

Malicious Node Detection Using Confidence Level Evaluation in a Grid-Based Wireless Sensor Network

Min-Cheol Shin, Yoon-Hwa Choi

Department of Computer Engineering, Hongik University, Seoul, Korea

Email: yhchoi@cs.hongik.ac.kr

Received December 19, 2012; revised January 21, 2013; accepted January 28, 2013

ABSTRACT

In this paper, we present a malicious node detection scheme using confidence-level evaluation in a grid-based wireless sensor network. The sensor field is divided into square grids, where sensor nodes in each grid form a cluster with a cluster head. Each cluster head maintains the confidence levels of its member nodes based on their readings and reflects them in decision-making. Two thresholds are used to distinguish between false alarms due to malicious nodes and events. In addition, the center of an event region is estimated, if necessary, to enhance the event and malicious node detection accuracy. Experimental results show that the scheme can achieve high malicious node detection accuracy without sacrificing normal sensor nodes.

Keywords: Sensor Networks; Malicious Node Detection; Grid-Based WSN; Faults; Confidence Levels

1. Introduction

Sensor networks consist of a large number of small sensor nodes with sensing, computation, and wireless communication capabilities to monitor various environments and detect events of interest [1]. Due to the limited resources of sensor nodes, the networks are vulnerable to faults and malicious attacks. Malicious nodes may generate arbitrary reports regardless of the actual sensor readings, leading to an incorrect decision, resulting in reduced network lifetime and loss of network connectivity. Hence it is important to identify malicious nodes in the presence of events and faults and isolate them upon detection.

Several faults, anomaly, or outlier detection schemes for wireless sensor networks have been presented in the literature [2-7]. Most of them focus on detecting faulty sensor nodes or removing anomalous sensor readings in a distributed manner, assuming that majority of the sensor nodes report correctly. Some efforts have also been made to distinguish events from faults by exploiting the notion that measurement errors due to faults are likely to be uncorrelated, while measurements in a target region are spatially correlated [8-13]. In [11] a secure event boundary detection scheme was presented to correctly identify event boundaries in adversarial environments. Event detection using decision tree classifiers running on individual sensor nodes and applying a voting scheme to reach consensus among detections made by various sensor nodes has been proposed for disaster management [12].

In fault, event, or anomaly detection in wireless sensor networks, malicious nodes are often ignored or lightly treated, although they are likely to appear in the networks. In the case where malicious nodes generate arbitrary readings that do not conform to the defined fault model, the resulting performance might be poorer than the estimated one. Moreover, if they behave intelligently, it would be more difficult to detect events in the presence of wrong reports and distinguish events from false alarms due to the malicious nodes.

Several schemes for detecting malicious nodes in wireless sensor networks have been proposed [14-18]. Curiaic *et al.* [14] presented a detection scheme using autoregression technique. In [15] signal strength is used to detect malicious nodes. A message transmission is considered suspicious if the strength is incompatible with the originator's geographical position. Xiao *et al.* developed a mechanism for rating sensors in terms of correlation by exploring Markov Chain [16]. A network voting algorithm is introduced to determined faulty sensor readings. Atakli *et al.* [17] proposed a malicious node detection scheme using weighted trust evaluation for a three-layer hierarchical sensor network. Trust values are employed and updated to identify malicious nodes behaving opposite to the sensor readings. Ju *et al.* [18] presented an improved intrusion detection scheme based on similar weighted trust evaluation. The mistaken ratio of each individual sensor node is used in updating the trust values. Trust management schemes have been proposed in routing and communications [19]. Some efforts have

also been made to combine communication and data trusts [20]. However, malicious node detection in the presence of events and various types of misleading sensor readings due to the compromised nodes have not been deeply investigated. In addition, the problem of distinguishing malicious nodes from events has not sufficiently been taken into account.

Meanwhile, clustering schemes in wireless sensor networks have been investigated for energy efficiency and scalability in routing and data aggregation. Grid-based schemes, where network areas are divided into small grids, have also drawn special attentions due to their simplicity [21-23]. In [24] an energy efficient framework for detecting events in clustered sensor networks was presented. A cellular approach to fault detection and recovery in sensor networks is presented in [25], where a virtual grid structure is used to detect energy-depleted nodes.

In this paper, we present a malicious node detection scheme using confidence level evaluation in a grid-based wireless sensor network. Inter-grid communications are employed, if necessary, to distinguish events from false alarms due to malicious nodes. Confidence levels of member nodes are updated to reflect their behavior in decision-making. The scheme is designed to identify malicious nodes even in the presence of relatively small event regions.

2. Background

In this section, we briefly introduce the network model for our malicious node detection scheme and define the behavior of malicious node to be identified.

2.1. Grid-Based Sensor Networks

Grid-based sensor networks have been proposed for energy efficient data aggregation and routing [21]. Our malicious node detection scheme is developed to conform to the protocol of the hierarchical networks. The sensor field in a grid based sensor network is assumed to be divided into $M \times N$ square-shaped grids as illustrated in **Figure 1**, where there are nine grids, A through I, and l is the side of a grid. Sensor nodes are assumed to be deployed randomly. Each sensor node is also assumed to know its own location. Immediately after deployment, the sensor network carries out grid construction process, and each sensor node figures out the grid it belongs to. Sensor nodes in each grid form a cluster, where a cluster head is selected dynamically. All other nodes in the cluster communicate directly with the cluster head. Two types of communication are defined here for malicious node detection: one for communication between the cluster head and cluster members and the other for communication between neighboring cluster heads.

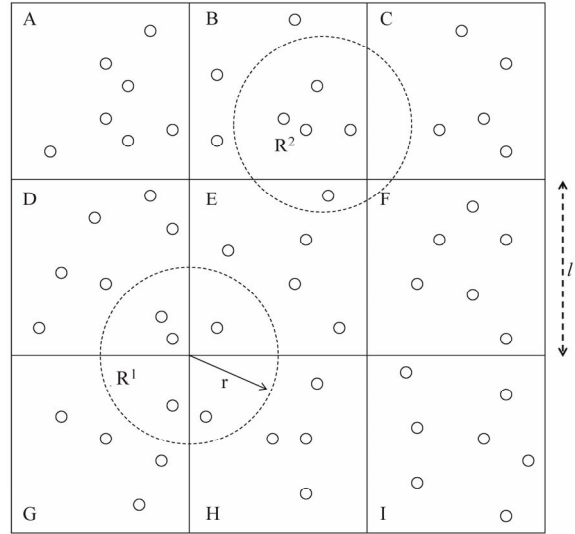


Figure 1. A sensor network with nine grids.

The decision made at a cluster head alone based on the sensor readings of its member nodes might not be accurate due to the difficulty in distinguishing between false alarms and events, especially for a relatively small event region located across multiple grids as illustrated in **Figure 1** (see R_1). Each grid, in that case, has insufficient number of event-nodes to apply a threshold test, such as the well-known majority voting. Consequently, lowering the threshold might be needed to achieve high event detection performance, causing a considerably high false alarm rate, unless the number of malicious nodes is negligibly small.

In order to cope with the expected poor performance, we estimate the event region, if necessary, with inter-grid communication, by finding the center of the nodes reporting an alarm, and then apply a threshold test to the estimated event region.

2.2. Modeling Malicious Nodes

In this paper, we assume that each sensor node is aware of the range of normal readings. For clarity, we name acceptable sensor data in case of no-event as “normal” readings. Any readings outside the normal range are called “unusual” readings for convenience. In other words, correct sensor readings in an event region are also called unusual readings. Hence each sensor node can make a binary decision on its own sensor reading, where a “1” indicates an unusual reading. Sensor nodes in an event region are expected to report a 1, unless the nodes are faulty.

We also assume that malicious nodes can change the sensor readings arbitrarily. In addition, they have some intelligence to report 0’s and 1’s alternately, to break down the network while remaining undetected, unless some sophisticated techniques are used to detect them.

In order to detect malicious nodes, we define a model for their behavior. We assume that all the sensor nodes become malicious randomly and independently with the same probability p_m . In addition, each malicious node sends its report inconsistent with the actual sensor reading with the probability p_{ma} . If $p_{ma} = 0.4$, for example, malicious nodes report 1(0) with a probability of 0.4 when the actual reading is 0(1).

In addition, normal sensor nodes in the network are also assumed to report against their readings, randomly and independently, with the same probability p_i . Hence malicious nodes have to be detected and isolated in the presence of such faults and events.

2.3. Event Model

The most important part in detecting malicious nodes is how to distinguish false alarms due to malicious nodes from events and identify malicious nodes in an event region. We thus define the event model to be used throughout the paper. An event region is assumed to be a circle with radius r , although the proposed scheme can be applied to event regions of other shapes with minor modifications.

In selecting a threshold for event detection in the face of faults and malicious nodes, the size of an event region plays an important role. Suppose that the side of a grid is l . Then the average number of sensor nodes in an event region, n_e , can be written as

$$n_e = d \cdot \frac{\pi r^2}{l^2},$$

where d is the average number of sensor nodes in a grid. For a relatively large event region, it is easy to set the threshold since at least one grid is likely to pass the threshold. For a relatively small event region compared to a grid, however, each grid might contain only a small number of event nodes, especially when the region lies across multiple adjacent grids. In that case, choosing a proper threshold is difficult or might be impossible to satisfy both high event detection accuracy and low false alarm rate.

3. Malicious Node Detection

In detecting malicious nodes, we employ confidence levels (weights) of sensor nodes to reflect the trustworthiness of their reports in decision-making. A sensor node with its weight below a preassigned lower bound is determined to be malicious, and it thus is logically isolated from the rest of the network. In addition, the center of an event region is estimated, if necessary, to achieve high event/malicious-node detection performance, while maintaining low false alarm rate.

3.1. Confidence Level

Malicious nodes are assumed to arbitrarily modify their readings without being easily detected. To monitor their behavior we define confidence level of a sensor node to represent its reliability, measuring its past behavior in reporting sensor readings. For a grid with n sensor nodes, v_1, v_2, \dots , and v_n , the cluster head maintains w_1, w_2, \dots , and w_n , as their weights (confidence levels), respectively, where $0 \leq w_i \leq 1$, and updates them each time a decision on the correctness of their reports is made. Initially all the weights are set to 1. At the time the weight reaches a predefined lower bound (0 in this paper), the corresponding node is determined to be malicious and logically isolated thereafter.

3.2. Decision Based on Center Estimation

In a grid-based sensor network, the decision made at each grid alone without inter-grid communication for receiving data from neighboring grids might be inaccurate when a relatively small event region is located across multiple adjacent grids. As the event region increases, however, at least one grid may have sufficient number of event nodes to pass a threshold test, such as the well-known majority voting. To cope with this problem, we apply a threshold test, if needed, to an estimated event region, computed based on the aggregated data obtained from the neighboring cluster heads.

In **Figure 2**, for example, an event region E is located across four grids, A, B, C, and D, such that each grid has insufficient number of event nodes to pass majority voting. The event can possibly be detected if the threshold is lowered. It, however, causes significant false alarms, especially for a fault-prone sensor network with a large number of malicious nodes.

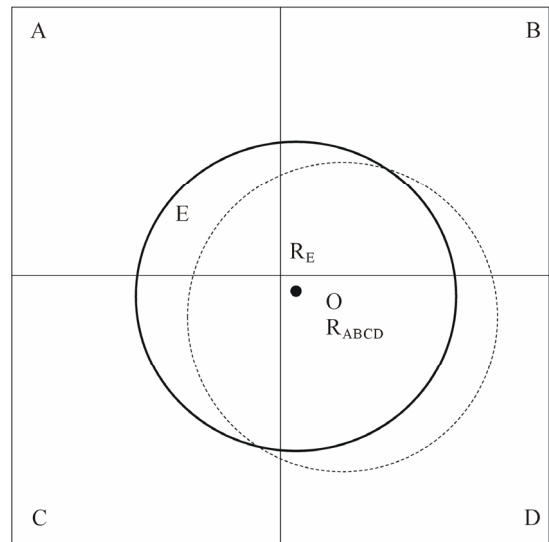


Figure 2. Estimation of the center of an event region.

The center of the alarms in the grid A, R_A , is defined here as the weighted average of the positions, r_i^s , of the nodes reporting a “1” (*i.e.*, an alarm), and it thus can be expressed as

$$R_A = \frac{\sum w_i r_i}{\sum w_i} \quad (1)$$

where each alarm node in the grid A contributes to the estimated center as long as its weight is not zero.

Similarly, the overall center of the alarms from the four grids, R_{ABCD} , can be written as

$$R_{ABCD} = \frac{w_A R_A + w_B R_B + w_C R_C + w_D R_D}{w_A + w_B + w_C + w_D} \quad (2)$$

where w_A represents $\sum w_i$ of the sensor nodes reporting a “1” in the grid A. Once the center is computed, a threshold test, such as weighted majority voting, will be applied to a circle centered at R_{ABCD} with radius \tilde{r} , where $\tilde{r} \leq r$.

3.3. Updating Confidence Levels

In the proposed detection, the decision on an event is made at the cluster head based on two threshold tests to be detailed in the next subsection. Once the decision is made, the cluster head needs to update the confidence levels of its member nodes accordingly. For each member node v_j the cluster head maintains two weights, w_j^1 and w_j^0 , where w_j^1 and w_j^0 represent the weights of v_j in case of no-event and an event, respectively. That is, a malicious node reporting a 1 in a no-event cycle loses its weight w_j^1 , while a malicious node reporting a 0 in an event region loses its weight w_j^0 . The weight w_j defined in the previous subsection is the smaller one between the two, *i.e.*, $w_j = \min(w_j^1, w_j^0)$.

In case of no-event, the cluster head updates the weights as follows.

$$w_j^1 = \min(0, w_j^1 - \alpha) \text{ for } s_j = 1 \quad (3)$$

$$w_j^1 = \max(1, w_j^1 + \beta) \text{ for } s_j = 0 \quad (4)$$

where s_j denotes the sensor reading of node v_j .

Malicious nodes reporting a 1 in the case of no-event lose their weights, w_j^1 , by α . Otherwise, they gain weights by β . The two parameters, α and β play an important role in distinguishing between malicious and normal nodes. If $\alpha = 0.2$ and $\beta = 0.05$, for example, a sensor node reporting a 1 every five cycles recovers its weight to 1.0. That is, for the chosen values of α and β a normal sensor node with some transient faults remain in the network unless the probability p_t is greater than 0.2. Malicious nodes reporting alarms more frequently than this gradually lose their weights, and will eventually be detected at the time the weights reach 0.

In the case of an event, the weights of the nodes within the event region need to be lowered if they have reported a 0. Due to the inaccuracy of the center estimation, however, we apply the updates only to sensor nodes within a circle of radius $\delta \cdot r$ ($\delta \leq 1$) centered at the estimated center, not to sacrifice normal nodes. The following updates are done at the cluster head.

$$w_j^0 = \min(0, w_j^0 - \alpha) \text{ for } s_j = 0 \quad (5)$$

$$w_j^0 = \max(1, w_j^0 + \beta) \text{ for } s_j = 1 \quad (6)$$

A malicious node in an event region loses its weight if it is within the reduced circle. As a result, the detection latency might increase. Such a node, however, can also be identified during no-event cycles if it reports a 1.

3.4. Malicious Node Detection in a Grid-Based WSN

In the proposed scheme, malicious nodes in a grid-based sensor network are detected using threshold tests along with confidence level evaluation. In addition, malicious nodes are distinguished from events by estimating the center of an event, if necessary. Our malicious node detection scheme can be described as follows:

Malicious Node Detection in a Grid-Based WSN

1. Each sensor node v_j sends a 1 (alarm) to the cluster head if $s_j = 1$.

2. Each cluster head computes $M_0 = \sum_{j=1}^d w_j (1 - s_j)$ and

$$M_1 = \sum_{j=1}^d w_j s_j.$$

3. If $\frac{M_1}{M_1 + M_0} > \theta_1$, then $E = 1$ (*i.e.*, an event) and update confidence levels accordingly

If $\frac{M_1}{M_1 + M_0} \leq \theta_1$ and $\frac{M_1}{M_1 + M_0} > \theta_2$ ($\theta_2 \leq \theta_1$), then estimate the center of alarms using inter-grid communication, and apply weighted majority voting to the estimated event region. If $E = 1$, update confidence levels accordingly

If $\frac{M_1}{M_1 + M_0} \leq \theta_2$, then $E = 0$ (*i.e.*, no-event) and update confidence levels accordingly.

4. Determine the nodes with $w_j = 0$ to be malicious.

Inter-grid communication is needed only for the second case in Step 3, where the center of alarms is computed to apply weighted majority voting within the estimated event region. In the simulation later, we choose $\theta_1 = 0.5$ (*i.e.*, majority voting) to make a decision locally in a grid alone. The value of θ_2 , however, has to be carefully chosen to achieve both high event detection accu-

racy and low false alarm rate.

Let P_f represent the probability that the report from a sensor node is incorrect due to faults or malicious attack. Let d denote the average number of nodes in a grid. Then for given l and r , the average number of event nodes in a grid, n_e , when an event region is located across four grids as illustrated in **Figure 1**, is

$$n_e = d \cdot \frac{\pi r^2}{l^2} \cdot \frac{1}{4} \quad (7)$$

At least one grid is likely to have more event nodes than the average in practice, unless the event nodes are equally divided into the four grids. In the case of no event, the average number of alarm nodes in a grid, N_{ne} , is

$$N_{ne} = d \cdot P_f \quad (8)$$

In the case of an event, the average number of alarm nodes, N_e , in a grid is given by

$$N_e = n_e (1 - P_f) + (d - n_e) P_f = d \cdot P_f + n_e (1 - 2P_f) \quad (9)$$

Hence the ratio $\frac{N_e}{d}$ is

$$\frac{N_e}{d} = \left(1 - \frac{1}{2} \left(\frac{\pi r^2}{l^2}\right)\right) P_f + \frac{1}{4} \frac{\pi r^2}{l^2} \quad (10)$$

From the above expressions (8) and (9), we can see that $\frac{N_e}{d}$ is greater than $\frac{N_{ne}}{d} (= P_f)$ until P_f reaches 0.5.

For a sensor network functioning correctly, P_f is expected to be much smaller than 0.5, and we thus assume that P_f lies between 0 and 0.3 for a working sensor network. Since malicious nodes identified are logically isolated from the network, P_f can be controlled to be lower than 0.3 unless a large number of nodes become malicious at the same time.

For $\frac{r}{l} > 0.6$, $\frac{N_e}{d}$ is greater than 0.3 for the entire range of P_f under consideration. Hence setting θ_2 to 0.3 in those cases can remove most of the false alarms while achieving high event detection accuracy. For a relatively small event region, however, it would be necessary to lower θ_2 to maintain high event detection performance. If

$$\frac{r}{l} = 0.5 \text{ and } P_f = 0,$$

for example, $\frac{N_e}{d} = \frac{1}{4} \frac{\pi r^2}{l^2} \approx 0.2$.

Even if P_f increases to 0.1, the ratio is still less than 0.3. Since P_f is unknown and might change over time, we choose the threshold θ_2 to be effective for a wide range of P_f . In this paper, we choose θ_2 to be

$$\theta_2 = \min \left(P_f^u, \frac{1}{4} \left(\frac{\pi r^2}{l^2} \right) \right) \quad (11)$$

where P_f^u denotes an upper bound on P_f for a functioning sensor network. In other words, if the malicious node detection scheme can control P_f below 0.3 by properly isolating malicious nodes upon detection, for example, P_f^u can be set to 0.3, and the resulting

$$\theta_2 = \min \left(0.3, \frac{1}{4} \left(\frac{\pi r^2}{l^2} \right) \right).$$

Lowering θ_2 causes more false alarms, requiring unnecessary inter-grid communication. Our scheme, however, quickly lowers the weights of malicious nodes, and it thus effectively reduces the number of false alarms.

We can extend the proposed malicious node detection scheme to cover malicious cluster-heads by employing spare node(s) in each grid for monitoring the behavior of the cluster-heads [26]. Since spare node(s) in a grid can also receive reports from the member nodes and perform the same function as the cluster head, each report from a malicious cluster head to its neighboring cluster heads or base station can be checked to see if there is any mismatch. Moreover, inter-grid communication to estimate the center of alarms for a relatively small event region allows adjacent cluster heads to compute the center of alarms at the same time. Such a redundancy makes it possible to immediately detect a malicious cluster head since cluster heads can monitor each other's behavior.

4. Simulation Results

Computer simulation is performed in a sensor network where sensor nodes are randomly deployed in a square area. The network area is divided into grids of the same size, each of which has 20 nodes on average. Four metrics, malicious node detection rate (MDR), misdetection rate (MR), event detection accuracy (EDA), and false alarm rate (FAR), are used in the performance evaluation. MDR is defined to be the ratio between the number of detected malicious nodes and the total number of malicious nodes in the network. MR is defined as the ratio of the number of good nodes determined as faulty to the number of good nodes. Event detection is also important since malicious nodes have to distinguished from event nodes. Hence to indicate the accuracy of event detection, EDA is defined to be the number of events detected to the total number of events generated. Finally, FAR is used to denote the ratio of the number of false alarm cycles to total number of no-event cycles operated.

We first evaluate MDR and MR for various values of p_{ma} when $p_t = 0.1$, $p_m = 0.2$, $\alpha = 0.1$, and $\beta = 0.02$ after 50, 100, 300, and 500 cycles of operation. The results are shown in **Figures 3** and **4**, respectively.

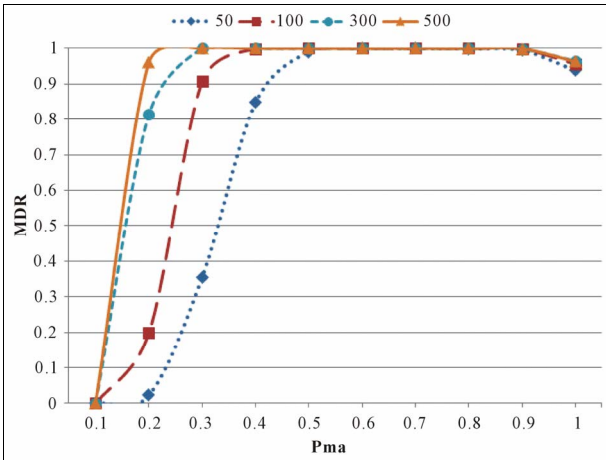


Figure 3. MDR for various values of p_{ma} when $\alpha = 0.1$ and $\beta = 0.02$.

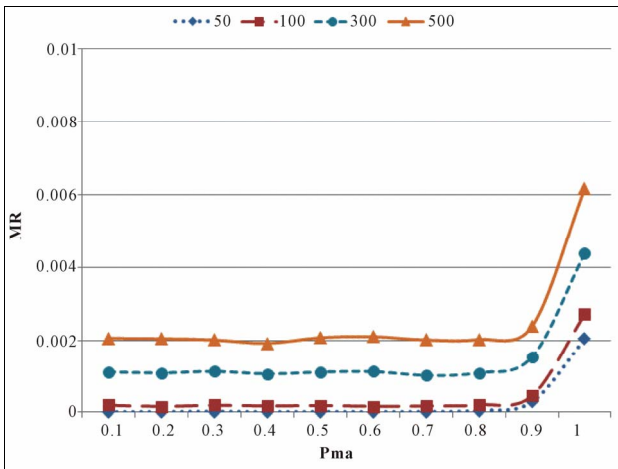


Figure 4. MR for various values of p_{ma} when $\alpha = 0.1$ and $\beta = 0.02$.

MDR for $p_{ma} \geq 0.2$ is almost perfect after 500 cycles. Since $\alpha = 0.1$ and $\beta = 0.02$ are chosen for the simulation, a malicious node reporting a 1 every 6 cycles can still retain its weight. Hence for a smaller value of p_{ma} (e.g., $p_{ma} = 0.05$) malicious nodes remain undetected. Further improvements in MDR can be made by changing the values of α and β unless malicious nodes behave similar to normal nodes (i.e. $p_{ma} \approx 0$). If $p_{ma} = 1.0$, malicious nodes might form a group to pass the threshold tests, although the probability is low, resulting in a small degradation in MDR.

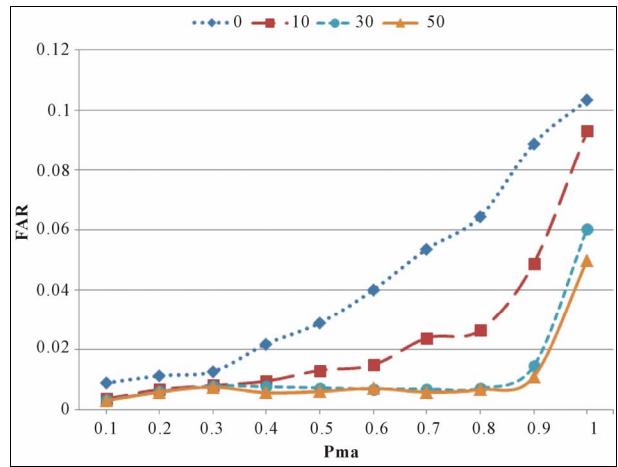
We have also observed that the required time to detect malicious nodes depends on p_{ma} , α , and β . If $p_{ma} = 0.5$, for example, malicious nodes are almost surely detected within 50 cycles. If p_{ma} is 0.2, on the other hand, most of the malicious nodes are detected after 500 cycles.

MR is controlled to be less than 0.002 for a wide range of p_{ma} as shown in Figure 4. It increases to 0.006 when

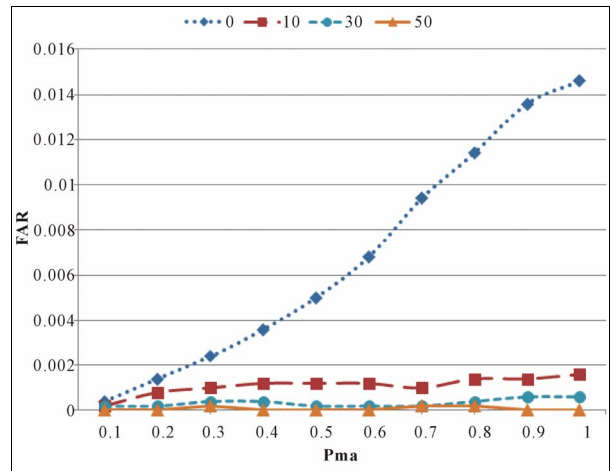
$p_{ma} = 1.0$ due to the false alarms caused by malicious nodes. Although the experiments are conducted for a small event region to see the worst case performance, some notable improvements in MDR and MR can be made as the size of events increases.

False alarms may occur if malicious nodes form a group to pass the thresholds. Such unwanted alarms cause unnecessary communication and computation, and they might shorten the network lifetime. FAR for a grid, when $p_t = 0.1$, $p_m = 0.2$, and $r = 0.5l$, is shown in Figure 5(a), where it increases with p_{ma} . A significant reduction in FAR is made as the number of cycles increases. This is due to the fact that the weights of malicious nodes are lowered with time.

In Figure 5(a), we can observe a sudden increase in FAR even after 50 cycles when $p_{ma} = 1$. In that extreme case, all the malicious nodes send wrong reports to the cluster head. As a result, about 30% of the sensor nodes on average generate an alarm. Such an increase in FAR



(a)



(b)

Figure 5. FAR for a grid for various values of p_{ma} when $r = 0.5l$, (a) $p_t = 0.1$ and $p_m = 0.2$; (b) $p_t = 0.05$ and $p_m = 0.1$.

disappears as p_t and p_m decrease as illustrated in **Figure 5(b)**, where $p_t = 0.05$ and $p_m = 0.1$ are chosen for comparison. Significant reductions in FAR are also observed. Moreover, FAR in that case is stable and persistent regardless of the values of p_{ma} , and is very close to 0 after 50 cycles. Since θ_2 increases with r up to P_f^u , FAR decreases as the size of the event region increases. We conducted the same simulation for $r = 0.7l$ when $p_t = 0.1$ and $p_m = 0.2$. The results are shown in **Figure 6**, where FAR becomes negligibly small with time regardless of the values of p_{ma} .

In malicious node detection, distinguishing malicious nodes from event nodes is also important to achieve high detection accuracy. Moreover, malicious nodes should be detected and isolated without sacrificing EDA. We thus evaluated EDA in the presence of malicious nodes. All the malicious nodes are generated simultaneously at the time the simulation starts. We then generated events at various different cycles to see the impact of weight changes over time. The resulting EDA are shown in **Figure 7**, where a weighted majority voting is applied within a circle with radius $\tilde{r} = 0.7r$. EDA is very close to 1 when $r \geq 0.7l$ as shown in **Figure 7**. High EDA is achieved even for a relatively small event region. In addition, some marginal improvements in EDA are also observed with time.

Finally, we comment on the accuracy of the estimation of an event center and the resulting accuracy in identifying malicious nodes in the corresponding event region. For convenience we first define $D_{centers}$ to be the distance between the center of an event region of radius r and the center of alarms in four adjacent grids, *i.e.*,

$$D_{centers} = \text{dist}(R_{alarms}, R_{event}).$$

$$\frac{D_{centers}}{r},$$

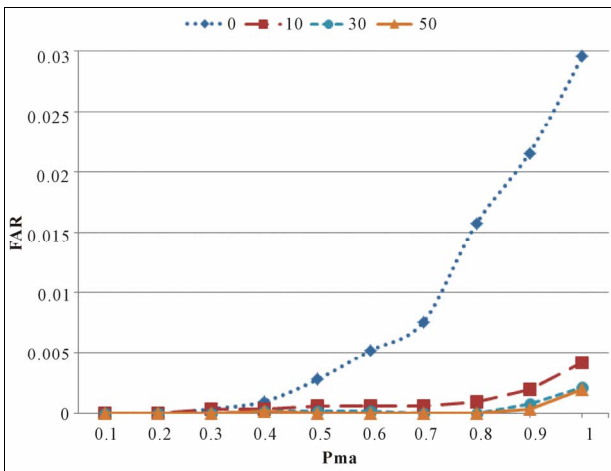
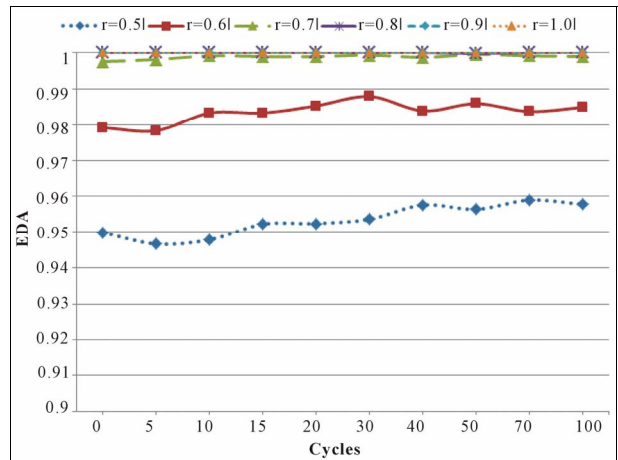


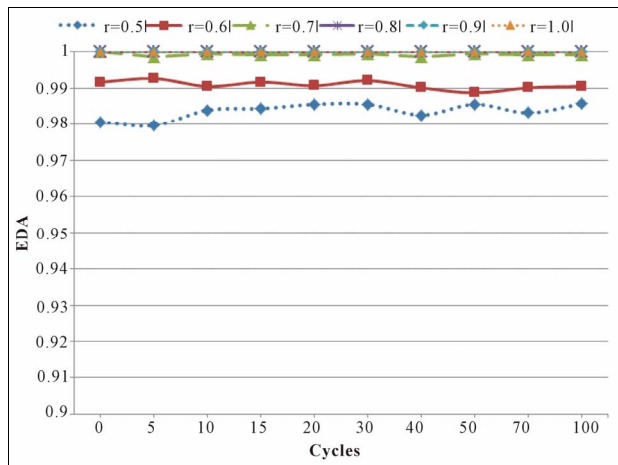
Figure 6. FAR for a grid for various values of p_{ma} when $r = 0.7l$, $p_t = 0.1$, and $p_m = 0.2$.

for various cycles of operation when $r = 0.5l$, is shown in **Figure 8**. The ratio slowly decreases with time and approaches 0.4 when $p_t = 0.1$ and $p_m = 0.2$. The ratio for $p_t = 0.05$ and $p_m = 0.1$ approaches 0.3 instead due to reduction in the number of fault induced alarms. Our weighted voting applied within the circle centered at the estimated center can tolerate the inaccuracy of the estimation, resulting in high EDA as already shown in **Figure 7**.

Malicious nodes reporting a 0 when an event has occurred can be detected if they are in an event region. However, it is difficult to find the exact boundary of an event region locally in a distributed manner, without significant overhead in computation and communication. A normal node close to the event boundary might be determined to be suspicious. Such an incorrect decision lowers the weight of the normal node, and it might lead to the loss of network connectivity. Hence we use a conservative approach in updating the weights in case of an event, not to sacrifice normal nodes. In the simulation we



(a)



(b)

Figure 7. EDA for $r = 0.5l$ and various values of p_{ma} after different cycles of operation when (a) $p_t = 0.1$, $p_m = 0.2$, (b) $p_t = 0.05$, $p_m = 0.1$.

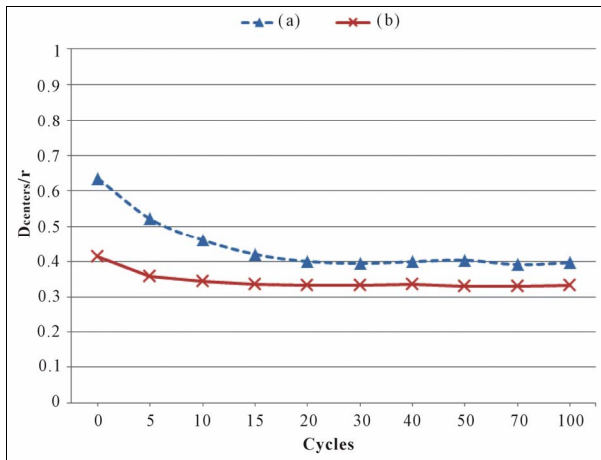


Figure 8. $\frac{D_{centers}}{r}$ for $r = 0.5l$ (a) $p_t = 0.1, p_m = 0.2$, (b) $p_t = 0.05, p_m = 0.1$.

use a circle with radius $r/2$ (i.e., $\delta = 0.5$) centered at the estimated center as the region where the weight updates are applied. To see the accuracy of the updates, we also obtain the distribution of sensor nodes in the reduced region. The ratio of the number of event nodes in the reduced region to the total number of nodes in the region, when $r = 0.5l$ and $p_{ma} = 0.7$ are shown in **Figure 9**, where two different values of p_m and p_t are selected for illustration. For both cases the accuracy improves over time and an over 95% accuracy has been achieved after 20 cycles of operation. Although the updates are made for the sensor nodes within a limited region, the effect is positive in achieving high event and malicious node detection performance.

Event detection accuracy may change with the event region size. Since the simulation results are shown for a relatively small event region to see the worst case performance, we can claim that the scheme performs better as the event region increases.

5. Conclusion

In this paper, we developed a malicious node detection scheme for a grid-based wireless sensor network. The network area is divided into square grids and malicious nodes are detected locally in a distributed manner. For a relatively small event region located across multiple adjacent grids, inter-grid communication is partially employed to enhance the event detection accuracy. Confidence levels (weights) are used to reflect the behavior of sensor nodes in reporting their readings in decision-making. Once the weights reach a predefined lower-bound, the corresponding nodes are logically isolated from the rest of the network. Thresholds are properly chosen to achieve high malicious node detection accuracy without sacrificing normal nodes. The simulation

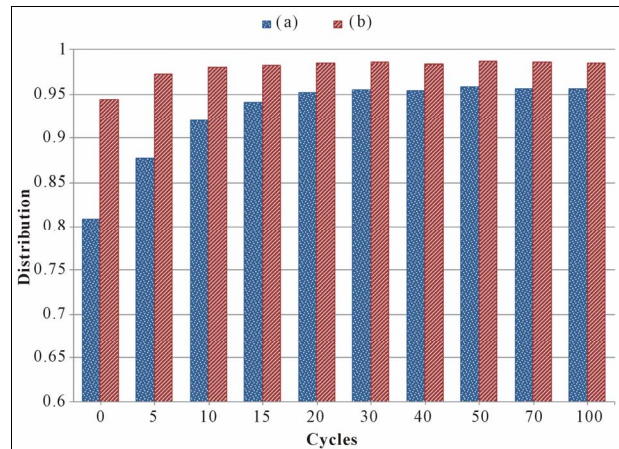


Figure 9. Distribution of normal and event nodes in a reduced region for $r = 0.5l$ and $p_{ma} = 0.7$, when (a) $p_t = 0.1, p_m = 0.2$; (b) $p_t = 0.05, p_m = 0.1$.

results are shown for relatively small event regions to see the worst case performance. Hence the proposed scheme is expected to perform better as the event region increases.

6. Acknowledgements

This research was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (NRF-2011-0007187).

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian and E. Cырci, "Wireless Sensor Networks: A Survey," *Computer Networks*, Vol. 38, No. 4, 2002, pp. 393-422. [doi:10.1016/S1389-1286\(01\)00302-4](https://doi.org/10.1016/S1389-1286(01)00302-4)
- [2] M. Yu, H. Mokhtar and M. Merabti, "Fault Management in Wireless Sensor Networks," *IEEE Wireless Communications*, Vol. 14, No. 6, 2007, pp. 13-19. [doi:10.1109/MWC.2007.4407222](https://doi.org/10.1109/MWC.2007.4407222)
- [3] H. Huangshui and Q. Guihe, "Fault Management Frameworks in Wireless Sensor Networks," *4th International Conference on Intelligent Computation Technology and Automation*, Shenzhen, 28-29 March 2011, pp. 1093-1096.
- [4] S. Rajasegarar, C. Leckie and M. Palaniswami, "Anomaly Detection in Wireless Sensor Networks," *IEEE Wireless Communications*, Vol. 15, No. 4, 2008, pp. 34-40. [doi:10.1109/MWC.2008.4599219](https://doi.org/10.1109/MWC.2008.4599219)
- [5] Y. Zhang, N. Meratnia and P. Havinga, "Outlier Detection Techniques for Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, Vol. 12, No. 2, 2010, pp. 1-12.
- [6] M. H. Lee and Y.-H. Choi, "Fault Detection of Wireless Sensor Networks," *Computer Communications*, Vol. 31, No. 14, 2008, pp. 3469-3475. [doi:10.1016/j.comcom.2008.06.014](https://doi.org/10.1016/j.comcom.2008.06.014)

- [7] X. Xu, B. Zhou and J. Wan, "Tree Topology Based Fault Diagnosis in Wireless Sensor Networks," *International Conference on Wireless Networks and Information Systems*, Hangzhou, 28-29 December 2009, pp. 65-69.
- [8] B. Krishnamachari and S. Iyengar, "Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks," *IEEE Transactions on Computers*, Vol. 53, No. 3, 2004, pp. 241-250. [doi:10.1109/TC.2004.1261832](https://doi.org/10.1109/TC.2004.1261832)
- [9] X. Luo, M. Dong and Y. Huang, "On Distributed Fault-Tolerant Detection in Wireless Sensor Networks," *IEEE Transactions on Computers*, Vol. 55, No. 1, 2006, pp. 58-70. [doi:10.1109/TC.2006.13](https://doi.org/10.1109/TC.2006.13)
- [10] M. Ding, D. Chen, K. Xing and X. Cheng, "Localized Fault-Tolerant Event Boundary Detection in Sensor Networks," *IEEE INFOCOM*, Miami, 13-17 March 2005, pp. 902-913.
- [11] K. Ren, K. Zeng and W. Lou, "Secure and Fault-Tolerant Event Boundary Detection in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 1, 2008, pp. 354-363. [doi:10.1109/TWC.2008.060550](https://doi.org/10.1109/TWC.2008.060550)
- [12] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki and P. Havinga, "Distributed Event Detection in Wireless Sensor Networks for Disaster Management," *2nd International Conference on Intelligent Networking and Collaborative Systems*, Thessaloniki, 24-26 November 2010, pp. 507-512.
- [13] C.-R. Li and C.-K. Liang, "A Fault-Tolerant Event Boundary Detection Algorithm in Sensor Networks," *Information Networking: Towards Ubiquitous Networking and Services*, Vol. 5200, 2008, pp. 406-414. [doi:10.1007/978-3-540-89524-4_41](https://doi.org/10.1007/978-3-540-89524-4_41)
- [14] D. I. Curiac, O. Baniyas, F. Dragan, C. Volosencu and O. Dranga, "Malicious Node Detection in Wireless Sensor Networks Using an Autoregression Technique," *3rd International Conference on Networking and Services*, 19-25 June 2007. [doi:10.1109/ICNS.2007.79](https://doi.org/10.1109/ICNS.2007.79)
- [15] W. Junior, T. Figueiredo, H. Wong and A. Loureiro, "Malicious Node Detection in Wireless Sensor Networks," *18th International Parallel and Distributed Processing Symposium*, Belo Horizonte, 26-30 April 2004, p. 24.
- [16] X.-Y. Xiao, W.-C. Peng, C.-C. Hung, and W.-C. Lee, "Using Sensor Ranks for In-Network Detection of Faulty Readings in Wireless Sensor Networks," *International Workshop Data Engineering for Wireless and Mobile Access*, June 2007.
- [17] I. M. Atakli, H. Hu, Y. Chen, W.-S. Ku and Z. Su, "Malicious Node Detection in Wireless Sensor Networks Using Weighted Trust Evaluation," *Proceedings of Spring Simulation Multi-Conference*, Ottawa, 14-17 April 2008, pp. 836-842.
- [18] L. Ju, H. Li, Y. Liu, W. Xue, K. Li and Z. Chi, "An Improved Detection Scheme Based on Weighted Trust Evaluation for Wireless Sensor Networks," *Proceedings of the 5th International Conference on Ubiquitous Information Technology and Applications*, Sanya, 16-18 December 2010, pp. 1-6.
- [19] M. Momani and S. Challa, "Survey of Trust Models in Different Network Domain," *International Journal Ad Hoc, Sensor & Ubiquitous Computing*, Vol. 1, No. 3, 2010, pp. 1-19. [doi:10.5121/ijasuc.2010.1301](https://doi.org/10.5121/ijasuc.2010.1301)
- [20] M. Momani, S. Challa and R. Alhmouz, "Can We Trust Trusted Nodes in Wireless Sensor Networks?" *International Conference on Computer and Communication Engineering*, Kuala Lumpur, 13-15 May 2008, pp. 1227-1232.
- [21] Y. Li, S. Xiong, Q. Chen and F. Fang, "Grid-Based Directed Diffusion for Wireless Sensor Networks," *Second International Conference on Communications and Networking in China*, Shanghai, 22-24 August 2007, pp. 814-818.
- [22] H. Yuan, H. Ma and H. Liao, "Coordination Mechanism in Wireless Sensor and Actor Networks," *First International Multi-Symposiums on Computer and Computational Sciences*, Hanzhou, 20-24 June 2006, pp. 627-634.
- [23] L. Yu, N. Wang, W. Zhang, and C. Zheng, "GROUP: A Grid-Clustering Routing Protocol for Wireless Sensor Networks," *International Conference on Wireless Communications, Networking and Mobile Computing*, Wuhan, 22-24 September 2006, pp. 1-5. [doi:10.1109/WiCOM.2006.287](https://doi.org/10.1109/WiCOM.2006.287)
- [24] S. Chugh, S. Dharia and D. P. Agrawal, "An Energy Efficient Collaborative Framework for Event Notification in Wireless Sensor Networks," *28th Annual IEEE International Conference on Local Computer Networks*, Bonn, 20-24 October 2003, pp. 430-438.
- [25] M. Asim, H. Mokhtar and M. Merabti, "A Cellular Approach to Fault Detection and Recovery in Wireless Sensor Networks," *Third International Conference on Sensor Technologies and Applications*, Athens, 18-23 June 2009, pp. 352-357. [doi:10.1109/SENSORCOMM.2009.61](https://doi.org/10.1109/SENSORCOMM.2009.61)
- [26] C.-W. Cho and Y.-H. Choi, "Concurrent Diagnosis of Clustered Sensor Networks," *LNCS*, Vol. 3976, 2006, pp. 1267-1272.