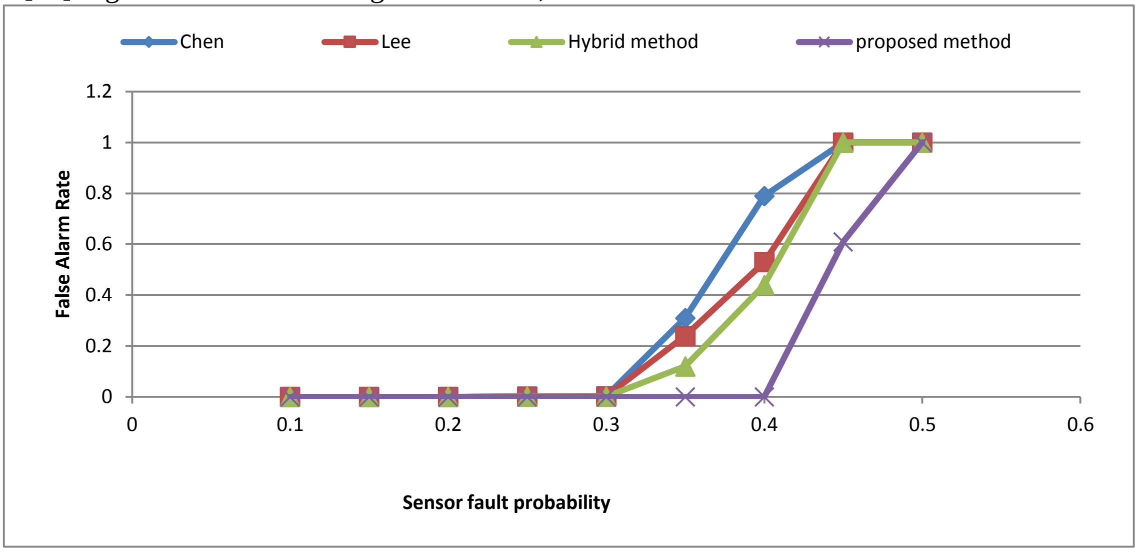

Figure 9. Fault alarm rate when N=150.

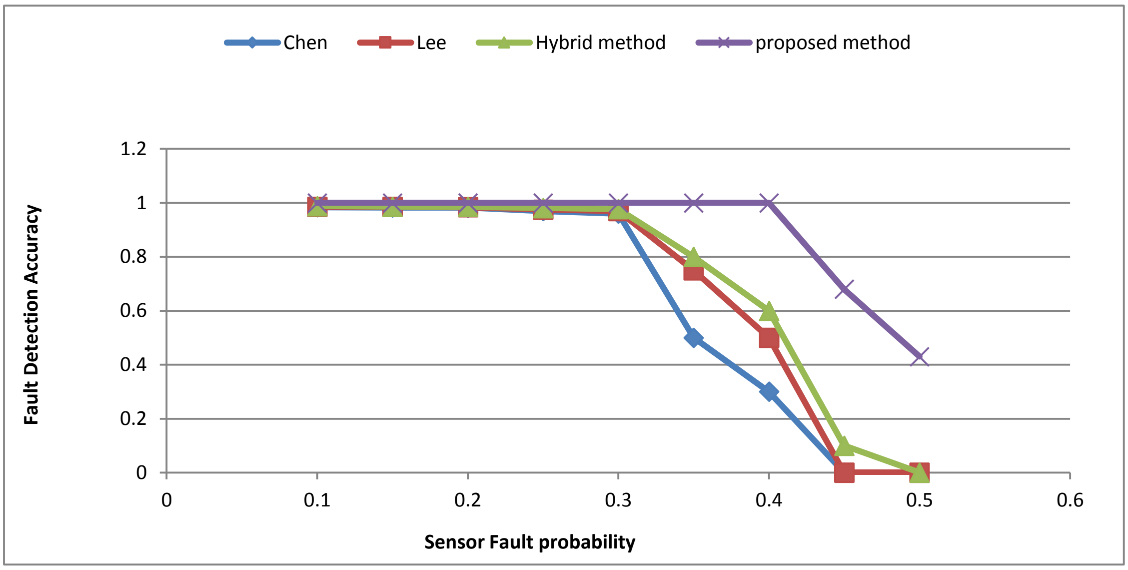

Figure 10. Fault detection accuracy when N=150.

5. Conclusion and directions for further research

Inasmuch as the failure rate of WSNs is remarkable, fault tolerance should be regarded as a significant attribute in these networks; this feature can be utilized to detect faulty nodes and emit them from the network. This paper presents a distributed fault detection algorithm for wireless sensor networks. In this paper, based on GAs, researchers proposed a new method for detecting faulty node. The proposed method was also intended to detect permanent faults in sensor nodes with an extremely high detection accuracy and low fault alarm rate in the network. The proposed algorithm is simple and detects faults in WSN with high accuracy. Faulty sensor nodes are identified based on comparisons between neighboring nodes and dissemination of the decision made at each node and each cluster. Simulation results revealed that the proposed method demonstrates better performance across parameters such as *DA* and *FAR* even when the number of faulty sensor nodes is high.

A direction for further research can use a combination of the method proposed in this paper with a learning automata technique for fault detection and network fault tolerance enhancement.

References

[1] Ian, F. A., Weilian, S., Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey. Computer networks, vol. 38, pp. 393-422.

[2] Yick, J., Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey. Computer networks, vol. 52, pp. 2292-2330, 2008.

[3] Babaei, S., B., Zekrizadeh N. & Nobahari, S. (2012). NHEEP: A New Hybrid Energy Efficient Partitioning Approach for Wireless Sensor Network Clustering. International Journal of Information and Electronics Engineering, vol. 2, pp. 323-327.

[4] Chenglin, Z., Xuebin, S., Songlin, S. & Ting, J. (2011). Fault diagnosis of sensor by chaos particle swarm optimization algorithm and support vector machine. Expert Systems with Applications, vol. 38, pp. 9908-9912.

[5] Warriach, E., Nguyen, T., Aiello, M. & Tei, K. (2012). Fault detection in wireless sensor networks: a hybrid approach. ACM/IEEE 11th International Conference onInformation Processing in Sensor Networks (IPSN), pp. 87-88.

[6] Warriach, E., Nguyen, T., Aiello, M. & Tei, K. (2012). A hybrid fault detection approach for context-aware wireless sensor networks. 2012 IEEE 9th International Conference onMobile Adhoc and Sensor Systems (MASS), pp. 281-289.

[7] Jourdan, D. & Weck, O. (2004). Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. 2004-Spring. 2004 IEEE 59thVehicular technology conference (VTC), pp. 2466-2470.

[8] Aldosari S., & Moura, J. (2004). Fusion in sensor networks with communication constraints. In Proceedings of the 3rd international symposium on Information processing in sensor networks, pp. 108-115, 2004.

[9] Mitchell, R. & Chen, R. (2014). survey of intrusion detection in wireless network applications. Computer Communications, vol. 42, pp. 1-23, 2014.

[10] Bari, A., Jaekel, A., Jiang, J. & Xu, Y. (2012). Design of fault tolerant wireless sensor networks satisfying survivability and lifetime requirements. Computer Communications, vol. 35, pp. 320-333.

[11] Ding, M., Chen, D., Xing, K. & Cheng, X. (2005). Localized fault-tolerant event boundary detection in sensor networks. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE in INFOCOM, 2005, pp. 902-913, 2005.

[12] Geeta, D., Nalini, N. & Biradar, R. C. (2013). Fault tolerance in wireless sensor network using hand-off and dynamic power adjustment approach. Journal of Network and Computer Applications, vol. 36, pp. 1174-1185.

[13] Jiang, P. (2009). A new method for node fault detection in wireless sensor networks. Sensors, vol. 9, pp. 1282-1294.

[14] Kaushik, B., Kaur, N. & Kohli, A. K. (2013). Achieving maximum reliability in fault tolerant network design for variable networks. Applied Soft Computing, vol. 13, pp. 3211-3224.

[15] Chen, J., Kher, S. & Somani, A. (2006). Distributed fault detection of wireless sensor networks. in Proceedings of the 2006 workshop on dependability issues in wireless ad hoc networks and sensor networks, pp. 65-72.

[16] Lee, S., Choe, H., Park, B., Song, Y. & Kim, C. (2011). LUCA: An energy-efficient unequal clustering algorithm using location information for wireless sensor networks. Wireless Personal Communications, vol. 56, pp. 715-731.

[17] Banerjee, I., Chanak, P., Rahaman, H. & Samanta, T. (2014). Effective fault detection and routing scheme for wireless sensor networks. Computers & Electrical Engineering, vol. 40, pp. 291-306.

[18] Cho, K., Jo, M., Kwon, T., Chen, H. & Lee, D. (2013). Classification and experimental analysis for clone detection approaches in wireless sensor networks. IEEE journal of Systems, vol. 7, pp. 26-35.

[19] Mahapatro, A. & Khilar, P. (2013). Fault diagnosis in wireless sensor networks: A survey. IEEECommunications Surveys & Tutorials, vol. 15, pp. 2000-2026.

[20] Xie, M. Hu, J., Han, S. & Chen, H. (2013). Scalable hypergrid k-NN-Based online anomaly detection in wireless sensor networks. IEEE Transactions onParallel and Distributed Systems, vol. 24, pp. 1661-1670.

[21] Heinzelman, W. B., Chandrakasan, A. P. & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications, vol. 1, pp. 660-670.

[22] Guo, S., Zhong, Z. & He, T. (2009). Find: faulty node detection for wireless sensor networks. In Proceedings of the 7th ACM conference on embedded networked sensor systems, pp. 253-266.

[23**]** Lee, M. & Choi, Y. (2008). Fault detection of wireless sensor networks. Computer Communication, vol. 31, no. 14, pp. 3469–3475.

[24] Ghaffari, A. (2015). Congestion control mechanisms in wireless sensor networks. Journal of network and computer applications, vol.52 , pp. 101-115.

نشریه هوش مصنوعی و داده کاوی

تشخیص خطا در شبکه‌های حسگر بی‌سیم خوشه‌بندی شده با استفاده از الگوریتم ژنتیک

علی غفاری*[*] و ساناز نوبهاری

دانشگاه آزاد اسلامی، واحد تبریز، گروه مهندسی کامپیوتر، تبریز، ایران.

ارسال ۲۰۱۴/۱۲/۰۳؛ پذیرش ۲۰۱۵/۰۴/۰۵

چکیده:

شبکه‌های حسگر بی‌سیم از تعداد زیادی گره‌های حسگر تشکیل شده است. این گره‌ها پدیده‌های مختلف محیطی را حس کرده و اطلاعات مرتبط با این پدیده‌ها را به‌صورت گام به گام به ایستگاه پایه یا چاهک ارسال می‌کنند. از طرفی، محدودیت ذاتی این شبکه‌ها، استفاده از اجزای ارزان قیمت در ساخت گره‌ها و توزیع آنها در مکان‌های غیرقابل دسترس و غیرقابل کنترل، باعث شده است که این گره‌ها از استعداد خرابی بالایی برخوردار باشند. بنابراین، حفظ عملکرد شبکه با قابلیت‌های مناسب، حتی زمانی‌که اتفاق‌های ناخواسته رخ می‌دهد، برای این شبکه‌ها ضروری می‌باشد که این مهم تحمل‌پذیری خطا نامیده می‌شود. از این‌رو، مدیریت خطا در این شبکه‌ها امری حیاتی و ضروری است. در این مقاله، یک روش جدید کشف خرابی با قابلیت تحمل‌پذیری خطا برای شبکه‌های فوق ارائه می‌شود. اساس روش پیشنهادی رای اکثریت است که به‌طور دائم گره‌های خراب را با دقت بالایی تشخیص می‌دهد. کارایی روش ارائه شده با نرم افزار متلب شبیه‌سازی شده است. برای بررسی کارایی، روش ارائه شده با الگوریتم‌های چن، لی و الگوریتم ترکیبی مقایسه می‌شود. نتایج شبیه‌سازی نشان می‌دهد که عملکرد روش پیشنهادی در پارامترهایی مانند کشف خرابی و نرخ اخطار نادرست حتی در شبکه‌هایی که چگالی گره‌های حسگر معیوب در آنها بالا باشد، بهتر است.

کلمات کلیدی: شبکه‌های حسگر بی‌سیم، تشخیص خطا، الگوریتم خوشه‌بندی، الگوریتم ژنتیک.