



# Fault Tolerance and Energy Efficient Clustering Algorithm in Wireless Sensor Networks: FTEC

Somaye Jafarali Jassbi<sup>1</sup> · Elham Moridi<sup>1</sup>

Published online: 1 April 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Wireless sensor networks are employed to monitor physical areas in different places. Fault tolerance and energy efficiency are main challenges of the networks. WSNs are designed to transmit sensed data to the base station when a part of the network is faulty. In this paper, an algorithm is presented to withstand the challenges. In the suggested algorithm, HEED clustering algorithm and sleep/wake-up method are used for the members of cluster to improve energy consumption. To detect and recover the faults of cluster heads and cluster member nodes, a node is selected as backup cluster head. Also, weighted median is employed to detect and recover the faults of cluster nodes. To recover a fault, the faulty node is isolated and replaced by a neighboring node going from sleep mode to wake-up mode. On comparing the proposed algorithm with HEED and DFD algorithms, the results of simulation reveal that energy consumption is improved and the rate of received correct data, detection accuracy and survival nodes are increase and false alarm rate are decrees.

**Keywords** Wireless sensor networks · Clustering · HEED · Fault tolerance · Weighted median

## 1 Introduction

In the recent years, rapid development of wireless communication technology has accelerated the growth of wireless sensor networks. WSNs are consisted of a set of sensors scattered in a wide geographical area. The networks are used for weather forecasting, detecting and supervising targets, agriculture, monitoring enemy's behavior and geographical position, monitoring wildfires, etc. [1]. Sensors are the most important part of the network and work independently without human intervention. These sensors are very small in size and have limitations on battery level, processing and memory, but energy is the most critical factor in increasing network lifetime. It is expected that

---

✉ Somaye Jafarali Jassbi  
sjassbi@gmail.com

Elham Moridi  
elham.moridi@srbiau.ac.ir

<sup>1</sup> Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

sensors operate independently in various areas for a long period of time and may not be accessible for maintenance and battery replacement due to their physical situation. Therefore, it is necessary to use fault-tolerant methods that are compatible with the energy efficiency of the sensors [2].

Fault tolerance and energy efficiency are 2 main challenges of WSNs. In the cell phone networks, the energy of the base station, responsible for providing the network with energy, is not limited while in the WSNs, the energy cannot be provided and it is required to be minimized to increase the lifetime of nodes [3]. Two important methods of reducing energy consumption in the sensor network are: clustering nodes and nodes sleep/active strategy. Clustering algorithms are proposed for energy efficiency in a wireless sensor network. Many clustering algorithms have been proposed for energy efficiency in a wireless sensor network [4–7]. In clustering algorithms, energy consumption improves due to the use of fusion and spatial correlation of data. The most important advantage of clustering algorithms is to aggregate data and reduces data sent to the data center, easier network management and scalability. The second method is to improve the energy efficiency of sleep/active, and many researches have been done in this regard [8–11]. In WSNs, it is optimum to turn off nodes having no data to send or receive and switch them on when they contain a new data packet. In this method, the nodes change from active to passive mode alternatively based on the network operation. This is called duty cycle, a fraction of time in which nodes are active during their lifetime. When sensor nodes share an operation, they need to coordinate the sleep and wake periods of time [12].

The key issue of WSNs is that the main consequences of energy shortage and discharge of the batteries of nodes are failure and transmission of incorrect data between nodes. Hence, decreased faults and increased fault tolerance in WSNs result in saving node energy and increased node survivability [13]. Also, the nodes need to tolerate failures and transmit data to the base station due to requirements. Detection and retrieval of faults are 2 major steps of fault tolerance. In the detection stage, the faults are identified through supervising the performance of nodes, then in the second stage, the faults are recovered through redundancy or other methods to avoid failure of network function.

Many approaches have been proposed to improve energy efficiency, but all approaches did not take into consideration the fault tolerance problem, even though WSNs are prone to fault. Therefore in this paper, a fault tolerance and energy efficient clustering algorithm in WSNs is presented to withstand the aforementioned challenges. In FTEC algorithm, we employed HEED algorithm for clustering nodes and selecting CHs. Due to the importance of CHs in aggregating and transmitting data to the base station, detecting and retrieval of faulty nodes are of great importance. In fact, faulty CHs result in transmitting incorrect data and the base station may make wrong decisions. In addition, reclustering is time and energy consuming. Hence, in FTEC, on selecting the main CH, a backup node is chosen for it to increase the fault tolerance of CH. The backup node supervises the performance of CH and copies its data until it reaches the destination. When there is a fault, it is not required to recollect the data, and it can be accessed through the backup node. Regarding cluster members, faulty nodes are identified through weighted median technique and a neighboring node goes from sleep mode to wake-up mode and replaces the faulty node. The rest of the paper is organized as follows. The related work is represented in Sect. 2. Section 3 includes the hypotheses of the study. HEED algorithm and the proposed method are discussed in Sect. 4, respectively. Section 5 includes the results of the simulation and evaluation of the proposed protocol. Section 6 concludes the paper.

## 2 Related Works

In this section, some of the cluster-based fault-tolerant algorithms that are consistent with energy efficiency are reviewed.

In [14], a fault retrieval architecture and fault tolerant framework in WSNs are represented. The main 3 steps include fault identification, detection, and retrieval. In the detection step, a neighborhood table is created for each node, then the nodes start evaluating. Next, the nodes compare the average neighbors' value with their own table values to perform required evaluation for detecting the fault of node. Moreover, the residual energy of nodes is used to predict the fault. Nodes whose energy level is lower than a threshold report their energy value to other nodes through a packet and then seek a node with higher energy level.

In the method suggested in [15], an organizer node and the CH are automatically chosen by the base station. The selection mechanism of the organizer node and the CH is based on power, efficiency, and energy. If the CH fails, the organizer node seeks to select a new CH. The used channel access method is TDMA in which each node transmits data in a specific succession using its own time slot. Whenever a packet is sent, the nodes of the cluster transmit their residual energy to the CH. On aggregating data, the CH employs CSMA method to send the data to the base station to avoid the collision of packets. Finally, the organizer node seeks to choose a CH with higher residual energy level.

In [16], a distributed algorithm for energy efficient and fault tolerant routing in WSNs has been represented. To balance the energy consumption of nodes, linear programming has been employed. In linear programming, a linear function is applied to determine maximum and minimum path cost. First, a set of CHs is created, and their distance from the base station is calculated. Second, the linear programming is used to choose the energy of the next CHs to the destination to maximize the network lifetime.

In the algorithm represented in [17], Bayesian modeling approach has been employed for online dynamic event region detection in WSNs. In fact, each node is associated with a virtual community and a trust index, which measures the trustworthiness of the node in its community. The trust value of the node ranges between 0 and 1. A trust value smaller than a threshold indicates that the node is faulty and its sensor reading cannot be trusted. Using a particle filter algorithm, the trust value of node is updated upon the arrival of new observations.

In [18], an algorithm is suggested for improving HEED algorithm to increase the reliability of network and routing fault tolerance. Clustering is performed using HEED algorithm and the CH is chosen based on the most residual energy and the distance to the base station. In fact, longer distance to the base station results in larger clusters and smaller number of CHs. To distribute the CHs in the network, they employed Gabriel graph. According to this graph, some paths are created between the origin and destination and then the best one for transmitting data is selected.

The performance of the algorithm suggested in [19] is similar to that of LEACH protocol, but the CH is provided with a backup to increase the fault tolerance of the CH and decrease the energy consumption of the network. Due to the similarity, it is needed to choose a backup for the CH in each round. The nodes transmit their data to the main CH. Whenever the data is transmitted from cluster members to the CH, the backup node checks the nodes performance based on the Beacon message received from the CH. On completing 3 rounds, if no responses are received, the backup advertises the failure of the CH and the cluster members are required to send data to the backup CH.

In [20], IDFCA algorithm has been represented to increase fault tolerance and network lifetime. The node with the highest residual energy level is chosen as CH. Also, cost function is used to determine distance. In the initial phase, booting process starts and unique identifiers are assigned to nodes and gates. The algorithm sends a warning message and saves the data such as residual energy, the distance to the base station, and identifier. Whenever the CH is faulty, the next CH starts to send data to the Base station. Since the preliminary structure of the network is hierarchical, the CH identifies the cluster members and reclustering is not needed.

All reviewed research studies on fault tolerance algorithms are illustrated in Table 1. They are represented based on fault tolerance method, fault tolerance steps, and major features achieved. Most of the research is only to detect or recover a fault in the cluster head. But in this paper, in addition to detecting the fault in the cluster members; we will identify and recover the fault in the cluster heads.

### 3 Network Model and Basic Concepts of the Proposed Method

In this section, the model of network and the pattern of nodes distribution are discussed. Section 3.1 includes defining nodes and the hierarchical relationship between them. In addition, the energy consumption model of the proposed method is discussed. Finally, we address HEED algorithm, which we seek to increase its fault tolerance.

#### 3.1 Network Model

The sensor network of the proposed model is homogeneous. In other words, all nodes have uniform characteristics, hardware, and process capabilities. The network model is

**Table 1** Fault tolerant clustering based mechanisms

Schemes	Fault tolerance method	Fault tolerance steps		Major features achieved
		Detection	Recovery	
[14]	Neighborhood table	√	√	(1) Energy efficiency (2) Fault tolerance nodes
[15]	Residual energy	√	√	(1) Energy efficiency (2) Fault tolerance cluster head (3) Avoid the collision
[16]	Linear programming	√	–	(1) Energy efficiency (2) Fault tolerance routing
[17]	Bayesian modeling	√	–	(1) Energy efficiency (2) Decrease latency
[18]	Residual energy	√	√	(1) Energy efficiency (2) Fault tolerance cluster head
[19]	Backup node	√	√	(1) Energy efficiency (2) Fault tolerance cluster head
[20]	Cost function	√	–	(1) Energy efficiency (2) Fault tolerance cluster head

illustrated in Fig. 1. The distribution of sensor nodes in the selected environment is random and uniform.

$N$  indicates the total number of network nodes, and the members include  $S_i = \{S_1, S_2, S_3, \dots, S_N\}$ . Each node is one these operation modes:

- *Wakeup* The node is active and communicate with other sensor nodes.
- *Sleep* The sensor is not capable of transmitting data. On receiving a proper signal from the base station, the node becomes active. When the node is in sleep node, it needs less energy.
- *Turnoff* The node is completely turned off.

In FTEC, the network model is hierarchical and clustering. Generally, clustering is aimed at scalability, improved energy efficiency, and longer lifetime in WSNs. Each cluster contains a head, called cluster head, to combine and aggregate data. Also, there are some regular nodes in each cluster. Non-cluster head sensor nodes are connected to the CH. The CH collects the data from cluster members. Each CH can transmit collected data to other CHs. In the proposed method, each CH has a backup node, called BCH, saving a copy of the data of CH. Also, there is a base station, which receives the data collected by the CH. The data can be directly transmitted to the base station or through other CHs. The network model is illustrated in Fig. 1.

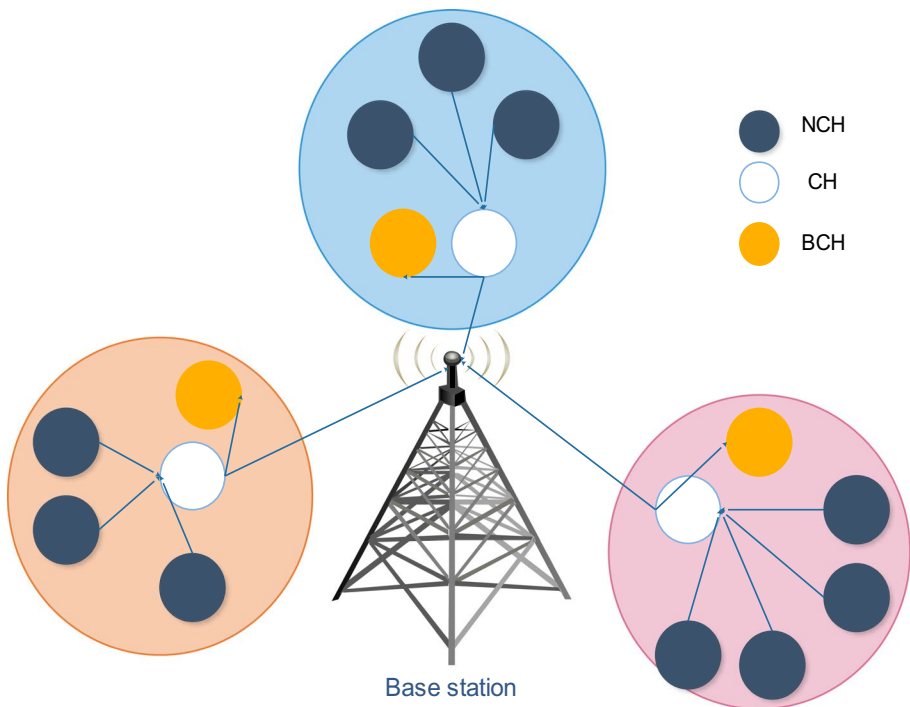


Fig. 1 The network topology model

### 3.2 Energy Model

We employed the radio model suggested in [21]. In this model, distance plays an important role to calculate energy consumption in multipath fading channel and free space. When the distance between transmitter and receiver is less than a threshold value,  $d_0$ , the free space model is employed, otherwise, the multipath model is employed. The energy needed to transmit an 1-bit message over a distance  $d$  is calculated through Eq. (1).

$$E_T(l, d) = \left\{ \begin{array}{ll} lE_{elec} + l_{\epsilon_{fs}} d^2 & d < d_0 \\ lE_{elec} + l_{\epsilon_{mp}} d^4 & d \geq d_0 \end{array} \right\} \quad (1)$$

$E_{elec}$  indicates the required energy to transmit one bit message over the transmitter–receiver circuit and depends on digital coding, modulation, filtering, and spreading of the signal.

The energy required by the amplifier in the free space and multipath is indicated as  $\epsilon_{mp}$  and  $\epsilon_{fs}$ , respectively. The energy of the amplifier depends on the distance between receiver and transmitter and on the acceptable bit rate fault. Also, on receiving data by a node, the consumed energy is calculated through Eq. (2).

$$E_R(l) = lE_{elec} \quad (2)$$

### 3.3 Hybrid Energy-Efficient Distributed Clustering (HEED) Algorithm

HEED algorithm has been used to cluster nodes [22]. To distribute load and power to nodes uniformly, we considered the same average number of nodes for each cluster. In addition, the average distance between cluster members should be the least. The different phases of HEED algorithm to choose the CH are as follows.

Initial phase: in this phase, the algorithm assigns equal probability to each node for being selected as a CH. The probability, called  $C_{prob}$ , is applied to limit the preliminary CHs. Equation (3) is used to determine the probability of selecting a sensor node as a cluster head.

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{Max}} \quad (3)$$

$E_{residual}$  and  $E_{Max}$  indicate the residual energy and maximum energy of node, respectively. When the battery of node is full, its energy is maximum.  $C_{prob}$  is not allowed to be lower than a threshold,  $P_{min}$ .

The residual energy of  $n$  neighboring node with  $E_{jr}$  is calculated as follows.

$$E_{residual} = \frac{1}{n} \times \sum_{j=1}^n E_{jr} \quad (4)$$

*Repetition phase* In this phase, each node repeats the algorithm to find a CH needing the least energy to communicate. When a node can receive no messages from any CHs, it selects itself as a CH and advertises it to its neighbors. Finally, each sensor doubles  $CH_{prob}$  and starts the second repeat. The phase is continued until  $CH_{prob}$  equals 1. Thus, there are two states that the CH can advertise to its neighbors.

*Temporary state* A sensor node changes into a temporary CH when the  $CH_{\text{prob}}$  is lower than 1. When the node repeats the algorithm, it can change into an NCH; in case it finds a CH with lower cost. A sensor can become a permanent CH when  $CH_{\text{prob}}$  equals 1 [23].

*Final phase* In this phase each sensor makes a final decision on whether it is possible to find a CH needing the least energy to communicate, otherwise, it advertises itself as a CH. The chosen CH reports its new role to other nodes of the network. In the proposed method, to improve energy efficiency, when the base station determines that the energy of a CH and its backup are completely consumed, reclustering with HEED is performed.

## 4 Proposed Method

In this section, the details of the proposed algorithm, FTEC, to improve energy consumption and increase fault tolerance in WSNs is discussed.

### 4.1 Choosing Sleep/Wakeup State for Nodes

In the proposed method, on clustering, all cluster nodes with overlapping sensing areas are identified to improve energy consumption. Also, the distance between neighboring nodes is estimated based on the power of input signals. The relative coordinate of neighboring nodes is determined through their data exchange and then is transmitted to the base station. In these techniques, the nodes are provided with a sleep schedule so that only a subset of nodes is active at a time and others are in sleep mode [24]. Figure 2 illustrates the overlapping sensing areas of nodes 1, 2, and 3. The nodes with overlapping sensing areas are in sleep mode. Node 2, which has the most overlapping area with nodes 1 and 3 ( $R_1 \cap R_3 \cong R_2$ ), is in sleep mode.

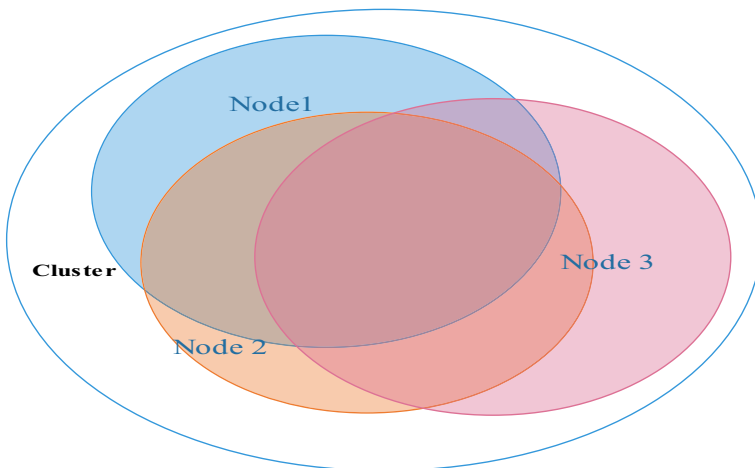


Fig. 2 Radio coverage of nodes

## 4.2 Choosing BCH in FTEC

Two main steps of fault tolerance are fault detection and retrieval. The most common fault detection mechanism is supervising the performance of nodes. In FTEC, on clustering nodes by HEED protocol and selecting CH, a backup is determined for the CH. Two main parameters, distance to the CH and residual energy, are considered to choose BCH. Equation (5) is used to choose BCH.

$$(for\ i = 1\ to\ m)BCH_i = Select\{ (MinDist(S_{NCH,Neighbor}, S_{CH})\ and\ \max\ E_{i,residual}) \} \quad (5)$$

where  $BCH_i$ ,  $m$ ,  $E_{i,residual}$ , and  $Dist(S_{NCH,Neighbor}, S_{CH})$  denote backup cluster head, number of cluster members, the maximum energy of neighboring nodes, and the distance between CH and node, respectively.

$$Dist(S_i, S_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (6)$$

The node containing the most residual energy and the least distance to CH is the most suitable one to be chosen as BCH [25].

Whenever the data is transmitted from cluster members to the CH, the backup node checks the nodes performance based on the Beacon message, received from the CH. Since Beacon packs are much smaller than data packs, the required energy to investigate the CH performance is neglect able. If BCH do not receive any responses from CH in 3 rounds, it reports that the fault of CH is determined. Then BCH is selected as a new CH and can choose a backup from NCHs [26]. Next, the data of cluster members is delivered to the new CH. In addition, whenever the data is transmitted from the CH to the base station, a copy of it is saved in BCH so that in case of any sudden failure of the CH, it is not needed to recollect the data.

## 4.3 Fault Detection and Recovery in FTEC

Here, we detect and isolate faulty nodes to prevent them from affecting the entire network. As mentioned, the NCHs collect their sensed data and transmit to the CH. Since faulty cluster members send incorrect data to the CH, the base station receives wrong data, which results in wrong decision making. Thus, collecting data from non-faulty nodes is significant in WSNs [27]. Due to the fact that there is not a long distance between cluster members, the assumption is that neighboring nodes sense uniform and correlated data and make similar decisions when sense a failure. As a result, faulty nodes of a cluster produce uncorrelated and different data. In the following part, the steps of data collection are discussed in details [28].

The proposed method provides a simple method to detect the fault of nodes. In this method, the CH seeks to detect faulty NCHs through a weighted median method. In fact, the purpose is to provide a detection system of low complexity. The weighted median is a method of distribution based on the contribution of neighbors. Here, before consulting the base station, each node consults its neighbors to detect its fault. When there is a significant difference between the data sensed by the node and the weighted median of the data of neighboring nodes, the node introduces itself as a faulty node to the CH. Thus, normally, the CH is informed of the fault when the suspicious node and its neighbors



have reached an agreement. This results in decreasing communication messages and saving the energy of the sensor.

Here,  $x_i$  and  $N_i$  denote the measured data of the  $i$ th sensor and the number of its neighbors, respectively. The weight of node  $i$  is indicated by  $\lambda_i$ , a positive integer, indicating the trust value of node  $i$ . Each sensor regularly transmits the measured data to all neighbors. Weighted median method is as follows.

- $x_i$  and  $\lambda_i$  of all neighbors are calculated. First, for all nodes,  $\lambda_i$  is considered as  $\lambda_{\max}$ .
- The measured values of neighbors are listed in ascending order.
- Each measured  $x_j$  is duplicated to the number of the corresponding  $\lambda_j$ .

$$\lambda_j \diamond x_j = \overbrace{(x_j, x_j, x_j, x_j, x_j)}^{\lambda_j \text{ times}} \tag{7}$$

Weighted median,  $\hat{x}_i$ , is obtained through Eq. (8).

$$\hat{x}_i = MED \left\{ \lambda_j \diamond x_j \mid_{j=1}^N \right\} \tag{8}$$

where  $f(x_i, \hat{x}_i)$  is obtained through Eq. (9).

$$f(x_i, \hat{x}_i) = \begin{cases} 1 & \text{if } \frac{x_i - \hat{x}_i}{\hat{x}_i} > \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Also,  $\varepsilon$  indicates a predetermine threshold. In WSNs applications,  $\varepsilon$  is set to the tolerant fault ratio of sensor measurements. If  $f(x_i, \hat{x}_i) = 1$ , the  $i$ th node is faulty and its trust value is decreased by 1, otherwise, it has sensed the data properly. When trust value of a node equals 0, the node is introduced as a faulty node.

**Table 2** Simulation parameters

Parameters	Values
Nodes number	2000
Network size	2000*2000
Sink area	1000*1000
Communication range	100 m
Data packet	2000 bites
Initial energy of sensor nodes	3.0 j
$E_{elec}$	50 nJ/bit
$\varepsilon_{fs}$	10 nJ/bit/m <sup>2</sup>
$d_0$	87.0 m

### 5 Results and Evaluation

In this section, we discussed the results of the simulation of the proposed algorithm. MATLAB R2017a was used to simulate the algorithm. FTEC algorithm has been compared with HEED and DFD [29] algorithms. There is a list of applied simulated parameters in Table 2. The base station is located in the center of nodes. In this simulation, different parameters such as consumed energy, correct data rate, number of survived nodes, and delay, detection accuracy and false alarm were examined.

To provide a more accurate evaluation, the simulation run was carried out with 2 scenarios. In the first scenario, energy, correct data rate, and delay were evaluated using 500–1000 nodes. In the second scenario, we employed the survived nodes over 500 and 1000 rounds. The number of nodes during 500 and 1000 rounds was 100 and 1000, respectively.  $P_{NCH} = 0.03$  and  $P_{NCH} = 0.05$  denote the probability of failure of cluster nodes and CHs, respectively.

Energy consumption is discussed in details in Sect. 3.2. To derive correct data rate, the number of correct data received in the destination is divided by the total number of transmitted packs from the origin. The energy consumption of the proposed method is the total amount of energy used to detect and recover the fault. Equation 10 is the total energy consumption calculated according to the energy model in Sect. 3.2.

$$E_t = E_d + E_r = 2 * (E_t(l, d) + E_r(l)) + \frac{3}{2}(E_t(l, 2d) - 2 * E_t(l, d)) \tag{10}$$

$E_d$  is the energy consumption for fault detection and  $E_r$  is total energy required for fault recovery.

Also, the number of survived nodes is the ratio of the existing nodes to the original number of nodes. Equation 11 is used to calculate the number of surviving nodes per run.

$$\text{Survived nodes} = \text{All nodes} - (\text{Faulty nodes} + \text{dead nodes}) \tag{11}$$

The network delay is defined as the consumed time by a packet to move from the origin to the destination. The time needed by the first data packet to arrive at the destination is calculated from the time that data packet is sent by the origin.

Detection accuracy is the ratio of the number of faulty node identified to the total number of faulty nodes. False alarm ratio is the ratio of the number of healthy sensor nodes diagnosed as faulty to the total number of healthy nodes. Assuming that the mean and variance and sensor readings are  $\mu$ ,  $\sigma$ ,  $s_i$  respectively.  $R_{s_i} = \{x_1, x_2, x_3, \dots, x_k\}$  is the dataset of sensor node  $s_i$  during  $T$ . The median and variance are expressed in Eqs. (12), (13).

$$\mu = \begin{cases} R\left[\frac{k}{2}\right] + 1 & k \text{ is odd} \\ \frac{R\left[\frac{k}{2}\right] + R\left[\frac{k}{2}\right] + 1}{2} & k \text{ is even} \end{cases} \tag{12}$$

$$\sigma = \sqrt{\frac{1}{k} \sum_{i=1}^k (x_k^i - \mu)^2} \tag{13}$$

Therefore, Eqs. (14) and (15) are used to calculate the detection accuracy and false alarm. The  $Q$  function is called the standard normal distribution. The threshold value is set by the user.

$$P_{Detection\ accuracy} = Q^{((threshold-\mu-s_i)/\sigma)} \tag{14}$$

$$P_{Fals\ alarm} = Q^{((threshold-\mu)/\sigma)} \tag{15}$$

Figure 3a illustrates the energy consumption of the network for 500 nodes. Due to reclustering in each round, the energy consumption of HEED algorithm is significantly higher than that of the proposed method. Since FTEC algorithm does not need reclustering and benefits sleep/wakeup mechanisms, it requires less energy, and the curve of its energy consumption rate shows a slighter increase compared with that of HEED algorithm. Figure 3b shows that on increasing the number of nodes by 1000, the energy consumption rate of

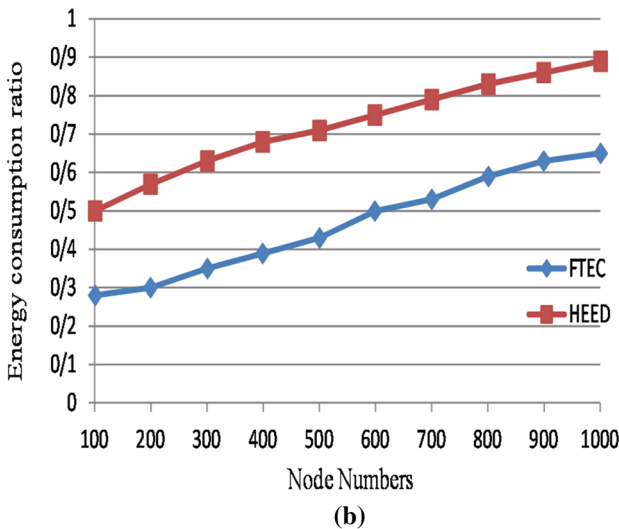
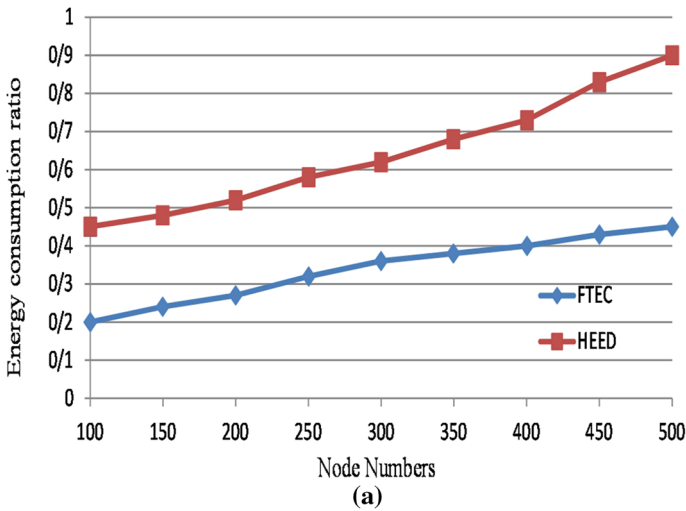


Fig. 3 Energy consumed in various node numbers

HEED algorithm equals 1. In fact, reclustering in each round results in increased overhead and energy consumption, which decreases the network lifetime. On the contrary, the proposed method requires reclustering in case of failure of CH and BCH, which leads to less overhead and energy consumption.

Figure 4a, b illustrate the correct data rate received at the destination for 500 and 1000 nodes. As shown in Fig. 4a, the proposed method benefits from detection and retrieval mechanism and does not send faulty data to the base station so that the rate of correct data is higher compared with that of HEED algorithm. In Fig. 4b, the number of nodes has

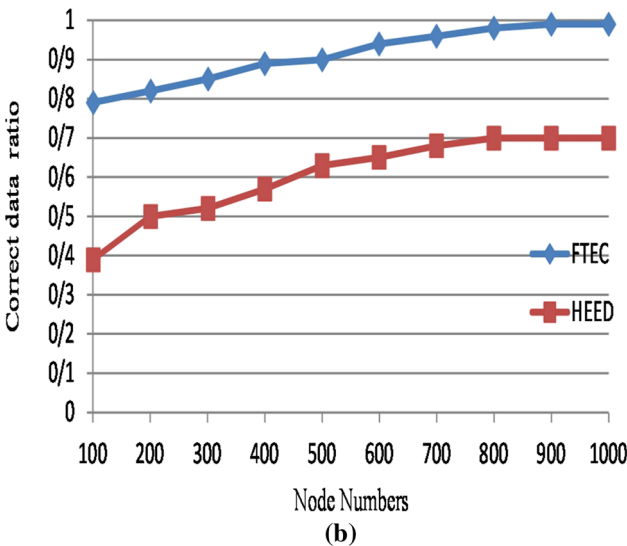
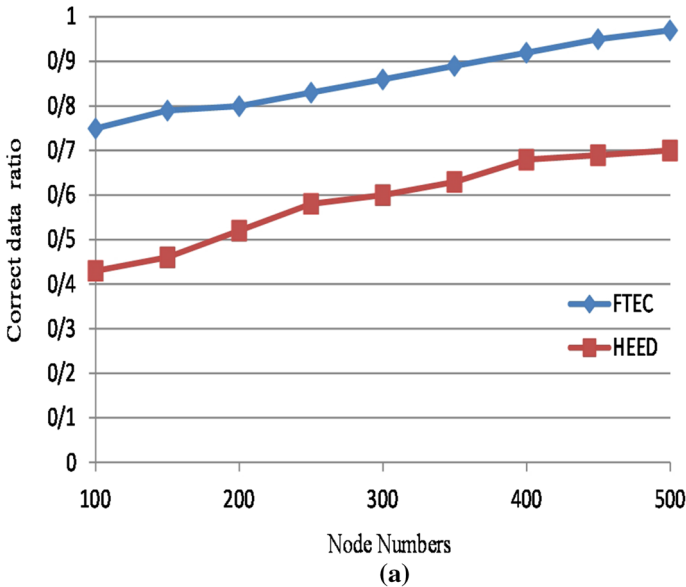
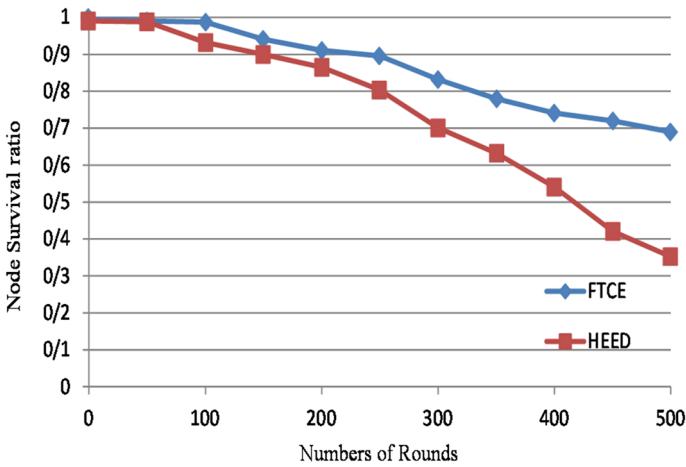


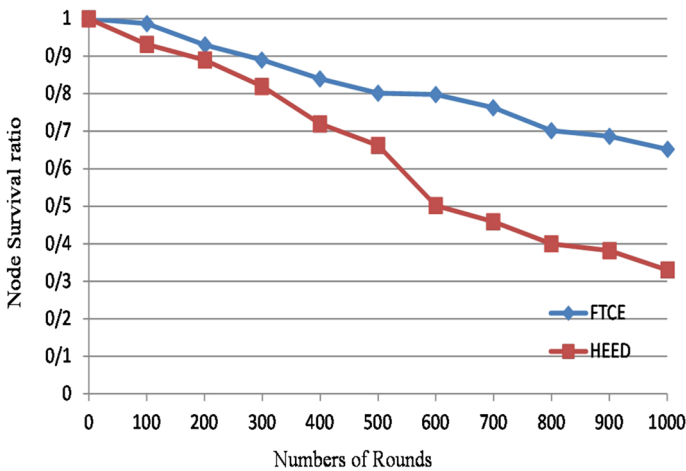
Fig. 4 Average correct data in various node numbers

increased to 1000 and the correct data rate has improved by 40% more than that of HEED algorithm. In other words, although the number of nodes has increased, fault detection mechanism is applies for all clusters, and faulty data is not transmitted to the destination.

Figure 5a, b illustrate the number of survived nodes in each round. Figure 5a displays the number of survived nodes over 500 and 1000 rounds in the proposed and HEED algorithm. On decreasing the number of nodes, HEED selects more CHs so that overhead and energy consumption are increased. Also, HEED algorithm is not provided with any methods to improve nodes energy consumption and the average number of residual nodes is less compared with that of the proposed method. As shown in Fig. 5b, by increasing the number of rounds up to 1000 rounds in the proposed method, the average number of survived nodes is still greater than that of HEED algorithm. In the proposed method, in addition

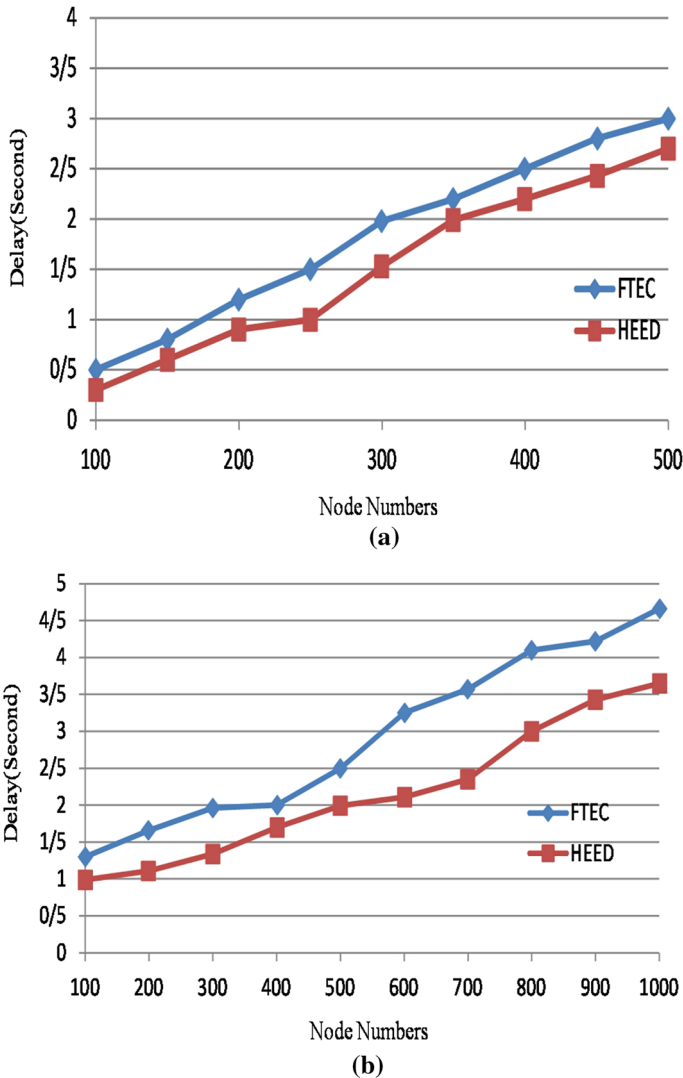


(a)



(b)

Fig. 5 Node survival ratio in various numbers of rounds



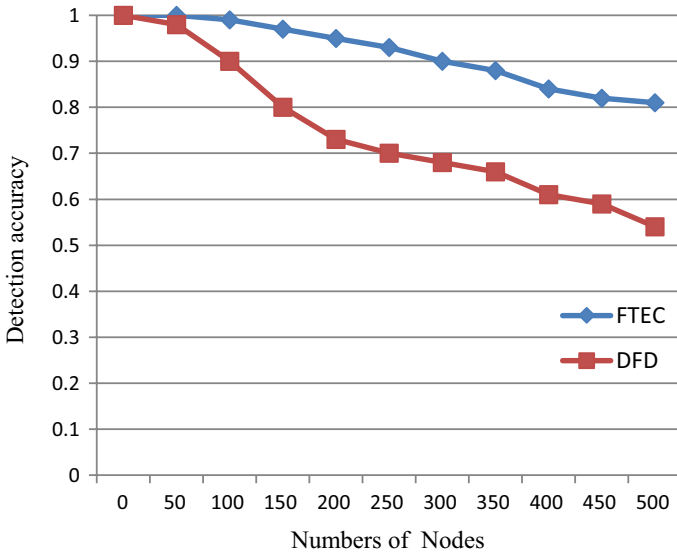
**Fig. 6** Delay in various node numbers

to sleep mode, which results in decreased energy consumption, reclustering is limited to when the CH and BCH are faulty.

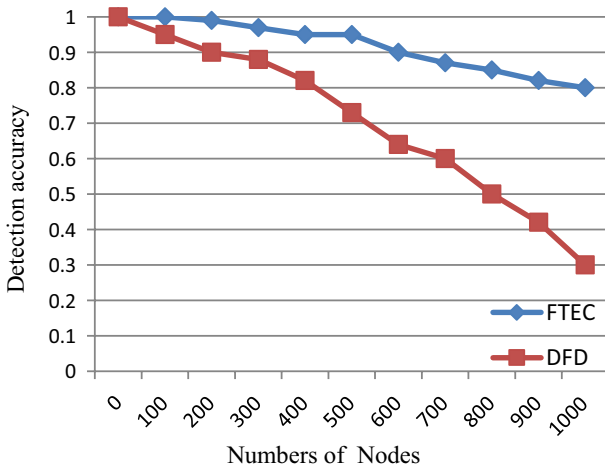
Figure 6a, b display the overall delay of the proposed algorithm according to various number of nodes. Delay is one of the service quality parameters of the network. A long delay results in a congested network. Due to applying fault tolerance mechanisms for the CH and cluster members, the proposed algorithm experiences a longer delay compared with HEED algorithm. HEED algorithm is not provided with any fault tolerance mechanism so that it saves time. As shown in Fig. 6a, when the number of nodes is small, there is a slight difference between the delay of the proposed algorithm and that of HEED. In fact, the failure of the CH or cluster members does not require

reclustering and retransmission, which are time consuming and cause delay. However, increased number of nodes (Fig. 6b) leads to longer delay compared with HEED algorithm. In other words, increased number of nodes causes time consuming fault detection and wakeup mode, which results in longer delay.

Figure 7a, b display the detection accuracy of the proposed algorithm according to various number of nodes. In the proposed method, due to the use of majority voting, detection accuracy is very high in fault detection. As shown in Fig. 7a, b, the proposed method reduces accuracy with increasing number of nodes. Figure 7a, b, show



(a)

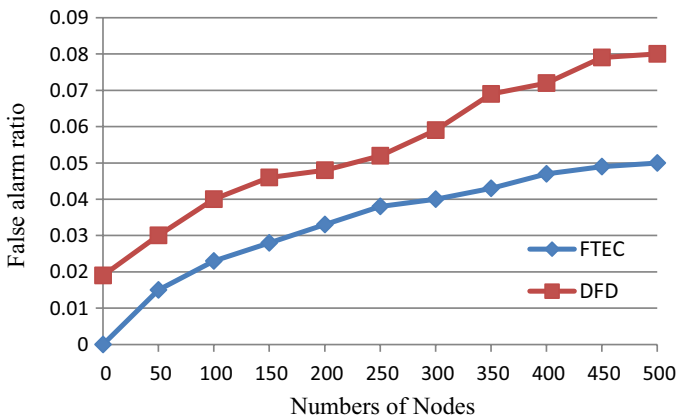


(b)

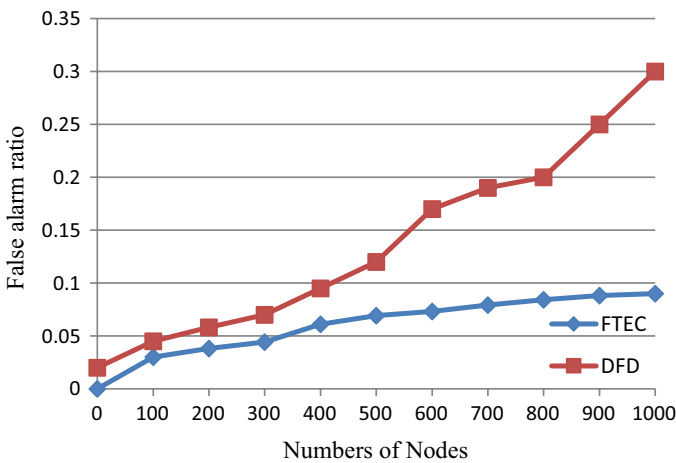
Fig. 7 Detection accuracy in various node numbers

that when the number of nodes is small and the probability of faulty node is large, the DFD algorithm performance will decrease sharply because the DFD algorithm judges too harshly if the node is normal condition. Because the number of nodes increases, the node's data is increased and their evaluation becomes harder to find a fault. Therefore, cooperation improves fault detection accuracy.

Figure 8a, b display the false alarm ratio of the proposed algorithm according to various number of nodes. As the number of nodes increases, false alarms also increase. In the DFD algorithm, the fault detection is by comparing the data of each node with its neighbors. Therefore, its false alarm rate is much higher than the proposed method. But in the proposed method, due to the use of a weighted median in fault detection, the false alarms are less.



(a)



(b)

Fig. 8 False alarm in various node numbers



## 6 Conclusion

We proposed FTEC algorithm to improve clustering-based energy efficiency and fault tolerance. We used HEED algorithm to cluster nodes and select a CH. Due to the importance of CHs and to increase the fault tolerance of them, a BCH is chosen. The BCH supervises the performance of the CH and copies its data. On collecting data, the CH detects and isolates faulty nodes using weighted median. Thus, the fault cannot be dispersed to higher network levels. Next, to recover a fault, the faulty node is replaced by a neighboring node going from sleep mode to wake-up mode through a message from the base station. The results revealed that FTEC algorithm significantly improved energy consumption and increased correct data rate and average number of survived nodes compared with HEED algorithm. In addition, the results of the comparison of the proposed algorithm with the DFD algorithm show that the accuracy of detection and false alarms respectively increased and decreased.

## References

1. Yetgin, H., et al. (2017). A survey of network lifetime maximization techniques in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 19(2), 828–854.
2. Rault, T., Bouabdallah, A., & Challal, Y. J. C. N. (2014). Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67, 104–122.
3. Erol-Kantarci, M., & Mouftah, H. T. (2015). Energy-efficient information and communication infrastructures in the smart grid: A survey on interactions and open issues. *IEEE Communications Surveys and Tutorials*, 17(1), 179–197.
4. Jorio, A., & Elbhiri, B. (2018). An energy-efficient clustering algorithm based on residual energy for wireless sensor network. In *2018 renewable energies, power systems & green inclusive economy (REPS-GIE)*. IEEE.
5. Leu, J.-S., et al. (2015). Energy efficient clustering scheme for prolonging the lifetime of wireless sensor network with isolated nodes. *IEEE Communications Letters*, 19(2), 259–262.
6. Nayak, P., & Devulapalli, A. (2016). A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime. *IEEE Sensors Journal*, 16(1), 137–144.
7. Sharma, G., & Sahni, V. (2018). Management, and technology, performance analysis of energy efficient clustering protocol using tabu in WSN. 15.
8. Cheng, L., et al. (2018). Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks. *Computer Networks*, 134, 66–77.
9. Dong, M., Ota, K., & Liu, A. (2016). RMER: Reliable and energy-efficient data collection for large-scale wireless sensor networks. *IEEE Internet of Things Journal*, 3(4), 511–519.
10. Kumar, V. et al. (2018) Optimal clustering in weibull distributed WSNs based on realistic energy dissipation model. In *Progress in computing, analytics and networking* (pp. 61–73). Berlin: Springer.
11. Tan, N. D., & Viet, N. D. (2015). SSTBC: Sleep scheduled and tree-based clustering routing protocol for energy-efficient in wireless sensor networks. In *2015 IEEE RIVF international conference on computing and communication technologies-research, innovation, and vision for the future (RIVF)*. IEEE.
12. Swain, R. R., Khilar, P. M., & Bhoi, S. K. (2018). Heterogeneous fault diagnosis for wireless sensor networks. *Ad Hoc Networks*, 69, 15–37.
13. Won, J., & Bertino, E. (2015). Distance-based trustworthiness assessment for sensors in wireless sensor networks. In *International conference on network and system security*. Berlin: Springer.
14. Mitra, S., & Das, A. (2017). Distributed fault tolerant architecture for wireless sensor network. *Informatica*, 41, 47–59.
15. Jamjoom, M. M. (2017). EEBFTC: Extended energy balanced with fault tolerance capability protocol for WSN. *International Journal Of Advanced Computer Science And Applications*, 8(1), 253–258.
16. Azharuddin, M., & Jana, P. K. (2015). A distributed algorithm for energy efficient and fault tolerant routing in wireless sensor networks. *Wireless Networks*, 21(1), 251–267.
17. Wang, J., Liu, B. (2017). Online fault-tolerant dynamic event region detection in sensor networks via trust model. In *Wireless communications and networking conference (WCNC)*. IEEE.
18. Zhou, Y., et al. (2016). Fault-tolerant multi-path routing protocol for WSN based on HEED. *International Journal of Sensor Networks*, 20(1), 37–45.

19. Qiu, M., et al. (2013). Informer homed routing fault tolerance mechanism for wireless sensor networks. *Journal of Systems Architecture*, 59(4–5), 260–270.
20. Kaur, M., & Garg, P. (2016). Improved distributed fault tolerant clustering algorithm for fault tolerance in wsn. In *2016 international conference on micro-electronics and telecommunication engineering (ICMETE)*. IEEE.
21. Heinzelman, W.R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences, 2000*. IEEE.
22. Younis, O., & Fahmy, S. (2004). HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4), 366–379.
23. Yang, M., Yang, D., & Huang, C. (2012). An improved HEED clustering algorithm for wireless sensor network. *Journal of Chongqing University*, 35, 101–106.
24. Chen, Y., & Son, S. H. (2005). A fault tolerant topology control in wireless sensor networks. In *The 3rd ACS/IEEE international conference on computer systems and applications*, 2005. IEEE.
25. Gupta, S.K., Kuila, P., & Jana, P.K. (2016). Energy efficient multipath routing for wireless sensor networks: A genetic algorithm approach. In *2016 international conference on advances in computing, communications and informatics (ICACCI)*. 2016. IEEE.
26. Cheraghlou, M. N., Khadem-Zadeh, A., & Haghparast, M. (2017). Increasing lifetime and fault tolerance capability in wireless sensor networks by providing a novel management framework. *Wireless Personal Communications*, 92(2), 603–622.
27. Gao, J.-L., Xu, Y.-J., & Li, X.-W. (2007). Weighted-median based distributed fault detection for wireless sensor networks. *Journal of Software*, 18(5), 1208–1217.
28. Vieira, T.P., Almeida, P. E., & Meireles, M. R. (2017). Intelligent fault management system for wireless sensor networks with reduction of power consumption. In *2017 IEEE 26th international symposium on industrial electronics (ISIE)*. IEEE.
29. Jiang, P. J. S. (2009). A new method for node fault detection in wireless sensor networks. *Sensors*, 9(2), 1282–1294.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Somaye Jafarali Jassbi** was born in Tehran, Iran, in 1983. She received the B.E. degree in computer architecture engineering in 2007, and the Ph.D. degrees computer architecture engineering in 2010 from the Azad University Science and Research branch, Tehran, Iran. In 2010, she joined the Department of computer Engineering, Azad University science and research branch, she became an Associate Professor in 2011. Her current research interests include Cloud computing, Internet of Thing (IOT), Wireless sensor network and computer architecture. She was head of computer department in 2012. She write, translate and published several professional books and papers in her fields.



**Elham Moridi** received B.Sc degree in Information technology (IT) Engineering from Dezful Branch, Payame Noor Universtiy, Dezful, Iran, in 2011. She received her M.S. degree in Computer Systems Architecture Engineering from Islamic Azad University, Dezful Branch, Iran in 2015. She is currently working toward the Ph.D degree in Computer Hardware Engineering at Department of Computer Engineering in Islamic Azad University Science and Research Branch, Tehran. Her Research interests include Wireless Sensor networks and Vehicle Ad hoc networks.