

Efficient Event Prewarning for Sensor Networks with Multi Microenvironments^{*}

Yinglong Li^{1,2,3}, Hong Chen^{1,2,**}, Suyun Zhao¹, and Shangfeng Mo^{1,2,3}

¹ Key Laboratory of Data Engineering and Knowledge Engineering, MOE, China

² School of Information, Renmin University of China, Beijing, China

³ Hunan University of Science and Technology, Xiangtan, China

{liyinglong, chong, zhaosuyun, moshangfengxy}@ruc.edu.cn

Abstract. Early detecting the approaching events is the primary way of minimizing their damages in the sensor-based systems. The majority of existing approaches of event description and detection rely on using crisp raw sensory data, which requires large amount of data transmission as well as is memory-consuming, moreover, these approaches are only applicable to homogeneous sensor networks. This paper describes a novel efficient framework for event prewarning in sensor networks with multi microenvironments, which mainly includes a simple and practical data preprocessing method, Node-level Noteworthy Event (NNE) detection algorithm, event probability encodings of NNEs and two distributed Node-level Alert Event (NAE) detection algorithms. We demonstrate our algorithms by experimentally evaluating their performance in various scenarios using real and synthetic data. Our NAE detection algorithm by leveraging spatial correlation only requires a small amount of data transmission and can detect over 90% of NAEs with few false negatives.

Keywords: microenvironment, sensor network, event probability, node-level noteworthy event, node-level alert event.

1 Introduction

Sensor networks can be viewed as energy constrained distributed database systems, and their tasks are monitoring physical environments, processing sensed information, and forwarding results to base stations (sink). Since data transmission consumes most of the energy, a significant challenge for such kind of systems is to design reliable, energy-efficient data processing algorithms to maximize the lifetime of sensor networks. Moreover, the low amount of data transmission also contributes to quick response time and less signal interference in wireless communication.

Event detection is a common required service in sensor network based applications such as environmental monitoring [1,2] and object tracking [3,4], which

^{*} This work was supported by the National Natural Science Foundation of China (grant no.: 61070056, 61033010 and 61202114).

^{**} Corresponding author.

has attracted increasing research attention. Various studies on event description and detection approaches have been reported in the literature [1]-[8]. However, these approaches rely on using raw sensor measurements, which results in large data transmission inside the network and long response time. Moreover, the aforementioned approaches are only applicable to sensor networks with single microenvironment, while in many sensor network applications, the monitoring area consists of multi MicroEnvironments (MEs), where the characteristics of each ME are different. For example, different types of materials with different ignition points (e.g., burning point of methanol is less than 30°C and fire point of turpentine is under 65°C) are placed in different areas of a warehouse, where each resulting segment (area) will have different tolerance for warehouse fire. Each of these segments of the warehouse, then, represents a ME as shown in Fig. 1. When users check the k most likely impending fire locations in such kind

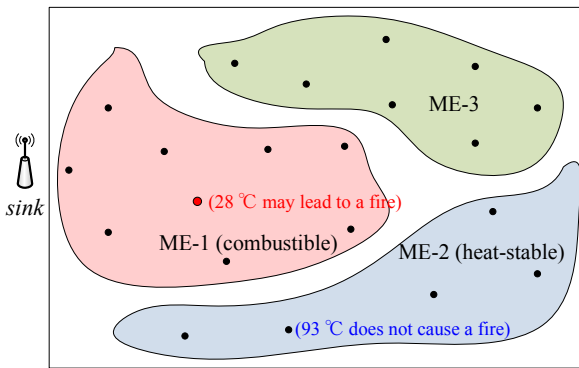


Fig. 1. An example of three MEs in a warehouse

of sensor networks, if the existing event detection methods (such as the weighted voting schemes in [7,8]) are used, they will lead to erroneous results, since the small temperature values in the combustible MEs might be the top k results.

In this paper we develop an event prewarning framework for sensor networks with multi MEs in a scenario of relatively short sampling period, which has not been studied before. The contributions of this paper are summarized as follows:

- A conceptually simple, yet practically effective data preprocessing approach is given to eliminate erroneous sensory data.
- The definition of Node-level Noteworthy Event (NNE) and the detection algorithm of NNE are proposed.
- Two distributed node-level alert event detection algorithms are devised, and extensive simulations are performed to validate our motivation.

The remainder of the paper proceeds as follows. Our network model is explained in Section 2, and Section 3 describes the data preprocessing approach. Then, node-level noteworthy event algorithm is proposed in Section 4. Section 5 details

two node-level alert event detection algorithms. Meanwhile, simulation results are presented in Section 6. Finally, Section 7 presents conclusions and future research direction.

2 Network Model

This section gives an overview of our network model. It also covers the almost reasonable assumptions.

In our geographical location based clustering networks, there are two types of nodes which are regular sensor node and Cluster Head (CH) sensor node respectively in the network. Each node (we use “node” to refer to a regular sensor node or CH sensor node) has a unique identification (*nodeID*). Tentative CH nodes are selected from regular nodes mainly based on their remaining energy via a non-probabilistic fashion [9]. In addition to collect its own sensory data, every CH node manages the topology information of all regular nodes within its cluster, as shown in Fig. 2(a).

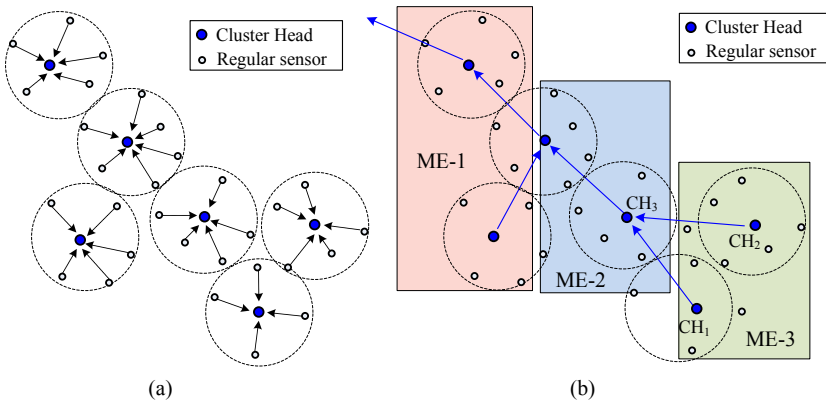


Fig. 2. (a) Cluster-based network. (b) CH-TAG network with three MEs.

CH nodes communicate with each other hop by hop. We assume that a data aggregation tree (i.e., TAG [10]) is constructed over these CH nodes, and eventually connected to base station, as is shown in Fig. 2 (b). We call the CH nodes based aggregation tree as CH-TAG. Our network also consists of multi MEs, and an example of three MEs in a CH-TAG network is shown in Fig. 2(b).

3 Data Preprocessing

Event detection techniques need to prevent erroneous data from influencing the detection reliability. Most of existing outlier (here mainly refers to errors and noise) detection [5] approaches relay on using the neighboring information, which

leads to a large amount of data transmission and is not suitable for energy constrained sensor networks. Therefore, we give a node-level erroneous data detection approach only leveraging the *temporal correlation* of the local measurements within the recent sampling periods.

The erroneous sensory data we concentrated here mainly come from FAULTY (failed) nodes and noisy data in normal working nodes. Therefore, our goal of data preprocessing is to identify and prune faulty nodes and eliminate noisy data in functioning nodes.

Definition 1 (FAULTY Node Rule). Most of the data in the FAULTY node significantly deviate from the normal pattern of sensed data [11]. A simple and effective method is that we can know whether a sensor node is a FAULTY node by checking the standard deviation and temporal correlation of the sensor readings within recent time window. FAULTY nodes should be pruned as early as possible.

Definition 2 (NOISY Data Rule). In functioning node, *smoothing factor* [12] based approach (*improved smoothing factor*) is used to identify and eliminate the noisy data, where the *improved smoothing factor* can be defined as follows: Given:

- a set of sensor measurements MS
- a dissimilarity function $D : D(MS) \rightarrow R^+$, where R^+ means positive real number.

$$ISF(M_i) = D(MS) - D(MS - \{M_i\}). \tag{1}$$

The *improved smoothing factor* (ISF) indicates how much the dissimilarity can be reduce by removing an element (sensor measurement) from the set MS . Observe that ISF may be negative for some measurement M_i if the dissimilarity of $MS - \{M_i\}$ is higher than that of the original set MS .

Example. Let

- the set MS be the set of float values 32.6, 31.9, 11.7, 32.8, 93.5, 33.3;
- the dissimilarity function $D : D(MS) \rightarrow R^+$ be the variance of the numbers in the set, i.e., $\sum_{i=1}^n (x_i - \bar{x})^2/n$, then we get $D(MS) = 646.22$

By computing the ISF for each measurement M_i via (1), we get:

Table 1. ISF calculation for each measurement M_i

M_i	$MS - \{M_i\}$	$D(MS - \{M_i\})$	$ISF(M_i)$
32.6	{31.9, 11.7, 32.8, 93.5, 33.3}	764.69	-118.47
31.9	{32.6, 11.7, 32.8, 93.5, 33.3}	762.32	-116.1
11.7	{32.6, 31.9, 32.8, 93.5, 33.3}	592.64	53.58
32.8	{32.6, 31.9, 11.7, 93.5, 33.3}	803.93	-157.71
93.5	{32.6, 31.9, 11.7, 32.8, 33.3}	70.43	575.79
33.3	{32.6, 31.9, 11.7, 32.8, 93.5}	766.82	-120.6
32.6	{31.9, 11.7, 32.8, 93.5, 33.3}	764.69	-118.47

Thus, the measurement 93.5 has the highest probability to be the noisy data as its *ISF* is the largest, followed by the element 11.7, as shown in Table 1. The measurement whose *ISF* is over a preset threshold can be eliminated, and the *improved smoothing factor* provides tunable accuracy guarantees based on users requirement that how much noise data should be eliminated.

4 Node-Level Noteworthy Event

Definition 3 (Node-level Noteworthy Event (NNE)). node u is a node-level noteworthy event once us measurements make the *event probability* (ep) of u reach a preset threshold value $thrd$ (usually 0.5).

4.1 NNE Detection Algorithm

After pruning FAULTY nodes and eliminating noisy data in functioning nodes, each node calculates its ep based on its measurements and ep function, where the ep function varies in different MEs, or the function parameters are different in each ME, which can be defined based on empirical or domain knowledge. In the warehouse fire monitoring system with multi MEs, if two properties (*temperature* (t) and *humidity* (h)) are mainly used for defining fire event, then an example of ep function can be defined as follows.

$$ep(t, h) = \begin{cases} 0, & t \leq t_{as}, h... \\ \omega \times avg(t)/(t_{ig} - t_{as}) + (1 - \omega)f(avg(h)), & t_{as} < t < t_{ig}, h... \\ 1, & t \geq t_{ig}, h... \end{cases} \quad (2)$$

Where avg means getting the average value of measurements. t_{as} means the absolute safe temperature, which by no means causes a fire, and t_{ig} means the ignition temperature, which varies in different MEs. The first part of (2) demonstrates that the higher temperature leads to higher probability of fire, and the $f(h)$ in the second part of (2) is a function of how *humidity* affect the fire event, which *humidity* usually has a negative impact on fire event. ω is a tunable value, which determines how much *temperature* and *humidity* influence the fire event respectively. Due to the different $[t_{as}, t_{ig}]$ in each ME, (2) can describe the characteristics of fire occurrence in different MEs. Therefore, each node in different MEs can determine whether it is a NNE by the Algorithm 1.

In Algorithm 1, $ep(i)$ is the event probability of node i , and $pFlag(i)$ is a flag to determine whether node i can be pruned. If $pFlag(i)$ is "TRUE", then node i is not a noteworthy node and should be pruned as early as possible.

4.2 ep Encoding of NNEs

In order to reduce the data transmission, we discretize the ep values and give their encodings. We use linguistic characters to define the severity grades (*GRADE*) of NNEs and each *GRADE* corresponds to a non-uniform ep sub range $\mathfrak{R}(ep)$.

Algorithm 1. NNE Detection

```

1: for each node  $i$  do
2:    $pFlag \leftarrow FALSE$ ;
3:   if  $i$  is FAULTY node then
4:      $pFlag \leftarrow TRUE$ ;
5:   else
6:     eliminate the noisy data
       using NOISY Data Rule;
7:     calculate the  $ep(i)$ ;
8:   if  $ep(i) < thrd$  then
9:      $pFlag \leftarrow TRUE$ ;
10:  else
11:     $i$  is a NNE;
12:  end if
13: end if
14: end for

```

When *GRADE* is closer to the event occurrence threshold, the interval of $\mathfrak{R}(ep)$ is smaller.

For example, as shown in Table 2, we define eight *GRADE*s ('*a*', '*b*', '*c*', '*d*', '*e*', '*f*', '*g*' and '*h*') (the first column, not be fully presented due to the limited space) whose interpretations are shown in the second column with different $\mathfrak{R}(ep)$. The *ep* sub range interval of *GRADE* '*a*' is the smallest as shown in the last column, and the threshold *thrd* is 0.5, which means every node whose *ep* is below 0.5 (below '*h*') can be pruned. We also give 3-bits encoding for each *GRADE* to reduce the total data transmission, as is shown in the third column of Table 2.

Table 2. An Example of *ep* encoding based on non-uniform *ep* sub ranges

<i>GRADE</i>	Interpretation	Encoding	$\mathfrak{R}(ep)$
<i>a</i>	<i>Almost happened</i>	000	[0.98, 1.00]
<i>b</i>	<i>Very serious</i>	001	[0.95, 0.98]
<i>c</i>	<i>Serious</i>	010	[0.91, 0.95]
<i>d</i>	<i>Alert</i>	011	[0.86, 0.91]
...
<i>h</i>	<i>Noteworthy</i>	111	[0.50, 0.62]

The above non-uniform *ep* sub ranges $\mathfrak{R}(ep)$ can be obtained via mathematical models based on the actual situation of specified application. The accuracy of event prewarning might be higher if more *GRADE*s are used.

5 Node-Level Alert Event Detection

Definition 4 (Node-level Alert Event (NAE)). A sensor node u belonging to NNEs becomes a node-level alert event if it has relatively large *ep* and has at least one neighboring NNE. Here, the relatively large *ep* means that the *ep* of u meets application-specific threshold, such as 0.86 ('*d*') used later in this paper.

NAEs are the node-level prewarning events that we focus in this paper. And we propose two following NAE detection approaches.

5.1 NAE Detection without Considering Spatial Correlation

NAE detection scheduling without considering spatial correlations, denoted as NAED-noSC, can be described as follows. Each NNE whose *GRADE* reaches a predefined threshold (*'d'* in this paper) is forwarded to the base station as NAE. The main data in each message packet is “*GRADE* (Encoding of *GRADE*)” and “*nodeID*”, denoted as “*GRADE+nodeID*”. The spatial correlation of these NAEs can be checked to further confirm the approaching events (or event regions) according to the global network topology information at the base station. The data transmission of NAED-noSC is very small, but there might be false positives and false negatives.

Accuracy analysis of NAED-noSC. In NAED-noSC, since every NNE with a “*Alert*” *GRADE* (*'d'*) is detected as NAE, there might be isolated NAEs (without any neighboring NNEs) leading to false positives. While those NNEs whose *GRADE*s are little bit smaller than *'d'* but have neighboring NNEs might not be detected as NAEs in NAED-noSC, which leads to false negatives.

5.2 NAE Detection by Leveraging Spatial Correlation

Existence of *spatial correlation* implies that the readings from sensor nodes geographically close to each other are expected to be largely correlated. We would be more confident that there is an actual fire if there are at least two neighboring nodes reporting high temperature and low humidity readings. There might be a false positive if there is just one NAE without any neighboring NNE. This is particularly reasonable in sensor networks where nodes are usually densely deployed.

The strategy here is that we update the *GRADE*s of spatial-correlated NNEs, and the Approach of *GRADE* Update of spatial-correlated NNEs (AoGU) can be described as follows. If there are two spatial-correlated NNEs, then both of their *GRADE*s are increased to a higher level *GRADE*s. For instance, there are three NNEs N_1 , N_2 and N_3 , and their *GRADE*s are *'h'*, *'f'* and *'e'* respectively. If N_1 and N_2 are spatial-correlated, and N_2 is N_3 's neighboring NNE, but there is no spatial correlation between N_1 and N_3 , then their *GRADE*s after grade update are *'g'*, *'d'* and *'d'* respectively. The magnitude of *GRADE* update is tunable, which is specified depending on the application scenario.

NAE Detection by Leveraging Spatial Correlations (NEAD-bySC).

Firstly, every CH node gets all the NNEs in its local cluster, and then each leaf CH node (CH-TAG based network) forwards all the NNEs in its cluster to its parent CH node. When a non-leaf CH node Q has obtained all the NNEs from its local cluster and its child clusters, it Checks the Spatial Correlation of these NNEs and Updates the *GRADE*s of spatial-correlated NNEs (CSC-UG) based on AoGU. Finally, all the non-isolated NAEs in Q 's child clusters are forwarded to base station as NAEs, and then Q forwards all the NNEs in its local cluster to its parent cluster, until all the non-leaf CHs finished the CSC-UG and NAE detection. More details of NAED-bySC are described in Algorithm 2.

Algorithm 2. NAED-bySC

```

Step 1. NNEs forwarding
1: for each NNE  $i$  do
2:    $uFlag(i) \leftarrow FALSE$ ;
3:    $i$  be forwarded to its local CH node;
4: end for
5: for each leaf CH node  $i$  do
6:    $i$  forwards all the NNEs to its parent
   CH node;
7: end for

Step 2. CSC-UG and NAE detection
1: for each non-leaf CH node  $i$  do
2:   CSC-UG of  $i$ 's NNEs is done;
3:   if  $GRADE(NNE_j)$  be updated then
4:      $uFlag(j) \leftarrow TRUE$ ;
5:   end if
6:   for each NNE  $k$  in  $i$  do
7:     if  $GRADE(k) \geq 'd' \& \& uFlag(k) ==$ 
    $TRUE$  then
8:        $k$  is detected as a NAE;
9:     end if
10:  end for
11:   $i$  forward all the NNEs to its parent
   CH node;
12: end for

```

Where $uFlag(i)$ is a flag to demonstrate whether the $GRADE$ of NNE i be updated, which is used for identifying whether i is an isolated NNE. If $uFlag(i)$ is “FALSE”, then i is a isolated NNE. In Algorithm 2, the main data structure of NNE and NAE message packet of line 3 in *step 1* and line 8 in *step 2* are both “ $GRADE + nodeID$ ”, and the main data structure of NNE message packet in line 11 of *step 2* is “ $GRADE + nodeID + LOCATION$ ”, where the “ $LOCATION$ ” is the geographic coordinates which are used for spatial correlation detection.

Example. As shown in Fig. 3, there are three clusters with cluster head CH_1 , CH_2 and CH_3 . Firstly, all NNEs are forwarded to its local CH node, such as NNE S_1 , S_3 and S_5 in CH_1 cluster are forwarded to CH_1 . Then each leaf CH node forwards their NNEs to its parent CH node, such as CH_1 and CH_2 forward their NNEs to CH_3 . Then CH_3 does CSC-UG based on the AoGU. S_{11} and S_3 are spatial-correlated, and S_6 is the neighboring node of S_5 and S_7 , as shown the green dotted lines in Fig. 3. All the $GRADE$ s of these NNEs are updated based on AoGU, as shown in Fig. 3. S_1 is not a NAE because it is an isolated NNE although it has a high $GRADE$, and all the non-isolated NAEs in CH_3 's child clusters (cluster CH_1 and CH_2) are forwarded to base station as NAEs. Finally, CH_3 forwards all the NNEs in its local cluster to its parent cluster, until all the CHs finishes the NAE detection.

Accuracy Analysis of NAED-bySC. In NAED-bySC, some spatial-correlated NNEs become NAEs due to their $GRADE$ s update, which reduces the false negative. For instance, in Fig. 3, the NNE S_{11} (‘ e ’) in cluster CH_3 is spatial-correlated with S_3 in cluster CH_1 , so the $GRADE$ of S_{11} is updated to ‘ d ’ and S_{11} becomes a NAE, while S_{11} is a false negative in NAED-noSC. All the isolated NNEs will not be detected as NAEs in NAED-bySC, so there is no false positive in NAED-bySC. For example, the NNE S_1 with $GRADE$ ‘ c ’ is not detected as NAE since

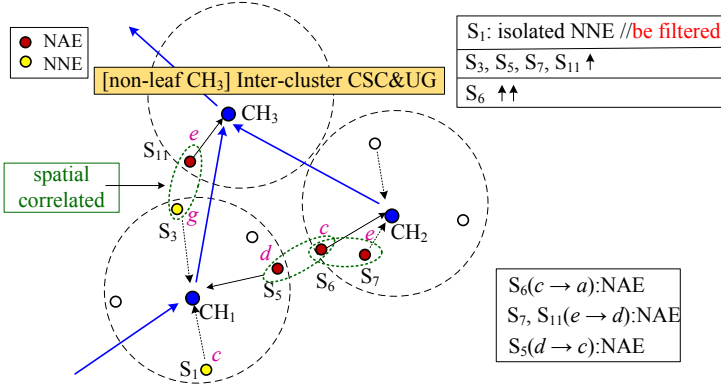


Fig. 3. Inter-cluster CSC-UG and NAE detection

it is an isolated NNE without any neighboring NNE. Therefore, the false rate of NAED-bySC is very low.

Tradeoff. Generally speaking, larger scope of inter-cluster CSC-UG contributes higher accuracy of NAE detection, but the data transmission will be larger. There is a tradeoff between data transmission and accuracy. In NAED-bySC, we only consider the spatial correlation of the nodes from parent-child clusters as well as the nodes from brother-brother clusters, without other spatial correlation. Since the spatial correlation we have considered accounts for the most of the all spatial correlation, NAED-bySC can guarantee high accuracy, while requires small amount of data transmission. More experimental analysis is given in later section 6.

6 Performance Evaluations

We use OMNET++ [13] to evaluate our algorithms in terms of data transmission and accuracy of NAE detection. We use part of real data (*Surface Temperature* and *Relative Humidity*) from LUCE [14] and part of synthetic data which is generated and injected with outliers via scenarios of other MEs due to lack of real data of sensor networks with multi MEs.

Owing to the lack of research work on event detection in multi MEs sensor networks in literature, we performed comparison among three following methods:

- *Optimized Conventional Method* (OCM)The idea of conventional method [4, 9, 10, etc.] is that every node uses the information from its neighboring nodes to check whether it is an event node (or there is an event region). For a fair comparison, event probabilities instead of raw sensor readings are used. Since all the spatial correlation is checked, the OCM can get the true NAEs accurately.
- NAED-bySC (Algorithm 2).
- NAED-noSC.

6.1 Comparison of Data Transmission

Since data transmission consumes most of the energy in sensor networks, We evaluate the data transmission of OCM, NAED-bySC and NAED-noSC in four different networks, where 40, 80, 120 and 160 nodes are randomly deployed in four regions of $300 \times 300 \text{ m}^2$, $400 \times 400 \text{ m}^2$, $500 \times 500 \text{ m}^2$ and $600 \times 600 \text{ m}^2$ respectively. The communication radius R is 80 m. ep is a *float* number, which requires 4 bytes space (due to the 32-bit simulation platform). *LOCATION* consists of x-coordinate and y- coordinate (both *float* number), and *GRADE* is 3 bits, and *nodeID* is *short int* that requires 16 bits space. We did not consider the data transmission of other information in message packet, e.g., HEAD.

Among the three methods mentioned above, the data transmission of OCM is the largest, and the one of NAED-noSC is the smallest in our four networks. The average data transmission of NAED-noSC is 1.3% of the one of OCM, and the average data transmission of NAED-bySC is 5.5 percent of the one of OCM. And the average data transmission of NAED-noSC is 23.89% of the one of NAED-bySC, as is shown in Fig. 4(a). So our algorithms can reduce the data transmission greatly compared with OCM. Although the data transmission of NAED-noSC is the smallest, its accuracy is not satisfactory (more details will be described in Subsection 6.2).

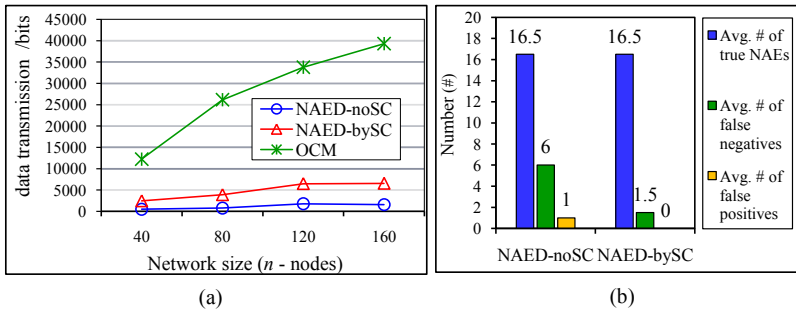


Fig. 4. (a) Comparison of data transmission. (b) Accuracy comparison.

6.2 Comparison of Accuracy

Besides good performance on data transmission, another important goal of event detection is accuracy. We study the accuracy performance of NAED-noSC, NAED-bySC and OCM based on our four networks mentioned in Subsection 6.1. Here accuracy is defined as:

$$1 - (\text{number of false positives} + \text{number of false negatives}) / \text{number of true NAEs}$$

As all the spatial correlation is examined for event detection in OCM, there is no false positive or false negative (accuracy is 100%). There are a few false positives in NAED-noSC, and the average (Avg.) number (#) of false positives in our four networks was 1 (the total number of true NAEs is 16.5), shown as in

Fig. 4(b). However there are relatively large number of false negatives in NAED-noSC, and the Avg. # of false negatives in the four networks was 6; There was no false positive in NAED-bySC, however there are a few false positives in NAED-bySC because that we do not consider all the spatial correlation. The Avg. # of false negatives of NAED-bySC in our four networks was 1.5, and other results are shown in Fig. 4(b). These experimental results verify the correctness of the accuracy analysis of NAED-noSC and NAED-bySC in Subsection 5.1 and Subsection 5.2.

The accuracies of NAED-bySC in the four networks were 90.9%, 93.75%, 90.91% and 88.89% respectively, and the average accuracy of NAED-bySC was more than 90% (91.11% actually). The accuracy of NAED-noSC was relatively lower than NAED-bySC, namely 54.55%, 62.5%, 59.09% and 55.56% respectively in our four networks, which are shown in Fig. 5(a).

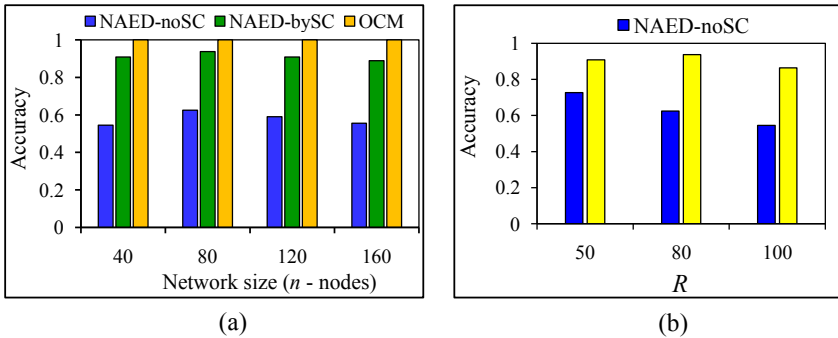


Fig. 5. (a) Accuracy comparison ($R = 80$). (b) Accuracy comparison (80 nodes).

With the increase of node density, the accuracy of NAED-noSC decreases. In our 80 nodes network, with the increase of R (equivalent to the increase of node density), the accuracy of NAED-noSC decreased, shown as the blue column chart in Fig. 5(b). This is because the increase of node density leads to the increase of spatial correlation, thus NAED-noSC will miss more true NAEs. The variation of node density (or the change of communication radius in the same network) almost has no effect on the accuracy of NAED-bySC. This is because that the increase of spatial correlation among nodes contributes to the increase of true NAEs, meanwhile, more NAEs will be detected in NAED-bySC. Therefore, there is almost no causal relationship between accuracy and node density, which is shown as the yellow column chart in Fig. 5(b).

7 Conclusions

This paper presents a novel efficient framework for event prewarning in sensor networks with multi MEs, which mainly includes a simple and practical data preprocessing method, NNE detection algorithm, event probability encodings of

NNEs and two distributed NAE detection algorithms (NAED-noSC and NAED-bySC). Experimental evaluation demonstrates our approach reduces the data transmission greatly compared with conventional approaches, and NAED-bySC guarantees good detection accuracy. When the network situation is bad, namely when the proportion of NNEs with relatively high *GRADES* is large, NAED-noSC is a more suitable approach with few false negatives. Our on-going work is to enhance our approach to deal with the phenomenon in which the alert event enlarges or disappears with time elapsed.

References

1. Aslan, Y.E., Korpeoglu, I., Ulusoy, Ö.: A Framework for Use of Wireless Networks in Forest Fire Detection and Monitoring. *Computers, Environment and Urban Systems* 36, 614–625 (2012)
2. Kapitanova, K., Son, S.H., Kang, K.-D.: Using Fuzzy Logic for Robust Event Detection in Wireless Sensor Networks. *Ad Hoc Netw.* 10, 709–722 (2012)
3. Hubbell, N., Han, Q.: DRAGON: Detection and Tracking of Dynamic Amorphous Events in Wireless Sensor Networks. *IEEE T. on Parall. and Distr.* 23, 1193–1204 (2012)
4. Shih, K.-P., Wang, S.-S., Chen, H.-C., Yang, P.-H.: COLLECT: Collaborative Event Detection and Tracking in Wireless Heterogeneous Sensor Networks. *Comput. Commun.* 31, 3124–3136 (2008)
5. Zhang, Y., Meratnia, N., Havinga, P.: Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Commun. Surveys & Tutorials.* 12, 159–170 (2010)
6. Premkumar, K., Prasanthi, V.K., Anurag, K.: Delay Optimal Event Detection on Ad Hoc Wireless Sensor Networks. *ACM T. on Sensor Netw.* 8, 1–39 (2012)
7. Luo, X., Dong, M., Huang, Y.: On Distributed Fault-tolerant Detection in Wireless Sensor Networks. *IEEE Trans. on Comput.* 55, 58–70 (2006)
8. Ould-Ahmed-Vall, E., Ferri, B.H., Riley, G.F.: Distributed Fault-tolerance for Event Detection Using Heterogeneous Wireless Sensor Networks. *IEEE T. on Mobile Comput.* 11, 1994–2007 (2012)
9. Taheri, H., Neamatollahi, P., Younis, O.M., Naghibzadeh, S., Yaghmaee, M.H.: An Energy-Aware Distributed Clustering Protocol in Wireless Sensor Networks Using Fuzzy Logic. *Ad Hoc Netw.* 10, 1469–1481 (2012)
10. Samuel, M., Michael, J.F., Joseph, H., Wei, H.: TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks. In: *Fifth Symp. Oper. Syst. Desi. and Impl.*, pp. 131–146. ACM (2002)
11. Sharma, A.B., Golubchik, L., Govindan, R.: Sensor Faults: Detection Methods and Prevalence in Real-world Datasets. *ACM T. on Sensor Networks.* 6, 1–34 (2010)
12. Arning, A., Agrawal, R., Raghavan, P.: A Linear Method for Deviation Detection in Large Database. In: *ACM KDD*, pp. 164–169. ACM (1996)
13. OMNET++ platform, <http://www.omnetpp.org>
14. LUCE, http://sensorscope.epfl.ch/index.php/Environmental_Data