

# Fault-Tolerant Event Detection in Wireless Sensor Networks using Evidence Theory

Kezhong Liu<sup>1,2,3,4</sup>, Tian Yang<sup>1</sup>, Jie Ma<sup>1,2,3</sup> and Zhiming Cheng<sup>4</sup>

<sup>1</sup>School of Navigation, Wuhan University of Technology  
Wuhan, Hubei 430063 - China  
[e-mail: kzliu, tiany, majie@whut.edu.cn]

<sup>2</sup>National Engineering Research Center for Water Transport Safety  
Wuhan, Hubei 430063 - China

<sup>3</sup>Hubei Key Laboratory of Inland Shipping Technology  
Wuhan, Hubei 430063 - China

<sup>4</sup>School of Information Engineering, Wuhan University of Technology  
Wuhan, Hubei 430070 - China  
[e-mail: 312385518@qq.com]  
\*Corresponding author: Jie Ma

*Received April 14, 2015; revised July 14, 2015; accepted August 23, 2015;  
published October 31, 2015*

---

## Abstract

Event detection is one of the key issues in many wireless sensor network (WSN) applications. The uncertainties that are derived from the instability of sensor node, measurement noise and incomplete sampling would influence the performance of event detection to a large degree. Many of the present researches described the sensor readings with crisp values, which cannot adequately handle the uncertainties inhered in the imprecise sensor readings. In this paper, a fault-tolerant event detection algorithm is proposed based on Dempster-Shafer (D-S) theory (also called evidence theory). Instead of crisp values, all possible states of the event are represented by the Basic Probability Assignment (BPA) functions, with which the output of each sensor node are characterized as weighted evidences. The combination rule was subsequently applied on each sensor node to fuse the evidences gathered from the neighboring nodes to make the final decision on whether the event occurs. Simulation results show that even 20% nodes are faulty, the accuracy of the proposed algorithm is around 80% for event region detection. Moreover, 97% of the error readings have been corrected, and an improved detection capability at the boundary of the event region is gained by 75%. The proposed algorithm can enhance the detection accuracy of the event region even in high error-rate environment, which reflects good reliability and robustness. The proposed algorithm is also applicable to boundary detection as it performs well at the boundary of the event.

---

**Keywords:** Wireless sensor networks, event detection, evidence theory, fault-tolerant

---

This work is supported by the National Natural Science Foundation of China under Grant Nos. 51279151 and 61203236, the Fundamental Research Funds for the Central Universities under Grant Nos. 2015-zy-106, 2014-zy-142 and 2015IAV041, the Funds for Hubei Key Laboratory of Inland Shipping Technology (No. NHHY2014007)

## 1. Introduction

In recent years, wireless sensor networks (WSNs) have been utilized in a wide range of applications, including military, industrial, agricultural, and environmental sensing uses [1-4]. In almost all of such applications, event detection is a key issue that masses of sensor nodes are distributed in a geographical region to make firm and accurate decisions about the presence or absence of specific events, such as the presence of toxic chemical in water and air [5], fire detection [6], or abnormal target search [7]. Because sensor nodes are typically directly exposed to the environment, they are highly vulnerable to damages caused by external physical and chemical forces. Instability and noise interference also exist within the wireless communication links between nodes, causing sensor network nodes to operate inconsistently, and thus be prone to failure. Therefore, the use of an event detection algorithm with fault-tolerant mechanisms is extremely essential. Fault tolerance is a mechanism that allows networks to determine whether an event has been detected, and identifies failed nodes when a fault occurs within some of the nodes in the network. A possible solution is to provide redundant information contained within the sensory data from neighboring nodes to compensate for untrusted sensor readings.

Such redundant information among nodes has been used for fault-tolerant processing of event detection due to the spatial correlation between adjacent nodes, as mentioned in references [8-12]. Krishnamachari *et al.* [8], who early studied the fault tolerance problem associated with event detection in the field of sensor network, proposed a type of Bayesian method (BFTA) based on spatial correlation. The BFTA algorithm assumes that events are spatially correlated and errors are spatially uncorrelated, and that each sensor has the same probability of error occurrence. A sensor node first obtains data from all the neighboring nodes which hold the same judgement as its own, and then uses the Bayesian conditional probability to calculate the estimated acceptance value to determine whether an error or an event has occurred. Chen *et al.* [9] made up for the errors in the work [8] and a optimized threshold calculation approach was presneted. Luo *et al.* [10] made improvements to the fault-tolerant event region model by dividing errors into two types, and thereby theoretically proved that, in the case of a given error rate with upper boundary, an optimal number of neighborhood nodes could be found so that the amount of power spent by the entire network to run the algorithm is maintained at a minimum. References [11-12] incorporated the mechanism of neighboring node collaboration to performe fault tolerance event detection based on the joint decision-making . These methods show that in distributed sensor networks, event nodes are spatially correlated such that it is possible to performe event detection and failed nodes identification in virture of data redundancy within nearby nodes. However, the algorithms do not take full advantage of data self-correlations of the node itself. It only uses the sensor network spatial correlation characteristics to achieve fault detection, which increases the complexity of the algorithm while failing to address detection at the boundary.

In fact, because the environmental noise or hardware failures often make the sampling data collected at nodes uncertain, the spatio-temporal correlation characteristics between nodes can be used to achieve fault-tolerant event detection at the node. Such use of spatial-temporal correlation between nodes to achieve event detection is also mentioned in references [13-16]. Cao *et al.* [13] proposed a spatial-temporal correlation-based event region detection algorithm that uses spatial-temporal correlation between nodes to improve the accuracy of event detection, which attempts to achieve event detection under unbiased conditions.

The aforementioned algorithm shows that temporal or spatial correlation does exist between event nodes, and the use of these features can help obtain a higher rate of event detection. However, the disadvantage is that each node is simply given the same weight, without considering the impact of neighboring nodes at varying distances on event decision-making. Furthermore, when a particular sensor node has been misjudged, it is absolutely considered a faulty node, and it becomes extremely difficult to re-determine and correct such a node.

Many other scholars have also studied the problem of different sensor nodes in the neighboring region affecting event decision-making, as mentioned in references [17-20]. Li et al. [17] established a distance-weighted model (DFWD) that analyzed the impact of network hierarchy on the central node using the Bays conditional probability model to detect fault-tolerant events. Their paper provided different weights to different nodes in order to improve event detection accuracy. Ould et al. [18] proposed the use of a maximum a posteriori probability criterion-based event detection integrated method that considered the detection results from surrounding neighboring nodes. Such an integrated decision method was designed to consider, to some extent, the effect of sensing error rate and the distance of different nodes on the detection results, although the computational complexity was large. Li et al [19] proposed a type of weight distributed fault-tolerant algorithm that used the neighboring region of a node and the information obtained from that region for event detection. Each node in the neighborhood was assigned a different weight, which was used to determine whether the occurrence of a detected event was properly ascertained, and the test result was corrected. The algorithm performs extremely well and shows that the introduction of node-weight can improve event detection probability. However, the external environment or hardware limitations can cause the sensor to detect abnormal data, which can lead to node faults be determined mistakenly. In addition, other existing event detection algorithms generally do not have good event detection at node boundaries, or simply completely ignore event detection at the node boundary.

To summarize, the current event detection problems that exist within WSNs from the perspective of spatial and temporal correlations, or simply from the spatial correlation of various different nodes, has led to research on reducing the impact of node sensing data uncertainty on false alarm and missing alarm rates; such research is currently extremely popular. However, the environmental models of uncertainties are difficult to describe accurately, and there are limited studies on event region and boundary detection. This paper applies characteristic analysis on the network structures and node state distributions to propose a D-S evidence theory-based fault-tolerant event detection algorithm (DSFTED). The method analyzes the impact of the spatial correlation between nodes at different distances and the node states on event detection performance. The output of each sensor node are characterized as weighted evidences instead of crisp values, where neighboring nodes status values are reasonably fused according to their individual contribution for the detection. The method is capable of effectively correcting nodes misjudged as failed nodes. Using D-S evidence theory, which possesses the characteristics of analyzing fuzzy situations, this method is applicable not only in event region detection, but also in event boundary detection.

## 2. Overview of D-S Theory

D-S evidence theory can be viewed as a general extension of the traditional classical probabilistic inference theory in the finite domain, whose main feature is to support the description of different levels of accuracy by introducing directly the uncertainty of the

unknown into the description.. This theory allows inferring from imprecise and imcomplete contexts, and has been applied in various domains such as object recognition, diagnosis, and particularly in multi-sensor based applications. The following is a brief introduction to D-S evidence theory [21].

The basic concept in D-S evidence theory is a probability function that must first be defined for evidence that supports a system hypothesis called a Basic Probability Assignment (BPA)

**Definition 1.** The frame of discernment  $\Omega$  is a finite hypothesis space that consists of mutually exclusive propositions. A Basic Probability Assignment (BPA) or mass function  $m$  is defined as a mapping of the set  $2^\Omega$  to  $[0,1]$  satisfying  $m(\emptyset)=0$  and  $\sum_{A \in \Omega} m(A)=1$ , where the subset  $A$  of  $\Omega$ , representing any hypothesis, is called the focal set elements if  $m(A)>0$ . In the case of event detection,  $m(A)$  provides a basic belief for event  $A$ , which corresponds to the trust level to proposition  $A$ .

Dempster's combination rules make it possible that multiple evidences can be fused to get a joint support contribution and at the same time reduce uncertainties. The rule is defined as follows:

**Definition 2.** Let  $m_1$  and  $m_2$  be mass functions for two pieces of evidence, the fusing result of these two evidences is a new mass function, which incorporates the joint information provided by the sources as formula (1) below:

$$m_1(A) \oplus m_2(A) = K^{-1} \sum_{B \cap C = A} m_1(B) m_2(C) \quad (1)$$

Where  $K$  is the normalization factor:  $K = \sum_{B \cap C \neq \emptyset} m_1(B) m_2(C)$

The generalized rule for combining  $n$  number of evidences is presented as formula (2):

$$m_{1\dots n}(A) = K^{-1} \sum_{\cap_i A_i = A} m_1(A_1) m_2(A_2) \dots m_n(A_n) \quad (2)$$

Where  $K = \sum_{\cap_i A_i \neq \emptyset} m_1(A_1) m_2(A_2) \dots m_n(A_n)$

### 3. Event Detection using Fusion of Multi Node Evidences

#### 3.1. Local Detection

The consecutive sensor readings of a particular sensor usually depict a smooth variation over time, which can be generally accepted as the property of temporal correlation. It means that a sensor's own reported readings would be similar to the readings it reported in the preceding instants. Consequently, identification of sudden, abnormal readings deviating notably from its common readings beyond a prespecified threshold helps the node to make its local decisions.

Assume  $n$  number of sensor nodes are uniformly distributed across a region of interest in order to detect a specific event. Whenever the sampling value of one node exceeds a threshold  $S_{th}$ , the event detection procedure is activated at the node. To confirm if an event does occur or not, the node can rely on the sampled data on which a number counting procedure is performed within a time window  $T$  that is called as detection window. The counting results are denoted

as two variables  $l$  and  $q$ , where  $l$  and  $q$  are increased by one respectively on each catching of an abnormal data and each occurring of an normal data that belows the threshold  $S_{th}$ . If the amount of the abnormal data  $l$  exceeds  $C_{th}$ , the node is considered as a detected event, otherwise, it is considered as a potential node error, namely

$$u_i = \begin{cases} 1, & l \geq C_{th} \\ 0, & otherwise \end{cases} \quad (3)$$

The node local-detection algorithm is described in **Table 1**.

**Table 1.** Local-detection algorithm

```

Input: threshold  $S_{th}, C_{th}$ , time window  $T$ 
Out: local-decision  $u_i$ , the counting number  $l$  and  $q$ 
(1) //For each node  $i$ 
(2) if Sampling value of the node exceeds  $S_{th}$  then
(3)   Trigger an event detection procedure
(4)   Initiate the number  $l=0$ 
(5)   Initiate the number  $q=0$ 
(6) end if
(7) //Start event detection procedure
(8) if The node  $i$  detection procedure is active then
(9)   while(  $T$  not end)
(10)    if abnormal data then
(11)      the number  $l$  is increased by one
(12)      if  $l \geq C_{th}$  then break
(13)    end if
(14)    else if normal data
(15)      the number  $q$  is increased by one
(16)    end if
(17)  end while
(18) end if
(19) // Local-decision
(20) if  $l \geq C_{th}$  then  $u_i=1$  event detected
(21) else  $u_i=0$  no event
(22) end if
(23) Back to step (2) for next iteration

```

### 3.2. Event Decision

The main advantage of local detection algorithm illustrated before is that each sensor node makes the best of local observations of the covered area and the temporal correlation inhered in the node. However, the local detection cannot eliminate faulty sensor readings caused by noise-related failures, biased measurement, node-linking failures and environmental failures. The uncertainties contained within the local decisions have not yet to be effectively addressed. In this case, a collaborative scheme could be used to promote the reliability of the event detection decisions. Due to the property of spatial correlation, at a particular instant, a sensor's sampled data value is similar to the ones sensed by its one-hop neighbors. Therefore, the local detection decisions of each node regarded as evidences can be fused together using D-S combination rules to improve the accuracy of event detection.

Consider a sensor network composed of massive nodes distributed over a detection field. Each node allocated with a sensor and is targeted to detect the presence of a specific event. In order to realize the evidence fusing based event detection process, the frame of discernment is firstly defined as  $\Theta \in (E, NE)$ , where  $E$  indicates the occurring of an event, and  $NE$  indicates none-event with  $E \cap NE = \emptyset$ . Then, the defined BPA mass function  $m : p\{E, NE\} \rightarrow [0, 1]$  satisfies:

$$m(\emptyset) = 0 \quad (4)$$

$$m(E) + m(NE) + m\{E, NE\} = 1 \quad (5)$$

Where  $m(E)$  represents the probability of an event being detected at the node,  $m(NE)$  represents the probability of normal detection at the node, and  $m\{E, NE\}$  represents the probability that the presence of an event at the node cannot be fully confirmed, which discloses the uncertainties as discussed before. For each node, three Basic Probability Assignment functions are defined to support the hypotheses correspondingly as follows:

$$m(E) = \frac{l}{m} \quad (6)$$

$$m(NE) = \frac{q}{m} \quad (7)$$

$$m\{E, NE\} = 1 - m(E) - m(NE) \quad (8)$$

On the basis of the local detection procedure at each node, these Basic Probability Assignment functions can be easily achieved accordingly. The output of BPA functions with regarding to a particular event may hold different values because of the diverse beliefs made by local decisions of each node on the event. Then, the evidence fusing precess of a node is realized by combining the BPA functions provided by its neighboring nodes. According to the combination rule represented in formula (2), a new BPA function as the fusing result can be obtained as:

$$\begin{aligned} m_{1,2,\dots,N}(E) &= m_1(E_1) \oplus m_2(E_2) \oplus \dots \oplus m_N(E_N) \\ &= \frac{\sum_{E_1 \cap E_2 \cap \dots \cap E_N = E} \prod_{i=1}^N m_i(E_i)}{1 - K} \end{aligned} \quad (9)$$

where  $K = \sum_{E_1 \cap E_2 \cap \dots \cap E_N = \emptyset} m_1(E_1) \oplus m_2(E_2) \oplus \dots \oplus m_N(E_N)$ . Different source evidences maintain different beliefs on the same event. The closer the data source to the event central, the greater is the belief level, with normal nodes having higher belief than failed nodes. In order to further improve the detection performance, it is necessary to assign a weight to the node BPA function, that is to assign each node a different weight  $C_i$ , which will be detailed in the following section. Then, a node-weighting evidence combination rule is proposed in formula (10) by rewriting formula (9). The output denoted as  $m_{1,2,\dots,N}(E)$  represents the node's final event decision using fusion of its  $N$  neighboring nodes generated evidences.

$$m_{1,2,\dots,N}(E) = (m_1(E_1) \oplus m_2(E_2) \oplus \dots \oplus m_N(E_N)) C_i$$

$$\begin{aligned}
&= \frac{\sum_{\cap_i A_i=A} m_1(E_1)m_2(E_2)...m_N(E_N)}{\sum_{\cap_i A_i \neq \emptyset} m_1(E_1)m_2(E_2)...m_N(E_N)} C_i \\
&= \frac{\sum_{\cap_i A_i=A} m_1(E_1)m_2(E_2)...m_N(E_N)}{1 - \sum_{\cap_i A_i \neq \emptyset} m_1(E_1)m_2(E_2)...m_N(E_N)} C_i \quad (10)
\end{aligned}$$

The above proposed fusing process will be implemented at each node that a comprehensive knowledge about the current event status can be achieved. Each sensor node covering the area of interest, along with its  $N$  neighboring nodes, may issue three claims  $m_{1,2,\dots,N}(E)$ ,  $m_{1,2,\dots,N}(NE)$  and  $m_{1,2,\dots,N}\{E,NE\}$ , which represent the node beliefs on abnormal event, normal status, and unknowns respectively. As a result, a more reliable and improved statement about the status of the node  $i$  is defined as follows:

$$R_i = \begin{cases} 1, & m_{1,2,\dots,N}(E) \geq \theta \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Using equation (11), an neighboring-evidence based event decision can be gained by each node. Then, if one node's local-detection status is  $u_i=1$  and  $m_{1,2,\dots,N}(E) \geq \theta$ , the node believes it has correctly detected an event, in which case its status is changed to Event; or if the node status satisfies  $u_i=0$  and  $m_{1,2,\dots,N}(E) \leq \theta$ , it believes that no event occurred. Otherwise, if  $u_i=1$  and  $m_{1,2,\dots,N}(E) \leq \theta$ , as well as if  $u_i=0$  and  $m_{1,2,\dots,N}(E) \geq \theta$ , which mean there occurs a collision between the node local-detection and the neighboring fusion decision, the node can be regarded as faulty node that may cause false or missing reports. The event decision algorithm is listed in [Table 2](#).

**Table 2.** Event decision algorithm

<p>Input: the local-detection <math>u_i</math>, threshold <math>\theta</math>  Out: the final status <math>R_i</math></p> <ol style="list-style-type: none"> <li>(1) //For each node</li> <li>(2) /compute <math>C_i</math>, <math>m_{1,2,\dots,N}(E)</math> and <math>R_i</math></li> <li>(3) Broadcast the geometric position (<math>LOC_i(x,y)</math>)</li> <li>(4) Broadcast the weight of node <math>i</math>, <math>C_{ii}</math></li> <li>(5) Broadcast the BPA function <math>m_i</math></li> <li>(6) Compute the weight of node <math>C_i</math></li> <li>(7) Compute the neighboring node's belief probability <math>m_{1,2,\dots,N}(E)</math></li> <li>(8) Determine the estimated value <math>R_i</math></li> <li>(9) <b>if</b> <math>u_i=1</math> <b>then</b></li> <li>(10)     <b>if</b> <math>m_{1,2,\dots,N}(E) \geq \theta</math> <b>then</b></li> <li>(11)         <math>R_i=1</math> report event</li> <li>(12)     <b>else</b> <math>R_i=0</math> false report</li> <li>(13)     <b>end if</b></li> <li>(14) <b>else if</b> <math>u_i=0</math></li> <li>(15)     <b>if</b> <math>m_{1,2,\dots,N}(E) \geq \theta</math> <b>then</b></li> <li>(16)         <math>R_i=1</math> missing report</li> <li>(17)     <b>else</b> <math>R_i=0</math> none event</li> <li>(18)     <b>end if</b></li> <li>(19) <b>end if</b></li> </ol>
---

### 3.3. Weight Setting

Taking advantage of the property of spatial correlation, collaborative event detection has

become more and more popular in wireless sensor networks. Thus, the inter-node cooperation and information sharing play an essential role in the collaboration. In this case, one node's status is greatly affected by its neighboring nodes within the surrounding area. Existing neighborhood-based detection methods [17-20] only consider the position relationship or different sensor information attributes between nodes. However, in an actual environment, the neighboring nodes are likely to make missing or false information errors that would affect the detection accuracy of the central node. Therefore, we propose a dual weighting model that incorporates the weight of node  $i$ , *i.e.*,  $C_{ii}$ , and the relative weight of its neighboring node  $j$ , *i.e.*,  $C_{ij}$ . The recorded node weight  $C_i$  is as follows:

$$C_i = r \times C_{ii} + \frac{\sum_{j=1}^N C_{ij} u_j}{N} (1-r) \quad (12)$$

Supposing the node  $i$  have  $N$  neighboring nodes, and  $r$  is the node self-weighting factor being set to  $r=0.5$ . Furthermore, the node weight  $C_{ii}$  is determined by its own status, where  $C_{ii}$  is set to zero for a failed node, and one for a normal node. Because the local decision results of one node may switch between right and wrong,  $C_{ii}$  should not be unchangeable. Using equation (13), the weight of node  $i$  can be calculated as

$$C_{ii} = e^{-x} \quad (13)$$

Where  $x$  ( $x \geq 0$ ) represents the intermediate variable for calculating  $C_{ii}$  with initial value of zero. If node detection is in error,  $x$  increases, which reduces the weight of the node; if node detection is correct,  $x=0$ , the value of  $x$  does not change, and its weight is still at maximum; if node detection is correct and  $x>0$ ,  $x$  decreases, which increases its weight; if an accidental error occurs at the node, its weight quickly returns to normal value. When the value of  $C_{ii}$  is less than the threshold  $\varepsilon_{error}$ , the node is considered an invalid node, and it is no longer included in event detection.

In addition, taking account of the spatial correlation between nodes, it is reasonable to believe that the geographical distribution of network nodes is highly correlated to the accuracy of event detection. Therefore, the weight of a node is related to its spatial distribution. The relative weight  $C_{ij}$ , which indicates the impacts caused by the neighboring node  $j$  on the central node  $i$ , is calculated using the neighboring node's own weight  $C_{jj}$  and the distance weight  $w_{ji}$  as follows:

$$C_{ij} = C_{jj} w_{ji} \quad (14)$$

The weight  $w_{ji}$  is obtained by following equations:

$$InvDist_{ji} = \sum_{j=1}^N Distance_{ji} - Distance_{ji} \quad (15)$$

$$w_{ji} = \frac{InvDist_{ji}}{\sum_{j=1}^N Distance_{ji}} \quad (16)$$

Where  $Distance_{ji}$  is the Euclidean Distance calculated based on the geometric position of nodes  $j$  and  $i$  as coordinates  $(x,y)$ , and  $N$  is the amount of neighboring nodes for the node  $i$ .

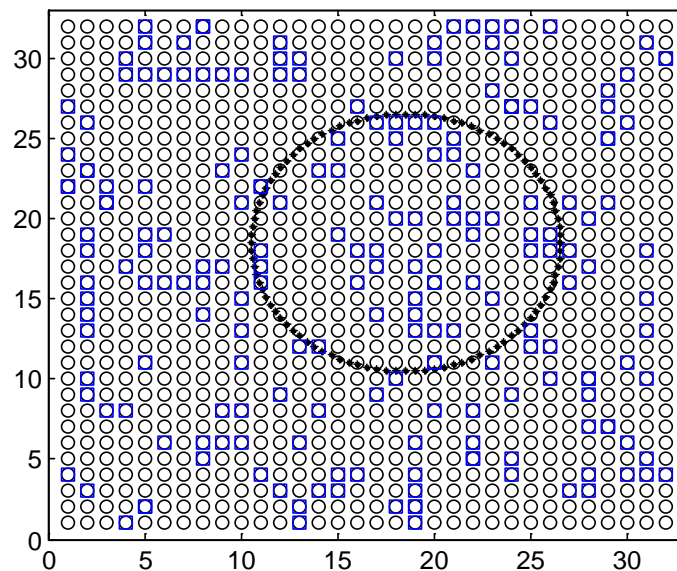


## 4. Experiments and Evaluation

### 4.1 Experimental Setup

From this section, a set of simulation experiments are conducted to evaluate the performance of the proposed algorithm DSFTED. The sensor network used in the simulation contains 1024 nodes randomly deployed (uniform distribution) in a grid region of 32 meters by 32 meters. The communication range between nodes was set to  $\sqrt{2}$  meters resulting in the fault-tolerance range that each interior node has four neighbors that are used in the collaborative detection mechanism. **Fig. 1** shows a sample scenario of the simulated event region. In the figure, one source event was placed at the coordinate (18.5, 18.5) and the sensing range was set to 8 meters. The black circles in the figure indicate normal nodes and the squares represent failed nodes, wherein the sensor failure probability for each node can be flexibly preset to meet the various demands of test.

The node self-weighting factor  $r$  and the self-weight threshold  $\epsilon_{error}$  should be set reasonably in the experiment. The node weight  $C_i$  contributes variously along with different self-affecting factors and distance weight  $W$ , which is shown in **Fig. 2**. As can be seen in the figure, a larger  $W$  means that it is closer to the central node and the weight is greater; the larger the self-weighting factor  $r$ , the greater is the weight of the node, which set  $r=0.5$ . If the node is determined to be a failed node,  $x$  is increased by one and the node is removed from calculation. This can cause erroneous judgment. Therefore, in order to accurately detect and quickly eliminate failed nodes, the self-weight value threshold is set to  $\epsilon_{error}=0.3$ . If the self-weight value of the node is below the threshold, the node stops communicating with its neighboring nodes.



**Fig. 1.** A sample scenario of event region for a simulated WSN with  $32 \times 32$  nodes

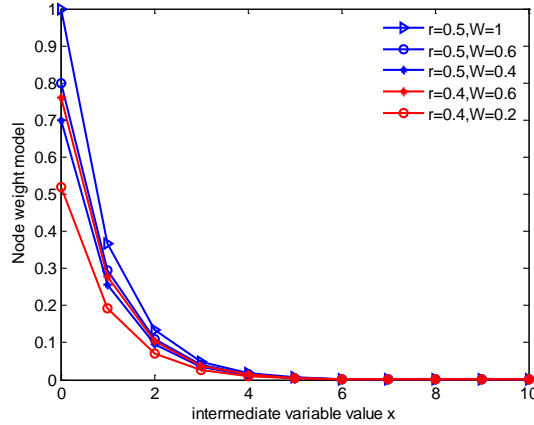


Fig. 2. Relationship between node weight and intermediate variable  $x$

The detection window is supposed to be  $T=1s$ , and the maximum amount of data collected during the time window is  $m=10$ . The node sampling value in the normal region follows the normal distribution  $N(\mu_n, \sigma_n)$ , where  $\mu_n=10$ ,  $\sigma_n=1$ , and the nodes in the event region follow  $N(\mu_f, \sigma_f)$ , where  $\mu_f=30$ ,  $\sigma_f=1$ . Following this, the rules of majority voting are used to obtain the node -local-decision threshold value  $C_{th}=8$  and the event decision threshold value  $\theta=1$ . Without loss of generality, all simulation results are obtained by averaging over 100 runs with varying node failure probabilities.

## 4.2 Performance Metrics

In order to evaluate the node's detection performance in the event region, the event region detection rate, event false alarm rate, event miss-alarm rate, and event fault detection rate were set up as performance evaluation indicators for the fault-tolerant algorithm:

- The event region detection rate, represented by ERDR, indicates the ratio between the number of normal nodes that correctly reported the occurrence of an event and the number of normal nodes within the event region.
- The event false alarm rate, represented by EFAR, indicates the ratio between the number of nodes that detected an event and the total number of nodes in the event region when no event occurs.
- The event missing-alarm rate, represented by EMAR, indicates the ratio when an event occurs between the number of nodes that failed to detect an event and the total number of nodes in the event region.
- The event fault detection rate, represented by EFDR, indicates the ratio between the number of nodes that accurately identified a fault and the total number of faulty nodes.

To evaluate detection effectiveness at event boundaries, the event boundary detection rate and false detection rate are used as the two indicators to perform the performance analysis:

- The event boundary detection rate, represented by EBDR, indicates the ratio of event boundary nodes detected to event boundary nodes.
- The event boundary false detection rate, represented by EBMAR, represents the ratio between the number of boundary nodes that have not detected and the total number of event boundary nodes.

### 4.3 Performance of Event Detection

The highlight of the proposed algorithm DSFTED took both the contribution of neighboring node and the spatial correlation between nodes into consideration. The algorithm also adopts distributed computing ideas that collaborative event detection scheme can be achieved. One hop communication is considered which means each node only contact with its neighboring nodes.

The proposed algorithm DSFTED is implemented in the Matlab-based environment using the experimental setting as section 4.1. The simulation results are compared with the optimal threshold event detection algorithm (OTDS) in reference [9] and the DFDW algorithm in reference [17].

Fig. 3, 4 and 5 show several snapshots of the event detection results using OTDS, DFDW, and the proposed algorithm, respectively, with 20% node fault rate and red solid dots representing the normal nodes that misjudged as errors, that is, the newly introduced error nodes. Blue solid square indicators represent the identified faulty nodes. As can be seen from Fig. 3, 4 and 5, when the sensor node fault rate reaches 0.2, more faulty nodes are accurately detected, and lower error nodes are introduced using the proposed algorithm.

According to the algorithm performance assessment indicators, we conducted four groups of experiments in order to compare OTDS, DFDW, and DSFTED.

As shown in Fig. 6, when the node fault rate is below 0.1, the event region detection rate of all three methods is above 0.78. However, as the fault rate continues to increase, the event detection performance of the OTDS and DFDW algorithms decreases significantly, which undeniably indicates that the DSFTED algorithm has better detection performance.

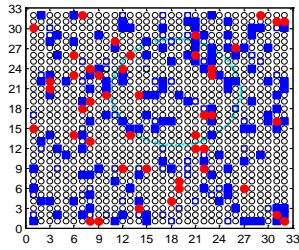


Fig. 3. The results of OTDS algorithm

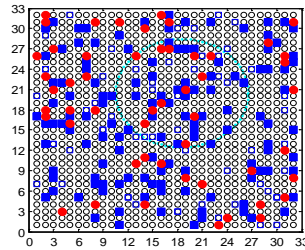


Fig. 4. The results of DFDW algorithm

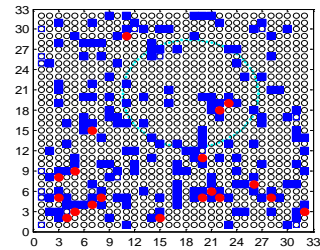


Fig. 5. The results of DSFTED algorithm

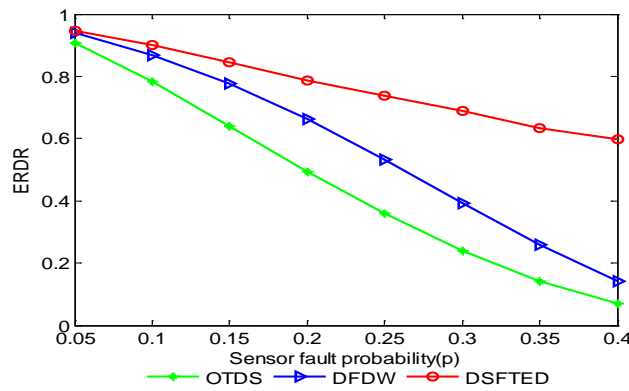
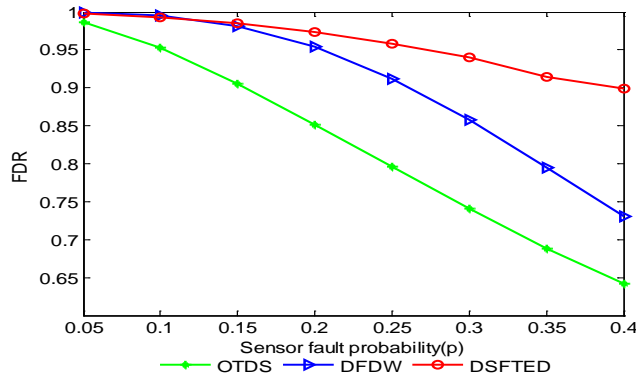


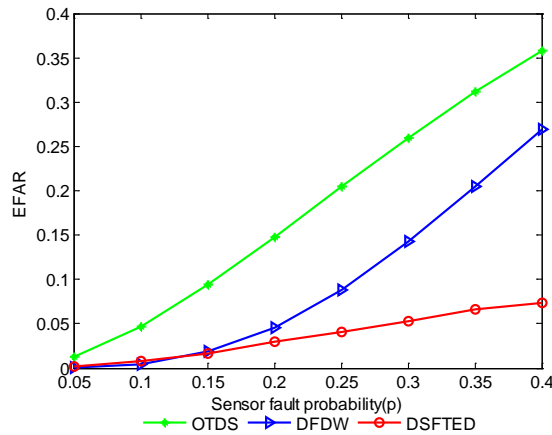
Fig. 6. The ERDR of OTDS algorithm [9], DFDW algorithm [17] and DSFTED algorithm

**Fig. 7** shows that when the node fault rate reaches 15%, the fault identification rate is 85% for OTDS and 95% for DFDW, whereas the proposed DSFTED algorithm reaches approximately 97%, which is a significant increase. This is because the majority of the failed nodes can be eliminated according to the size of the node weight. When the weight of a particular node is lower than the threshold value  $\varepsilon_{error}=0.3$ , other nodes do not use the data collected from that node, which reduces the impact of the failed node on other normal nodes, so that the fault tolerance capability of the algorithm increases.



**Fig. 7.** The FDR of OTDS algorithm [9], DFDW algorithm [17] and DSFTED algorithm

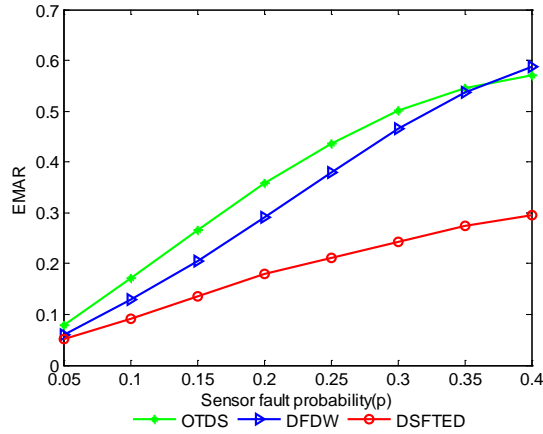
**Fig. 8** shows that when the node fault rate is below 15%, the false alarm rates calculated using the DFDW algorithm and the algorithm proposed this paper are significantly lower than for OTDS. In addition, when the node fault rate is high, the proposed algorithm always generates relatively low false alarm rates.



**Fig. 8.** The EFAR of OTDS algorithm [9], DFDW algorithm [17] and DSFTED algorithm

**Fig. 9** shows that as the node fault rate increases, the miss-alarm rate from OTDS tends to increase. When the node fault rate is 15%, OTDS reaches 37%, DFDW reaches approximately 21%, and the proposed algorithm reaches 14%, which is significantly lower than the other two algorithms. This is because, when the status of the node spatial distribution and the node status are considered, after the effect on event detection achieved, a few failed nodes can be eliminated to some extent, whereas a few nodes determined as failed can be recovered to

normal nodes in order to avoid miss-alarm errors.

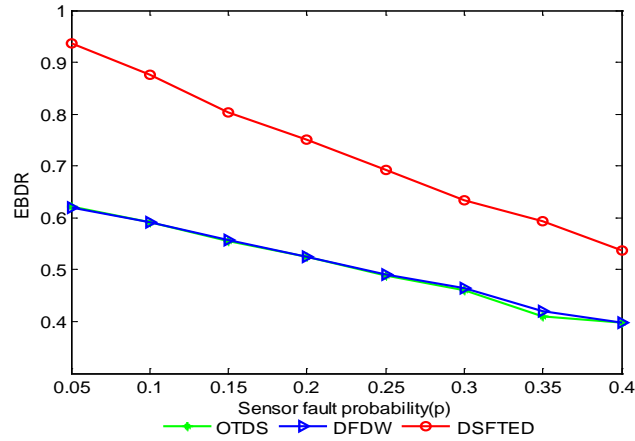


**Fig. 9.** The EMAR of OTDS algorithm [9], DFDW algorithm [17] and DSFTED algorithm

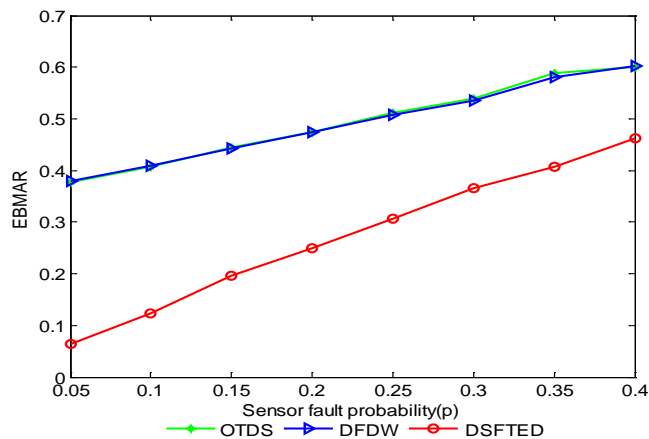
#### 4.4 Performance of Event Boundary Detection

The three algorithms are also applied on event boundary nodes within the same experimental scenario as before. **Fig. 10** shows the relationship between event boundary detection rate and the network fault rate. It can be noticed that the event detection rate calculated with the proposed algorithm is far better than with the OTDS and DFDW algorithms. For example, when the fault rate is 15%, the event detection rate from DSFTED is approximately 80%, whereas OTDS and DFDW are similar, and both are lower than 58%. **Fig. 11** represents the relationship between missing-alarm rate and the network fault rate. It shows that DSFTED under a considerably high fault rate still gains a relatively lower miss-alarm rate when comparing with OTDS and DFDW algorithms. The reasons are:

- The D-S evidence theory-based fault tolerance model is effective in resolving the uncertainty problem inhered in the event boundary node readings. Through incorporation of the double-weight model that assigns nodes with different weights results in diverse beliefs being fused to improve the discrimination of events occurred on the edge of event region;
- The introduced node weight model is not only based on network topology, but also adaptable to node changes. Data collected from nodes with weight less than  $\varepsilon_{error}=0.3$  are removed from the event boundary algorithm. In particular, after the prior detection, the faulty nodes will not have an impact on event boundary detection in the next round, and the nodes misjudged as failed nodes can be restored to normal nodes and rejoin the event decision-making.



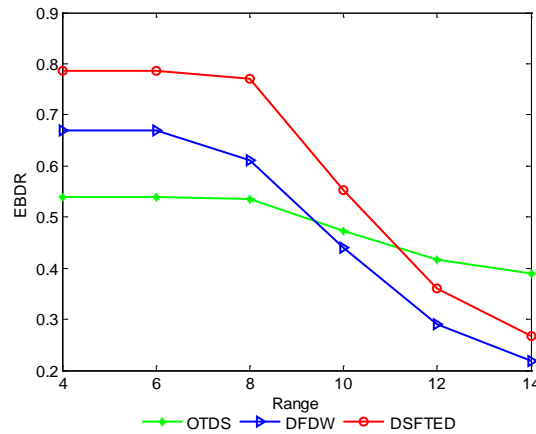
**Fig. 10.** The EBDR of OTDS algorithm [9], DFDW algorithm [17] and DSFTED algorithm



**Fig. 11.** The EBDR of OTDS algorithm [9], DFDW algorithm [17] and DSFTED algorithm

**Fig. 12** presents the event boundary detection rate (EBDR) versus detection range using the OTDS, DFDW, and DSFTED algorithms when the node fault rate reaches 20%. Considering the same event region as defined before with a center at (18.5, 18.5) and the radius being set as eight, the nodes at the edge of different ranges varying from 4 to 14 ( $3 < \text{range} < 15$ ) are examined respectively. The results indicate the same trend for all the algorithms that the detection rate decreases with the increasing range. When the range is beyond the event region ( $\text{range} \geq 8$ ), the detection rate of DSFTED declines very fast compared with the other two algorithms. This can be reasonably explained that the nodes inside the event region are inclined to be supported with more consentaneous evidences from its neighbors that finally authenticate such nodes as events. In contrast, the nodes approaching the edge of the event region are more likely to be affected by confused evidences that cause the decrease of event detection rate. As shown in the figure, a sudden change occurs for all the three algorithms at the event edge ( $\text{range}=8$ ). However, the proposed DSFTED algorithm behaves more seriously. It seems that an abrupt decline point divides the detection curve into two parts that the upper represents the inside of the event region and the lower represents the

outside of the event region. Therefore, such abrupt decline can be regarded as a crucial feature which may be applied to identify event boundary nodes from other nodes. Although this paper did not discuss too much on this issue, the DSFTED algorithm could be further prompted to deal with it.



**Fig. 12.** The EBDR of OTDS algorithm [9], DFDW algorithm [17] and DSFTED algorithm

#### 4.5 Energy Consumption Analysis

Wireless sensor networks are usually embodied with severe energy constraints since the sensor nodes often operate with finite battery resources and limited recharging. An effective energy using and saving mechanism will be beneficial to promoting life cycle of network system. The energy concerns can be addressed by engineering design at all layers including data sensing, data processing and message passing. It has been recognized that energy savings can be obtained by dispersing computation within the network in a distributed form, such as the above discussed algorithms OTDS [9], DFDW [17] and the proposed algorithm DSFTED as well. Assuming all the three algorithms are deployed in a region of interest sharing the same environment and node sensing abilities, the cost of data sensing for each algorithm can be regarded as equal, and then such cost may be safely ignored when making energy consumption analysis. Additionally, the computing complexity of each algorithm can be proved to be linear that the data processing only contributes a little to the energy cost. Therefore, it is reasonable to believe that the majorities of energy cost in a sensor network are due to the data communication between nodes.

To comprehensively understand the data exchanging schemes and make comparative analysis of energy consumption caused by communication, some definitions should be reclaimed here. Considering a sensor network composed of  $n$  nodes and each node owns  $N$  neighbors, the data exchanging usually arises between the interior node and its neighboring nodes. Thus, the communication cost mostly depends on the data exchanging frequency, the more frequent the node communicates with its neighbors, the more energy it cost. Given a synchronized time window, just as the predefined detection window  $T$ , the data exchanging frequency of each algorithm can be examined. In the algorithm of OTDS, each node sampling will trigger a binary decision involving one round of data exchanging between the node and its  $N$  neighbors. During the time window  $T$ , the total amount of data exchanging of the entire network generated by OTDS can be calculated as  $n \cdot N \cdot F / T$ , where  $F$  represents the sampling times within the window  $T$ , namely  $f = F / T$  indicates the data exchanging frequency. The



DFDW algorithm has a similar triggering scheme like OTDS that it is endowed with the same data exchanging frequency, i.e.  $F/T$ . However, a dual-neighborhood detection scheme was incorporated in DFDW. Before a node gather information from its own neighbors to make the final decision, the node's neighboring nodes should finish one round data collection from their external neighbors. Then, the amount of network nodes involved in the communication to accomplish once detection is denoted as  $N'$  satisfying  $N \leq N' \leq N + N^2$ , and the statistical result of data exchanging is  $n \cdot N' \cdot F/T$ . Unlike the two algorithms as discussed before, the proposed algorithm DSFTED uses belief probability instead of binary value as evidence to make the fusing decision, and only one time data exchanging need to be performed for the decision during the window  $T$ . So, the amount of data exchanging can be calculated using  $n \cdot N \cdot 1/T$ . Without loss of generality, it is acceptable to use the amount of data exchanging to represent the energy consumption of the sensor network as summarized in **Table 3**.

**Table 3.** Comparison of energy consumption

Algorithm	Energy cost	Ratio
OTDS	$n \cdot N \cdot F/T$	1
DFDW	$n \cdot N' \cdot F/T$ ( $N \leq N' \leq N + N^2$ )	$N'/N$
DSFTED	$n \cdot N \cdot 1/T$	$1/F$

Being divided by the common factor  $n \cdot N \cdot F/T$ , the energy cost can easily be transferred to the resulting ratio as shown in the left column of the table. Thus we can see that DFDW cost more energy than OTDS because the ratio value  $N'/N$  is bigger than one. The proposed algorithm may rank the minimum cost if  $F > 1$  or  $1/F < 1$ . In general, we believe the value of  $F$  is larger than one since both OTDS and DFDW cannot make their decisions only using once data sampling and exchanging within the time window  $T$ . It is necessary to point out that the analysis does not consider the message load during the communication since we may reasonably assume a uniform data format is applied in all communications and the length of message is short enough to be ignored. Although a relatively tough analysis and comparison have been demonstrated, we may safely conclude that the proposed algorithm can maintain the energy consumption in a considerably lower level which is acceptable in practical applications.

## 5. Conclusion

This paper introduced a D-S evidence theory-based weighted fault tolerance mechanism that can process uncertain states, and therefore, it is capable of achieving both event and event boundary detections. The proposed algorithm DSFTED applies a double-weight model to characterize the impact of both node geographical distribution and node self-state, where the spatio-temporal correlations within nodes are modeled as evidences particularly using BPA functions. The experimental results showed that in case of high faulty node rate, the approach could achieve lower miss-alarm rate and false alarm rate, and higher fault identification rate. Further research work can be conducted on the issue of data error correction during transmission, and on methods to prolong network lifetime to ensure maximized correction effect.



## References

- [1] I.F. Akyildiz, et al. "Wireless sensor networks: a survey," *Computer networks*, vol. 28, no. 4, pp. 393-422, 2002. [Article \(CrossRef Link\)](#).
- [2] P. M. Ashok Kumar, "Anomalous Event Detection in Traffic Video Based on Sequential Temporal Patterns of Spatial Interval Events," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 9, no. 1, pp. 169-189, 2015. [Article \(CrossRef Link\)](#).
- [3] N. Amini, A. Vahdatpour, W. Xu, M. Gerla, and M. Sarrafzadeh, "Cluster size optimization in sensor networks with decentralized cluster-based protocols," *Computer Communications*, vol. 23, no. 2, pp. 207-220, 2012. [Article \(CrossRef Link\)](#).
- [4] D.D. Geeta, N. Nalini, and R.C. Biradar, "Fault tolerance in wireless sensor network using hand-off and dynamic power adjustment approach," *Journal of Network and Computer Application*, vol. 36, no. 4, pp. 1174-1185, 2013. [Article \(CrossRef Link\)](#).
- [5] T. Arici, and Y. Altunbasak, "Adaptive sensing for environment monitoring using wireless sensor networks," *Wireless Communications and Networking Conference*, vol. 4, pp. 2347-2352, 2004. [Article \(CrossRef Link\)](#).
- [6] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proc. of the Second European Workshop on. IEEE*, pp. 108-12, 2005. [Article \(CrossRef Link\)](#).
- [7] R.R. Brooks, C. Griffin, and D. Friedlander, "Self-organized distributed sensor network entity tracking," *International Journal of High Performance Computing Applications*, vol. 16, no. 3, pp. 207-219, 2002. [Article \(CrossRef Link\)](#).
- [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, pp. 241-250, 2004. [Article \(CrossRef Link\)](#).
- [9] Q. Chen, K.Y. Lam, and P. Fan, "Comments on distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transaction on Computers*, vol. 54, no. 9, pp. 1182-1183, 2005. [Article \(CrossRef Link\)](#).
- [10] X.W. Luo, M. Dong, and Y.L. Huang, "On distributed fault-tolerant detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 58-70, 2006. [Article \(CrossRef Link\)](#).
- [11] G. Wittenburg, N. Dziengel, and C. Wartenburger, "A system for distributed event detection in wireless sensor network," in *Proc. of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 94-104, 2010. [Article \(CrossRef Link\)](#).
- [12] S.J. Yim, and Y.H. Choi, "An adaptive fault-tolerant event detection scheme for wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 2332-2347, 2010. [Article \(CrossRef Link\)](#).
- [13] D.L. Cao, J.N. Cao, and B.H. Jin, "A fault-tolerant algorithm for event region detection in wireless sensor networks," *Chinese Journal of Computers*, vol. 10, no. 30, pp. 1770-1776, 2007. [Article \(CrossRef Link\)](#).
- [14] G. Wittenburg, and N. Dziengel, "Cooperative event detection for wireless sensor networks," *Communications Magazine*, vol. 50, no. 12, pp. 124-131, 2012. [Article \(CrossRef Link\)](#).
- [15] J. Yin, D.H. Hu, and Q. Yang, "Spatio-temporal event detection using dynamic conditional random fields," in *Proc. of the Twenty-First international joint conference on artificial intelligence*, pp. 1321-1326, 2009. [Article \(CrossRef Link\)](#).
- [16] B. Yao, and Q.C. Chen, "On the temporal-spatial correlation based fault-tolerant dynamic event region detection scheme in wireless sensor networks," in *Proc. of 3rd International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 511-523, 2007. [Article \(CrossRef Link\)](#).
- [17] P. Li, H. Li, and M. Wu, "Distributed event region fault-tolerance based on weighted distance for wireless sensor networks," *Journal of Systems Engineering and Electronics*, vol. 20, no. 6, pp. 1351-1360, 2009. [Article \(CrossRef Link\)](#).
- [18] E. Ould-Ahmed-Vall, B. Heck Ferri and G. Riley, "Distributed fault-tolerance for event detection using heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, pp. 1994-2007, 2012. [Article \(CrossRef Link\)](#).

- [19] H. Li, Z. Xie, and P. Li, "Distributed and weighted fault-tolerant algorithm for event region detection in wireless sensor network," *Journal of System Simulation*, vol. 7, no. 14, pp. 3750-3755, 2008. [Article \(CrossRef Link\)](#).
- [20] C.Y. Wang, X.C. Yang, and X.X. Bao, "A spatio-temporal correlation based outlier detection technology using multiple attributes in wireless sensor networks," *Journal of Computer Research and Development*, vol. 4, no. 2, pp. 82-87, 2008. [Article \(CrossRef Link\)](#).
- [21] G. Sharer, "A mathematical theory of evidence," *Princeton: Princeton University Press*, vol. 1, 1976. [Article \(CrossRef Link\)](#).



**Kezhong Liu** received his BS and MS degrees in School of Navigation from Wuhan University of Technology (WHUT), China, in 1998 and 2001, and the Ph.D. degree in School of Electronic and Information Engineering from Huazhong University of Science & Technology (HUST), China, in 2006. He was a visiting scholar at Arizona State University (ASU) from 2013 to 2014. He is currently a professor in School of Navigation at WHUT. His research interests include wireless sensor networks, ubiquitous computing and intelligent waterborne transportation systems.



**Tian Yang** received his Bachelor's degree in School of Electrical and Electronic Engineering from Hubei University of Technology, China, in 2013. He is currently a Master student in School of Navigation at Wuhan University of Technology. His research interests include the wireless sensor networks, machine learning, and ubiquitous computing.



**Jie Ma** received his Ph.D. degree in School of Computer Science & Technology from Huazhong University of Science & Technology (HUST), China, in 2010. He is currently a lecturer in School of Navigation, Wuhan University of Technology (WHUT). His research interests include information fusion, machine learning and intelligent waterborne transportation systems.



**Zhiming Cheng** received her BS degree in School of Telecommunication from Hankou College, China, in 2012, and her MS degree in School of Information Engineering from Wuhan University of Technology, China, in 2015. Her research interests are in the areas of network communication.