A Novel Event Detection Framework for Wireless Sensor and Actor Networks

Haoyun Wang[1,2], Yuhan Fei[1], Jiaojiao Liu[1], Yingjun Xiong[1,2], Shougang Ren[1,2], and Huanliang Xu[1,2]
[1.] College of Information Science and Technology, Nanjing Agricultural University Nanjing 210095, China
[2.] Collaborative Innovation Center of Meat Production and Processing
Email: {wanghy, 2014114004, 2012114003, xyj, rensg, huanliangxu}@njau.edu.cn

Abstract —Wireless Sensor and Actor Network (WSAN) can observe the physical world using event detection methods by sensor nodes and implement appropriate operation by actor nodes. Compared with Wireless Sensor Network (WSN), WSAN set high requests for real-time, collaboration, mobility, etc. Therefore, the event detection methods applied to WSN cannot be adopted for WSAN. In this paper, we present a three-layer Cooperative Event Detection Framework (3-CEDF) for WSAN. Firstly, sensor nodes cooperate with each other to affirm the occurrence of events via voting mechanism. Then, cluster-head nodes use a modified K-means algorithm and a proposed type matching mechanism to detect the event types with the boundary information of event data. Finally, actor nodes get the priority of events processing using a modified k-nearest neighbor algorithm with the slope similarity measure. The performances of effectiveness, real-time, mobility and communication traffic for 3-CEDF are obtained through analysis and simulations, when 3-CEDF is compared to the centralized event detection method adopted for WSN.

Index Terms—Wireless sensor and actor network, event detection, cooperative algorithm, type matching, similarity measure

I. INTRODUCTION

Wireless Sensor and Actor Network (WSAN) [1] derives from Wireless Sensor Network (WSN) [2]. The concept of WSAN, proposed by I. F. Akyildiz [3] in 2004, has caused the extensive attention of scholars. WSAN consists of sensor nodes and actor nodes. We can deploy sensor nodes massively in the monitored area. In WSAN, sensor nodes detect environmental events and sent event data to corresponding actor nodes, then actor nodes can respond quickly and take real-time control of the surrounding environment. WSAN can observe the physical world, process data, make decision and implement appropriate operation. Therefore, WSAN has been used more widely in many applications [4], such as battlefield surveillance system, building microclimate control system, nuclear biochemical attack detection system, home automation system and environmental testing system, etc.

Compared with event detection for WSN, WSAN sets higher requests for real-time [5], collaboration [6], mobility [7], etc. To make sure that actor nodes can take right action, the event data reported from sensor nodes to actor nodes must be valid. Therefore, the end-to-end delay between sensor nodes and actor nodes needs to be shorter. Besides, the algorithms and protocols in WSAN had better to be collaborative due to the distributed deployment of nodes. In particular, actor nodes in WSAN are mobile, for example, the robots in the distributed application or the soldiers equipped with digital transceivers on the battlefield. Moreover, the energy consumption of nodes need to be reduced and the lifetime of WSAN can be prolonged [8].

When events happening, sensor nodes detect them immediately in WSAN and report them to actor nodes. As soon as receiving the event data, actor nodes take actions to control the events. Because the architecture of WSAN is different from WSN, we cannot apply directly the existing event detection algorithms for WSN to WSAN. In this paper, We proposes a three-layer Cooperative Event Detection Framework (3-CEDF) for WSAN. Firstly, sensor nodes cooperate to detect events and report the event data to cluster-head nodes. Secondly,, cluster-head nodes use a modified K-means algorithm [9] to classify the event data received with the boundary information and detect the event type by matching the predefined event model and call corresponding actor nodes to move toward to the event area. Finally, actor nodes get the priority of events processing and take appropriate actions. Considering the features of WSAN, 3-CEDF meets the requirements of effectiveness, real-time and energy efficiency.

The current researches on event detection algorithms mostly focus on WSN applications. The cooperative detection algorithms applied in WSAN are not presented yet. This paper analyzes the performance of effectiveness, real-time, mobility and communication traffic for 3-CEDF, as well as the influence of different key parameters to its performance compare with the proposed detection methods for WSN. The experiment results illustrate the proposed algorithm show a better performance compare with traditional event detection algorithm for WSN.

The rest of the paper is organized as follows. In Section 2 we talk about related work, In Section 3 we

Manuscript received June 15, 2015; revised January 4, 2016.
This work is supported by Fundamental Research Funds for the Central Universities (No.KYZ201421); The Agricultural Project for New Variety, New Technology and New Model of Jiangsu Province (SXGC[2014]309); Government sponsored project under the science and technology development initiative : Research and demonstration of electronic food safety traceability (2015BAK36B05).
Corresponding author email: huanliangxu@njau.edu.cn
doi:10.12720/jcm.11.1.1-14

propose the concepts and definitions related to 3-CEDF. We describe the cooperative event detection algorithm implemented by adjacent sensor nodes in Section 4, and the detection algorithm executed by sensor nodes and cluster-head nodes in Section 5. In Section 6 we provide the algorithm performed collaboratively by cluster-head nodes and actor nodes. In Section 7 we analyze the performance of the proposed algorithms. Section 8 shows the simulation results. Finally, we concludes this paper and discusses the future of work in Section 9.

II. Related Work

Although WSN has been deployed in many applications [13], more and more applications require the combination of actor nodes and sensor nodes. In WSAN, sensor nodes collect information from physical environment, and actor nodes take actions to change the environment. Cooperation is an important characteristic of WSAN [14]. Different from WSN in which sink nodes take charge of data collection and mining, WSAN needs cooperation mechanism between sensor nodes and actor nodes to realize the performance goals of applications. The coordination between sensors and actors can establish paths to transmit data and help actors to find related positions or precise locations of events. After receiving event data, actor nodes need to cooperate with each other to make correct decisions and finish related tasks [15]. In this paper, 3-CEDF utilizes three-layer cooperation relations in WSAN to make normal sensor nodes, cluster-head nodes and actor nodes work together, which reducing energy consumption and prolonging working life of WSAN. At the same time, all kinds of nodes use their own sliding windows [16] to ensure the real-time requirement of event detection for WSAN.

At present, the event detection method for WSN can be classified into three types: detection methods based on threshold [17], methods based on spatial-temporal model [18] and methods based on pattern [19]. If sensor readings are more than default threshold, detection methods based on threshold will determine the occurrence of events. Detection methods based on spatial-temporal model use the correlation between time and space of sensor reading to make the prediction of event happening. Methods based on pattern can recognize what happens by model matching. The first two kinds of event detection methods have certain limitations. Detection methods based on threshold need long-time sample series to train machine-learning algorithms and obtain optimized threshold. However, this method cannot meet the real-time requirement of event detection for WSAN. Detection methods based on spatial-temporal model need to build correct prediction model and the parameters of prediction model are difficult to adjust according to the specific WSAN application field. Different with WSAN, WSAN contain actor nodes, which execute predefined control strategies according to the types of detected events. Therefore, the detection method using pattern matching can be adopted for WSAN.

In the event detection algorithms for WSN, sensor nodes need to proceed data fusion [20] several times when uploading perceived data. If some sensor nodes are failed or artificial damaged, their false readings will influence the outcome of data fusion and cause wrong decisions. In [21], an event detection method was presented which could get event eigenvalues through analyzing principal components of the data got by central nodes, but its computational complexity is high. In [22], detection vectors were injected actively to make cross validation with original data. However, that method could change the environment of event area. All above methods detect events in centralized manner. In [23], a distributed detection method based on local voting was proposed, which using majority rule to filter out error readings of sensor nodes and to ensure data accuracy. The paper also analyzed the time-space relativity of sensor readings obtained by sensor nodes in event region, nodes on event boundary and error nodes respectively. In this paper, 3-CEDF uses local voting mechanism to send sensor readings of center region to avoid the interference of false sensor readings, and this mechanism makes actor nodes control the development of happening events more effectively.

Sensor readings in WSAN form into time series. The efficiency and accuracy of time series data mining depend directly on the representation methods of time series. At present, main representations of time series involve DWT (Discrete Wavelet Transform) [24], PAA, PLA (Piecewise Linear Approximation) [25] and SAX (Symbolic Aggregate Approximation) [26]. Among them, PAA uses a fixed-width sliding window to obtain the subsequences and expresses the data in sliding window by mean values. PAA has less computational complexity and can be implemented on the resource-constrained sensor nodes. Besides, it can reduce the dimension of time series effectively, reserve the partial features of original data, and advantage the follow-up steps of event detection. In [27], time series clustering algorithm based on boundary information was presented. On that basis, upper bound and lower bound of time sequences were taken into consideration in [28]. By calculating the boundary of time series, more feature information and maximum fluctuation range of the subsequences are added in addition to mean values recorded by PAA. In [29], angle information of time series were used to calculate angle similarity distance, which can overcome the shortcomings of less robustness and unclear physical conception. However, it has a large amount of calculation.

In this paper, the above algorithm is simplified to calculate slope similarity distance between multiple time series received by actor nodes to analyze the development trend of happening events.

The performances of effectiveness, real-time, mobility and communication traffic for 3-CEDF are obtained through analysis and simulations, when 3-CEDF is compared to the traditional event detection method adopted for WSN. Algorithm show a better performance

compare with traditional event detection algorithm for WSN.

III. Concepts and Definitions

This paper proposes a three-layer Cooperative Event Detection Framework (3-CEDF) for WSAN. Firstly, sensor nodes cooperate to detect events. Once the sensor readings exceed the predefined threshold, sensor nodes ask their neighbors to vote and get the credit degree according to the voting results. If the degree value is more than 0.5, sensor nodes report the event data to cluster-head nodes. Secondly, cluster-head nodes use a modified K-means algorithm [9] to classify the event data received with the boundary information and get the probability distribution using statistical method. Then, cluster-head nodes detect the event type by matching the predefined event model and call corresponding actor nodes to move toward to the event area. Finally, actor nodes get the priority of events processing using a modified k-nearest neighbor algorithm (KNN) [10] with the slope information of event data. The logical diagram for 3-CEDF displayed in Fig. 1.

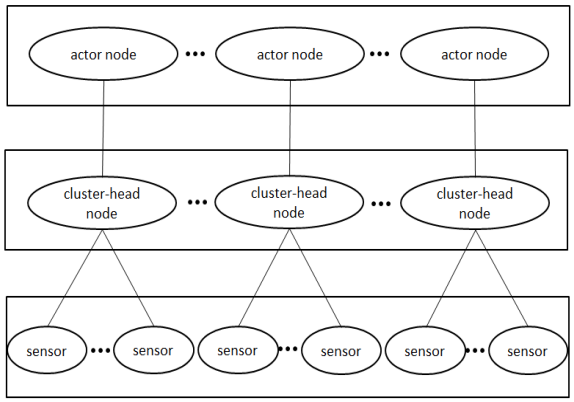


Fig. 1. The logical diagram for 3-CEDF

Assume that the monitored area is a two-dimensional space denoted by *R*. In *R*, there are n_s sensor nodes $s_{i,i\in\{1,...,n_S\}}$ and n_a actor nodes $a_{j,j\in\{1,...,n_a\}}$, which have d_a types. Every sensor node has d_s sensors and each type of actor nodes deals with different event. When the clustering of sensor nodes becomes steady, there are n_c clusters in *R* and each normal sensor node belongs to respective cluster. Sensor nodes can send sensory data through corresponding cluster-head nodes $c_{v,v\in\{1,...,n_c\}}$ to different actor nodes.

A. Multidimensional Vector Time Sequence

The attribute values of *R* change dynamically. Several sensors on sensor nodes sample the attribute values periodically. At time *t*, the attribute vector obtained by sensor node s_i can be defined as $\bar{m}_{s_i}(t)=(x_{s_i}^1(t),x_{s_i}^2(t),...,x_{s_i}^{d_s}(t))$. The multidimensional vector time sequence constituted by attribute vectors sampled from time t_1 to time t_j can be denoted as $\bar{\mathbf{m}}_{s_i}(t_1,t_j)=\{\bar{m}_{s_i}(t_1),\bar{m}_{s_i}(t_2),...,\bar{m}_{s_i}(t_j)\}$.

In order to detect events and their types from complex multidimensional vector time sequences, we divide the sequences into several subsequences, which can be used to implement further data mining, such as clustering, classifying, abnormity detection, etc. In this paper, sensor nodes can use a method similar to Piecewise Aggregate Approximate (PAA) [11] to split received time sequences with equal length, and the partitioning granularity is g_s. Bigger the g_s is, lower the sensitivity of subsequences is. However, if g_s is too small, data features will be easy to lose and energy consumption of data mining algorithms will increase.

Considering the real-time requirement for WSAN applications, three kinds of sliding windows are used on sensor nodes, cluster-head nodes and actor nodes to support on-line detection in this paper, of which the window sizes are W_s,W_c,W_a respectively. Longer the size of sliding window is, bigger the proportion of historical data in the sliding window is, and less sensitive to the change of latest data is. Nevertheless, if the size is too small, data features in sliding windows will be ignored easily, and energy consumption of maintaining sliding windows on nodes will be increased. So configuring g_s and W_s,W_c,W_a correctly will affect the requirement of real-time, accuracy and energy consumption for event detection algorithms for WSAN. These parameters should be adjusted to meet specific application requirements in practice.

B. Mean Vector Sequence

At time t_{W_s}, sensor node s_i uses the sliding window $\left[t_1,t_{W_s}\right]$, of which window size is W_s and step size is 1, to get the multidimensional vector time sequence $\bar{\mathbf{m}}_{s_i}(t_1,t_{W_s})=\{\bar{m}_{s_i}(t_1),\bar{m}_{s_i}(t_2),...,\bar{m}_{s_i}(t_{W_s})\}$, in which $\bar{m}_{s_i}(t_{j,j\in\{1,2,...,W_s\}})=(x_{s_i}^1(t_j),x_{s_i}^2(t_j),...,x_{s_i}^{d_s}(t_j))$. According to g_s, $\bar{\mathbf{m}}_{s_i}(t_1,t_{W_s})$ can be divided into θ_s isometric subsequences, and $\theta_s=W_s/g_s$. Based on PAA, the mean value of each subsequence will be calculated and the mean vector sequence $\bar{\mathbf{p}}_{s_i}(t_{W_s})=\{\bar{p}_{s_i,1}(t_{W_s}),\bar{p}_{s_i,2}(t_{W_s}),...,\bar{p}_{s_i,\theta_s}(t_{W_s})\}$ will be obtained. The j_{th} vector $\bar{p}_{s_i,j}(t_{W_s})$ in $\bar{\mathbf{p}}_{s_i}(t_{W_s})$ can be expressed as follow.

$$\bar{p}_{s_i,j}^d(t_{W_s})=\sum_{v=g_s(j-1)+1}^{g_s j}\bar{m}_{s_i}^d(t_v)/g_s \qquad (1)$$

In Eq. (1), $\bar{m}_{s_i}^d(t_v)$ stands for the d_{th} dimension element in $\bar{m}_{s_i}(t_v)$, $\bar{p}_{s_i,j}^d(t_{W_s})$ indicates the d_{th} dimension element in $\bar{p}_{s_i,j}(t_{W_s})$, and $\bar{p}_{s_i,j}(t_{W_s})=(\bar{p}_{s_i,j}^1(t_{W_s}),...,\bar{p}_{s_i,j}^d(t_{W_s}),...,\bar{p}_{s_i,j}^{d_s}(t_{W_s}))$.

C. Boundary Vector Sequence

When some event happen in the monitored area *R*, cluster-head node c_i will receive real-time boundary vector sequence $\bar{\mathbf{q}}_{s_i}$ from sensor node s_i, and judge the event type from it. Firstly, sensor node s_i uses sliding window $\left[t_1, t_{W_s}\right]$ to obtain multidimensional vector time sequence $\bar{\mathbf{m}}_{s_i}(t_1, t_{W_s})$ at time t_{W_s}. Then, $\bar{\mathbf{m}}_{s_i}$ is split into θ_s isometric subsequences, of which the upper bounds and lower bounds can be calculated to get the boundary vector sequence $\bar{\mathbf{q}}_{s_i}(t_{W_s}) = \{\bar{q}_{s_i,1}(t_{W_s}), \bar{q}_{s_i,2}(t_{W_s}), ..., \bar{q}_{s_i,\theta_s}(t_{W_s})\}$. The calculation method of the j_{th} vector $\bar{q}_{s_i,j}(t_{W_s})$ in $\bar{\mathbf{q}}_{s_i}(t_{W_s})$ is as follows.

$$\bar{q}^d_{s_i,j}(t_{W_s}) = \bar{u}^d_{s_i,j}(t_{W_s}) + \bar{l}^d_{s_i,j}(t_{W_s}) \cdot \mathbf{i} \tag{2}$$

$$\bar{u}^d_{s_i,j}(t_{W_s}) = \max\left\{\bar{m}^d_{s_i}(t_{g_s(j-1)+1}), \bar{m}^d_{s_i}(t_{g_s(j-1)+2}), ..., \bar{m}^d_{s_i}(t_{g_s j})\right\} \tag{3}$$

$$\bar{l}^d_{s_i,j}(t_{W_s}) = \min\left\{\bar{m}^d_{s_i}(t_{g_s(j-1)+1}), \bar{m}^d_{s_i}(t_{g_s(j-1)+2}), ..., \bar{m}^d_{s_i}(t_{g_s j})\right\} \tag{4}$$

In the above formulas, $\bar{q}^d_{s_i,j}(t_{W_s})$ stands for the d_{th} dimension element in $\bar{q}_{s_i,j}(t_{W_s})$, and $\bar{q}_{s_i,j}(t_{W_s}) = (\bar{q}^1_{s_i,j}(t_{W_s}), ..., \bar{q}^d_{s_i,j}(t_{W_s}), ..., \bar{q}^{d_s}_{s_i,j}(t_{W_s}))$. $\bar{u}^d_{s_i,j}(t_{W_s})$ and $\bar{l}^d_{s_i,j}(t_{W_s})$ indicate the upper bound and lower bound of the subsequence $\left\{\bar{m}^d_{s_i}(t_{g_s(j-1)+1}), \bar{m}^d_{s_i}(t_{g_s(j-1)+2}), ..., \bar{m}^d_{s_i}(t_{g_s j})\right\}$ respectively.

D. Slope Vector Sequence

Cluster-head nodes choose appropriate actor node to control the event development when detecting event type. An actor node can receive many scheduling requests from different cluster-head nodes. Therefore, actor nodes need to do similarity detection on the boundary vector sequences received from cluster-head nodes to ensure the execution priority of control policies. When actor node a_i receives boundary vector sequence $\bar{\mathbf{q}}_{c_i}(t)$ from cluster-head node c_i, it can calculate corresponding slope vector sequence $\bar{\mathbf{r}}_{c_i}(t) = \{\bar{r}_{c_i,1}(t), \bar{r}_{c_i,2}(t), ..., \bar{r}_{c_i,\theta_s-1}(t)\}$. The j_{th} vector $\bar{r}_{c_i,j}$ in $\bar{\mathbf{r}}_{c_i}(t)$ can be expressed as (5).

$$\bar{r}^d_{c_i,j}(t) = \begin{cases} (real(\bar{q}^d_{c_i,j+1}(t)) - real(\bar{q}^d_{c_i,j}(t)))/g_s, \\ \forall real(\bar{q}^d_{c_i,j+1}(t)) > real(\bar{q}^d_{c_i,j}(t)) \cap imag(\bar{q}^d_{c_i,j+1}(t)) \geq imag(\bar{q}^d_{c_i,j}(t)); \\ 0, \forall real(\bar{q}^d_{c_i,j+1}(t)) \geq real(\bar{q}^d_{c_i,j}(t)) \cap imag(\bar{q}^d_{c_i,j+1}(t)) \leq imag(\bar{q}^d_{c_i,j}(t)); \\ 0, \forall real(\bar{q}^d_{c_i,j+1}(t)) \leq real(\bar{q}^d_{c_i,j}(t)) \cap imag(\bar{q}^d_{c_i,j+1}(t)) \geq imag(\bar{q}^d_{c_i,j}(t)); \\ (imag(\bar{q}^d_{c_i,j+1}(t)) - imag(\bar{q}^d_{c_i,j}(t)))/g_s, \\ \forall real(\bar{q}^d_{c_i,j+1}(t)) \leq real(\bar{q}^d_{c_i,j}(t)) \cap imag(\bar{q}^d_{c_i,j+1}(t)) < imag(\bar{q}^d_{c_i,j}(t)). \end{cases} \tag{5}$$

In the above formulas, $\bar{r}^d_{c_i,j}(t)$ stands for the d_{th} dimension element in $\bar{r}_{c_i,j}(t)$, and $\bar{r}_{c_i,j}(t) = (\bar{r}^1_{c_i,j}(t), ..., \bar{r}^d_{c_i,j}(t), ..., \bar{r}^{d_s}_{c_i,j}(t))$. $Real(\cdot)$ indicates the real part of complex number, $imag(\cdot)$ stands for the imaginary part of complex number, and "$\cap$" stands for the intersection of two sets.

IV. Cooperative Event Detection Algorithm between Adjacent Sensors

This paper presents a new three-layer cooperative event detection framework for WSAN. Firstly, sensor nodes cooperate with each other to detect events. Once sensor nodes find anomalous readings, they will ask surrounding sensor nodes to vote and calculate the credibility degrees of happening events according to voting results. When confirming events happen, sensor nodes will send event data to cluster-head nodes to detect the types of events.

At time t_{W_s}, sensor node s_i obtains $\bar{\mathbf{m}}_{s_i}(t_1, t_{W_s})$ from sensor readings in the sliding window. By Eq. (1), sensor node s_i transforms $\bar{\mathbf{m}}_{s_i}(t_1, t_{W_s})$ to corresponding mean vector sequence $\bar{\mathbf{p}}_{s_i}(t_{W_s})$. With the moving of sliding window, sensor node s_i can get $\bar{\mathbf{p}}_{s_i}(t_{W_s})$ and $\bar{\mathbf{p}}_{s_i}(t_{W_s+1})$ respectively at two adjacent time points. When the similarity degree between the two sequences is over threshold δ_s, sensor node s_i will estimate that some event is happening. The similarity of two mean vector sequences can be calculated as follows.

$$D_{s_i}(\bar{\mathbf{p}}_{s_i}(t_{W_s+1}), \bar{\mathbf{p}}_{s_i}(t_{W_s})) = \left\|\bar{\mathbf{p}}_{s_i}(t_{W_s+1}) - \bar{\mathbf{p}}_{s_i}(t_{W_s})\right\| = \frac{1}{\theta_s \cdot d_s}\sum_{j=1}^{\theta_s}\sum_{v=1}^{d_s}\left|\bar{p}^v_{s_i,j}(t_{W_s+1}) - \bar{p}^v_{s_i,j}(t_{W_s})\right| \tag{6}$$

When sensor node s_i finds some anomaly, it will ask its neighbor nodes to vote. If neighbor node s_j also deems there is an anomaly, s_j will returns 1 to affirm. Otherwise, s_j will return 0 to deny. After sensor node s_i receives vote results from all the neighbor nodes, it will confirm some event is truly happening if support rate v_{s_i} is over δ_v. The value of δ_v is usually set over 0.5. If v_{s_i} is just over zero, sensor readings of node s_i indicate an isolated event which is caused by various errors and interferences. Moreover, if v_{s_i} equals to 0.5, sensor node s_i is probably at the edge of event area, and the data obtained by s_i cannot reflect the real state of happening event.

If sensor node s_i detects some anomaly at time t_j, it will confirm the occurrence of some event after receiving voting results from all the neighbor nodes at time t_v. Then, node s_i will transfers event-related boundary

vector sequence $\bar{\mathbf{q}}_{s_i}(t_v)$ to corresponding cluster-head node c_i. Table I provides the steps of Cooperative Event Detection Algorithm Between Adjacent Sensors (CEDBAS).

TABLE I: PROCEDURES OF CEDBAS

Let W_s be the sliding window of sensor node s_i;
Let $\bar{\mathbf{m}}_{W_s}(t_{j-W_s}, t_{j-1})$ and $\bar{\mathbf{m}}_{W_s}(t_{j+1-W_s}, t_j)$ be the samples at time t_{j-1} and time t_j;
Let $\bar{\mathbf{p}}_{s_i}(t_{j-1})$ and $\bar{\mathbf{p}}_{s_i}(t_j)$ be the mean vector sequences calculated from $\bar{\mathbf{m}}_{W_s}(t_{j-W_s}, t_{j-1})$ and $\bar{\mathbf{m}}_{W_s}(t_{j+1-W_s}, t_j)$;
Let $\bar{\mathbf{q}}_{s_i}(t_v)$ be the boundary vector sequence calculated from $\bar{\mathbf{m}}_{W_s}(t_{v+1-W_s}, t_v)$ at time t_v;

1: if ($\left\|\bar{\mathbf{p}}_{s_i}(t_j) - \bar{\mathbf{p}}_{s_i}(t_{j-1})\right\| > \delta_s$) then
2: ask neighbor nodes to vote;
3: when a new message arrives
4: if (s_i is asked to vote) then
5: send 1 to the neighbor node;
6: end if
7: when all votes arrive at time t_v
8: if ($v_{s_i} > \delta_v$) then
9: report $\bar{\mathbf{q}}_{s_i}(t_v)$ to cluster head c_i;
10: end if
11: elseif ($\left\|\bar{\mathbf{p}}_{s_i}(t_j) - \bar{\mathbf{p}}_{s_i}(t_{j-1})\right\| \le \delta_s$) then
12: when a new message arrives
13: if (s_i is asked to vote) then
14: send 0 to the neighbor node;
15: end if
16: end if

V. COOPERATIVE EVENT DETECTION ALGORITHM BETWEEN SENSORS AND CLUSTER-HEADS

When sensor node s_i finds some event, node s_i will cooperate with cluster-head node c_i to detect the event type and choose appropriate actor node to handle the event. Cluster-head node c_i receives event-related boundary vector sequences from several normal sensor nodes using the sliding window of which window size is W_c and step size is W_s. The data from the sliding window will be analyzed using modified *K*-means clustering algorithm.

For clustering analysis, cluster-head node needs to calculate the similarity between two boundary vector sequences. Cluster-head node c_i receives boundary vector sequence $\bar{\mathbf{q}}_{s_i}(t_{m,m\in\{1,2,\ldots,W_c\}})$ and $\bar{\mathbf{q}}_{s_j}(t_{n,n\in\{1,2,\ldots,W_c\}})$ from sensor node s_i and s_j separately during the time interval $\left[t_1, t_{W_c}\right]$. The similarity of boundary vector sequences can be calculated by Eq. (7).

Two boundary vector sequences can be merged into one independent boundary vector sequence if the similarity is high. The combining method of boundary vector sequences is shown as Eq. (8).

$$
\begin{aligned}
D_s(\bar{\mathbf{q}}_{s_i}(t_m), \bar{\mathbf{q}}_{s_j}(t_n)) &= \left\|\bar{\mathbf{q}}_{s_i}(t_m) - \bar{\mathbf{q}}_{s_j}(t_n)\right\| \\
&= \begin{cases}
\dfrac{1}{\theta_s \cdot d_s}\sum_{l=1}^{\theta_s}\sum_{v=1}^{d_s}\dfrac{real(\bar{q}^v_{s_j,l}(t_n)) - imag(\bar{q}^v_{s_j,l}(t_n))}{real(\bar{q}^v_{s_i,l}(t_m)) - imag(\bar{q}^v_{s_i,l}(t_m))}, \\
\forall real(\bar{q}^v_{s_i,l}(t_m)) \ge real(\bar{q}^v_{s_j,l}(t_n)) \cap imag(\bar{q}^v_{s_i,l}(t_m)) \le imag(\bar{q}^v_{s_j,l}(t_n)); \\
\dfrac{1}{\theta_s \cdot d_s}\sum_{l=1}^{W_s/g_s}\sum_{v=1}^{d_s}\dfrac{real(\bar{q}^v_{s_i,l}(t_m)) - imag(\bar{q}^v_{s_i,l}(t_m))}{real(\bar{q}^v_{s_j,l}(t_n)) - imag(\bar{q}^v_{s_j,l}(t_n))}, \\
\forall real(\bar{q}^v_{s_j,l}(t_n)) \ge real(\bar{q}^v_{s_i,l}(t_m)) \cap imag(\bar{q}^v_{s_j,l}(t_n)) \le imag(\bar{q}^v_{s_i,l}(t_m)); \\
\dfrac{1}{\theta_s \cdot d_s}\sum_{l=1}^{W_s/g_s}\sum_{v=1}^{d_s}\dfrac{real(\bar{q}^v_{s_j,l}(t_n)) - imag(\bar{q}^v_{s_i,l}(t_m))}{real(\bar{q}^v_{s_i,l}(t_m)) - imag(\bar{q}^v_{s_j,l}(t_n))}, \\
\forall real(\bar{q}^v_{s_i,l}(t_m)) \ge real(\bar{q}^v_{s_j,l}(t_n)) \cap imag(\bar{q}^v_{s_i,l}(t_n)) \le imag(\bar{q}^v_{s_j,l}(t_m)) \le real(\bar{q}^v_{s_j,l}(t_n)); \\
\dfrac{1}{\theta_s \cdot d_s}\sum_{l=1}^{W_s/g_s}\sum_{v=1}^{d_s}\dfrac{real(\bar{q}^v_{s_i,l}(t_m)) - imag(\bar{q}^v_{s_j,l}(t_n))}{real(\bar{q}^v_{s_j,l}(t_n)) - imag(\bar{q}^v_{s_i,l}(t_m))}, \\
\forall real(\bar{q}^v_{s_j,l}(t_n)) \ge real(\bar{q}^v_{s_i,l}(t_m)) \cap imag(\bar{q}^v_{s_j,l}(t_m)) \le imag(\bar{q}^v_{s_j,l}(t_n)) \le real(\bar{q}^v_{s_i,l}(t_m)); \\
0, \forall others.
\end{cases}
\end{aligned}
\tag{7}
$$

$$U_c(\bar{q}^d_{s_i,v}(t_m),\bar{q}^d_{s_j,v}(t_n))=\begin{cases} real(\bar{q}^d_{s_j,v}(t_n))+imag(\bar{q}^d_{s_j,v}(t_n))\cdot\mathbf{i}, \\ \forall real(\bar{q}^d_{s_i,v}(t_m))\geq real(\bar{q}^d_{s_j,v}(t_n))\cap imag(\bar{q}^d_{s_i,v}(t_m))\leq imag(\bar{q}^d_{s_j,v}(t_n)); \\ real(\bar{q}^d_{s_i,v}(t_m))+imag(\bar{q}^d_{s_i,v}(t_m))\cdot\mathbf{i}, \\ \forall real(\bar{q}^d_{s_j,v}(t_n))\geq real(\bar{q}^d_{s_i,v}(t_m))\cap imag(\bar{q}^d_{s_j,v}(t_n))\leq imag(\bar{q}^d_{s_i,v}(t_m)); \\ real(\bar{q}^d_{s_j,v}(t_n))+imag(\bar{q}^d_{s_i,v}(t_m))\cdot\mathbf{i}, \\ \forall real(\bar{q}^d_{s_i,v}(t_m))\geq real(\bar{q}^d_{s_j,v}(t_n))\cap imag(\bar{q}^d_{s_j,v}(t_n))\leq imag(\bar{q}^d_{s_i,v}(t_m))\leq real(\bar{q}^d_{s_j,v}(t_n)); \\ real(\bar{q}^d_{s_i,v}(t_m))+imag(\bar{q}^d_{s_j,v}(t_n))\cdot\mathbf{i}, \\ \forall real(\bar{q}^d_{s_j,v}(t_n))\geq real(\bar{q}^d_{s_i,v}(t_m))\cap imag(\bar{q}^d_{s_i,v}(t_m))\leq imag(\bar{q}^d_{s_j,v}(t_n))\leq real(\bar{q}^d_{s_i,v}(t_m)); \\ 0,\forall others. \end{cases} \tag{8}$$

Cluster-head node c_i uses modified *K*-means algorithm to cluster the boundary vector sequences received by sliding window at time *t* according to the order of reported events. The value of *K* in the clustering algorithm equals to the types of actor nodes. The algorithm utilizes Eq. (7) to calculate the similarity between new coming sequence and the center point $\bar{\mathbf{M}}_{c_i,j,j\in\{1,2,...,d_a\}}$ of existing classification and uses Eq. (8) to update the classification center points. Cluster-head node c_i can get d_a classifications $C_{j,j\in\{1,2,...,d_a\}}$ after clustering and each classification corresponds to a set of boundary vector sequences received from different sensor nodes. Each classification has its own center point $\bar{\mathbf{M}}_{c_i,j,j\in\{1,2,...,d_a\}}(t)=\{\bar{M}_{c_i,j,1}(t),\bar{M}_{c_i,j,2}(t),...,\bar{M}_{c_i,j,\theta_s}(t)\}$ and $\bar{M}_{c_i,j,l}(t)$ is the l_{th} vector of $\bar{\mathbf{M}}_{c_i,j}(t)$.

Based on center points of classification, cluster-head node can do event type matching. There are d_a kinds of actor nodes in *R*, and each kind of actor nodes can only process corresponding kind of events. After clustering, cluster-head node c_i will match each center point $\bar{\mathbf{M}}_{c_i,j,j\in\{1,2,...,d_a\}}(t)$ with d_a kinds of probability distribution functions of event attributes $F_{v,v\in\{1,2,...,d_a\}}(x)$, to make sure of the event type which best matches with the classification center points . The matching method is shown as Eq. (9).

$$D_c(\bar{\mathbf{M}}_{c_i,j}(t),F_v)=\frac{1}{\theta_s\cdot d_s}\sum_{l=1}^{\theta_s}\sum_{d=1}^{d_s}F_v^d(real(\bar{M}_{c_i,j,l}(t)))-F_v^d(imag(\bar{M}_{c_i,j,l}(t))) \tag{9}$$

In Eq. (9), $F_v^d(x)$ stands for the probability distribution function of the d_{th} dimension attribute.

If $D_c(\bar{\mathbf{M}}_{c_i,j}(t),F_n)=\max_{v=\{1,2,...,d_a\}}\{D_c(\bar{\mathbf{M}}_{c_i,j}(t),F_v)\}$, the event type of classification of which the center point is $\bar{\mathbf{M}}_{c_i,j}(t)$ is *n*, and sensor nodes which send this type of events can be confirmed. Meanwhile, there exists a situation that the probability distribution function of one event type may matches several classification center points $\bar{\mathbf{M}}_{c_i,l,l\neq j}(t)$, which can be merged into $\bar{\mathbf{M}}'_{c_i,n}(t)$ by Eq. (8). After combination, node c_i will report new classification center point $\bar{\mathbf{M}}'_{c_i,n}(t)$, which matches the event type *n*, to the actor node a_n to process the event. Table II shows the steps of cooperative event detection algorithm between sensor nodes and cluster-head nodes (CEDBSC).

TABLE II: PROCEDURES OF CEDBSC

Let W_c be the sliding window of cluster-head c_i ;
Let $\mathbf{S}=\{s_1,s_2,...,s_n\}$ be the set of sensor nodes reporting to the sliding window at time t ;
Let $\bar{\mathbf{Q}}=\{\bar{\mathbf{q}}_{s_1},\bar{\mathbf{q}}_{s_2},...,\bar{\mathbf{q}}_{s_n}\}$ be the set of boundary vector sequences reported to the sliding window at time t ;
Let parameter *K* of *K*-means algorithm equals to d_a ;
Let $F_{v,v\in\{1,2,...,d_a\}}(x)$ be the probability distribution functions of predefined events;
1: classify $\bar{\mathbf{Q}}=\{\bar{\mathbf{q}}_{s_1},\bar{\mathbf{q}}_{s_2},...,\bar{\mathbf{q}}_{s_n}\}$ into d_a sets using *K*-means algorithm at time t and calculate $\bar{\mathbf{M}}_{c_i,j,j\in\{1,2,...,d_a\}}(t)$ to be the center point of each set;
2: calculate $D_c(\bar{\mathbf{M}}_{c_i,j}(t),F_{v,v\in\{1,2,...,d_a\}}(x))$;
3: if $(D_c(\bar{\mathbf{M}}_{c_i,j}(t),F_n)=\max_{v=\{1,2,...,d_a\}}\{D_c(\bar{\mathbf{M}}_{c_i,j}(t),F_v)\})$ then
4: if $(D_c(\bar{\mathbf{M}}_{c_i,l,l\neq j}(t),F_n)=\max_{v=\{1,2,...,d_a\}}\{D_c(\bar{\mathbf{M}}_{c_i,l,l\neq j}(t),F_v)\})$ then
5: merge $\bar{\mathbf{M}}_{c_i,l,l\neq j}(t)$ with $\bar{\mathbf{M}}_{c_i,j}(t)$ and calculate new center point $\bar{\mathbf{M}}'_{c_i,n}(t)$;
6: end if
7: confirm the event type as n ;
8: notify corresponding actor node a_n to deal with the event of type n ;
9: report $\bar{\mathbf{M}}'_{c_i,n}(t)$ to node a_n ;
10: end if

VI. COOPERATIVE EVENT DETECTION ALGORITHM BETWEEN CLUSTER-HEADS AND ACTORS

After finishing type matching with event data received from sensor nodes, cluster-head node c_i will ask corresponding actor nodes to handle detected events. At the same time, node c_i will transmits classification center point $\bar{\mathbf{M}}'_{c_i,j}(t)$ related to event type j to actor node a_j which takes charge of dealing with the events of type j. Node a_j uses the sliding window, of which window size is W_a and step size is W_c, to get boundary vector

sequence $\bar{\mathbf{M}}'_{c_i,j}(t)$ and transform it into slope vector sequence $\bar{\mathbf{r}}_{c_i}(t)$. At time t_{W_a}, actor node a_j receives $n_{a_j}(t_{W_a})$ scheduling requests from cluster-head nodes using sliding window $\left[t_1, t_{W_a}\right]$ and makes sure of the priority of events processing using the modified KNN. Node a_j can use Eq. (10) to calculate the similarity between $\bar{\mathbf{r}}_{c_i}(t_m)$ and other received slope vector sequence $\bar{\mathbf{r}}_{c_{j,j\neq i}}(t_n)$.

$$\begin{aligned} D_c(\bar{\mathbf{r}}_{c_i}(t_m), \bar{\mathbf{r}}_{c_{j,j\neq i}}(t_n)) &= \left\| \bar{\mathbf{r}}_{c_i}(t_m) - \bar{\mathbf{r}}_{c_{j,j\neq i}}(t_n) \right\| \\ &= \frac{1}{(\theta_s - 1)\cdot d_s} \sum_{l=1}^{\theta_s - 1} \sum_{v=1}^{d_s} \left| \bar{r}^{v}_{c_i,l}(t_m) - \bar{r}^{v}_{c_j,l}(t_n) \right| \end{aligned} \tag{10}$$

Node a_j also can choose the k most similar slope vector sequences with $\bar{\mathbf{r}}_{c_i}(t_m)$ to get the average similarity $\bar{\mathbf{r}}^{avg}_{c_i}(t_{W_a})$ of $\bar{\mathbf{r}}_{c_i}(t_m)$. The calculation method of average similarity is shown as below.

$$\bar{\mathbf{r}}^{avg}_{c_i}(t_{W_a}) = \begin{cases} \dfrac{1}{k} \displaystyle\sum_{j=1, j\neq i}^{k} D_c(\bar{\mathbf{r}}_{c_i}(t_m), \bar{\mathbf{r}}_{c_{j,j\neq i}}(t_n)), \\ \qquad\qquad \forall k \le n_{a_j}(t_{W_a}) - 1 \\ \dfrac{1}{n_{a_j}(t_{W_a}) - 1} \displaystyle\sum_{j=1, j\neq i}^{n_{a_j}(t_{W_a})-1} D_c(\bar{\mathbf{r}}_{c_i}(t_m), \bar{\mathbf{r}}_{c_{j,j\neq i}}(t_n)), \\ \qquad\qquad \forall k > n_{a_j}(t_{W_a}) - 1 \end{cases} \tag{11}$$

TABLE III: PROCEDURES OF CEDBCA.

Let W_a be the sliding window of actor a_j which deal with the event of type j;
Let $\bar{\mathbf{M}}'_j = \{\bar{\mathbf{M}}'_{c_1,j}, ..., \bar{\mathbf{M}}'_{c_i,j}, ...\}$ be the set of boundary vector sequences reported to the sliding window at time t;
1: calculate $\bar{\mathbf{r}} = \{\bar{\mathbf{r}}_{c_1}, ..., \bar{\mathbf{r}}_{c_j}, ...\}$ to be the slope vector sequences from $\bar{\mathbf{M}}'_j$ at time t;
2: calculate $D_c(\bar{\mathbf{r}}_{c_i}, \bar{\mathbf{r}}_{c_{v,v\neq i}})$ and get the average similarity $\bar{\mathbf{r}}^{avg}_{c_i}(t)$ using k-nearest neighbor sequences;
3: if $\bar{\mathbf{r}}^{avg}_{c_i}(t) = \max\limits_{l=\{1,2,...,n_{a_j}(t)\}} \left\{\bar{\mathbf{r}}^{avg}_{c_l}(t)\right\}$ then
4: deal with the event reported by cluster-head c_i with priority;
5: end if

If $\bar{\mathbf{r}}^{avg}_{c_i}(t_{W_a}) = \max\limits_{v=\{1,2,...,n_{a_j}(t_{W_a})\}} \left\{\bar{\mathbf{r}}^{avg}_{c_v}(t_{W_a})\right\}$, the event reported by cluster node c_i will be prioritized. At the same time, node a_j will realize the sensor nodes related to cluster-head node c_i according to $\bar{\mathbf{r}}^{avg}_{c_i}(t_{W_a})$. Using localization algorithms [12], node a_j can move to the position where events occur accurately to implement control strategies. Table III shows the steps of cooperative event detection algorithm between cluster-head nodes and actor nodes.

VII. PERFORMANCE ANALYSIS

The cooperative event detection method for WSAN in this paper is a three-tiered framework, which consists of CEDBAS, CEDBSC and CEDBCA. By the coordination between sensor nodes, cluster-head nodes and actor nodes, the types of happening events in the surrounding environment can be detected and related strategies are implemented to control the development of happening events in WSAN applications. The proposed framework should meet the requirements of real-time, low energy consumption and mobility in practice. The validity of event detection framework will be analyzed via simulation in Section 7.

A. Computational Complexity

In CEDBAS, sensor node s_i can get $\bar{\mathbf{p}}_{s_i}(t_j)$ by Eq. (1) at time t_j, and its computational complexity is $O(d_s \times W_s)$. Its computational complexity of calculating the similarity between two sequences by Eq. (6) is $O(2 \times d_s \times \theta_s)$. If node s_i wants to get $\bar{\mathbf{q}}_{s_i}(t_v)$ by Eq. (2), it needs to send data to cluster-head node at time t_v, and its computational complexity is $O(d_s \times W_s)$. So the maximum computational complexity of sensor node s_i is $O(2 \times d_s \times W_s)$. From the above, it can be seen that the computational complexity of sensor node will be high if W_s is too big and the data information contained in $\bar{\mathbf{p}}_{s_i}(t_j)$ will be too little to match correct pattern if W_s is too small.

In CEDBSC, cluster-head node c_i can uses Eq. (7) to do clustering at time t in order to get d_a classifications and its computational complexity is $O(3 \times \theta_s \times d_s \times d_a \times (n_s - n_c - d_a))$. When using K-means for clustering, Eq. (8) is used to find the center point of each classification and its computational complexity is $O(3 \times \theta_s \times d_s \times (n_s - n_c - d_a))$. In CEDBSC, Eq. (9) is used to do event type matching after clustering, and its computational complexity is $O((2m+1) \times \theta_s \times d_s \times {d_a}^2)$, in which m is the computational complexity of probability distribution function $F_{v,v\in\{1,2,...,d_a\}}(x)$ of event attribute. After event type matching, Eq. (8) is used to combine classification center points of the same event type, and its computational complexity is $O(3 \times \theta_s \times d_s \times (d_a - 1))$. So the maximum computational complexity of cluster-head node c_i at time t is $O(\theta_s \times d_s \times d_a \times (3 \times (n_s - n_c) + 2 \times m \times d_a))$. Considering the step size of sliding window on cluster-head node is W_s, the average maximum computational complexity of cluster-head node is $O((d_s \times d_a \times (3 \times (n_s - n_c) + 2 \times m \times d_a))/g_s)$ in CEDBSC. From the above, it can be seen that the computational complexity of cluster-head node and the sensitivity of divided subsequences will be low if g_s is

too big. However, if g_s is too small, the pattern of subsequences will be disrupt easily, and its computational complexity will be high. Besides, normal sensor nodes can use cluster-head election algorithm to take turns to be cluster-head nodes and prolong their working life.

In CEDBCA, actor node a_j can uses Eq. (5) to transform received boundary vector sequences into slope vector sequences, and its computational complexity is $O(4\times(\theta_s-1)\times d_s)$. The computational complexity of calculating the similarity between slope vector sequences by Eq. (10) is $O((2\times\theta_s\times d_s-1)\times n_c\times(n_c-1))$, and its computational complexity of calculating the average similarity by Eq. (11) is $O(k\times n_c)$. So the maximum computational complexity of actor node a_j at time t is $O((2\times\theta_s\times d_s+k)\times n_c)$. Because the step size of the sliding window belonging to actor node a_j is W_c , the average maximum computational complexity of actor node is $O(((2\times\theta_s\times d_s+k)\times n_c)/W_c)$ in CEDBCA. Considering actor nodes in WSAN application have enough energy, they can work long enough.

B. Real-Time

To make event detection meet the requirement of real-time for WSAN applications, the proposed algorithms can be adjusted further in practice. Considering the demand of real-time, sensor node s_i does not request each neighbor node to return its voting result when node s_i calculates support rate v_{s_i} in CEDBAS. Node s_i just waits time t_v and counts the voting value received from neighbor nodes to realize the real-time event detection of sensor nodes.

The number n_{c_i} of boundary vector sequences received by cluster-head node c_i from sensor nodes change over time. Considering the demand of real-time, the size W_c of sliding window can be adjusted dynamically in CEDBSC. If node c_i receives $n_{c_i}(t_1)$ sequences at time t_1 and $n_{c_i}(t_1)\geq p\cdot d_a$, in which p is a positive integer, then the size of sliding window belonging to node c_i can be adjusted to $(1-1/p)\cdot W_c$ at time t_{W_s} . But if $n_{c_i}(t_1)<d_a$, then the size of sliding window belonging to node c_i can be adjusted to W_c .

Similar to the adjustment method of sliding window in CEDBSC, the size W_a of sliding window belonging to actor node a_j can be adjusted dynamically to meet the demands of real-time in CEDBCA. If node a_j receives $n_{a_j}(t_1)$ sequences at time t_1 and $n_{a_j}(t_1)\geq p\cdot k$, then the size of sliding window belonging to node a_j can be adjusted to $(1-1/p)\cdot W_a$ at time t_{W_c} . But if $n_{a_j}(t_1)<k$, then the size of sliding window belonging to node a_i can be adjusted to W_a .

C. Mobility

Compared with the existing WSN applications, there are actor nodes in WSAN applications, which have mobility. Cluster-head node c_i needs to send processing request to related actor node a_i after detecting the event type. With the development of events, node c_i may send the same request repeatedly to the same actor node. At the same time, actor node a_i may receive event-related boundary vector sequences from several cluster-head nodes when it move to process the event reported by node c_i . This brings new challenges to the requirements of real-time and accuracy for event detection algorithms. In CEDBCA, actor node a_i only transforms the latest boundary vector sequence into slope vector sequence at time t_{W_c} , if a_i receives more than one processing request from the same cluster-head node in sliding window $\left[t_1,t_{W_a}\right]$. Because the step size of sliding window on node a_i is W_c , node a_i will still implement current control strategies during time interval $\left[t_{W_a+1},t_{W_a+W_c}\right]$, even if it receives requests from several cluster-head nodes. However, actor node a_i begins to receive new processing requests and get the priority of event handling at time $t_{W_a+W_c+1}$. If new event processing request is prioritized, actor node a_i will stop processing the current event and turn to handle the new processing request.

VIII. Simulation Evaluation

In this paper, Matlab is used to build the simulation environment to evaluate the performance of 3-CEDF. In the simulation, event area R is $400\times400\text{m}^2$, in which there are 441 normal sensor nodes, 16 cluster-head nodes and 4 actor nodes. All the normal sensor nodes and cluster-head nodes are deployed evenly in the event area. Sensor nodes will find their corresponding cluster-head nodes according to their geographic location when the clustering phase of sensor nodes has been completed. There are 3 kinds of sensors on normal sensor nodes and the sampling frequency is 1Hz. The window size W_s and the partitioning granularity g_s of normal sensor nodes are 16 and 4 respectively. In addition, the window size W_c of cluster-head nodes is 24. When normal sensor node finds some event happening, cluster-head node will detect the event type and select corresponding actor node to process the event. The window size W_a and movement speed of actor nodes are 36 and 2m/s. Each actor node can only deal with one type of event.

In the simulation, there are 4 kinds of events in R. The probability distribution functions of 3 dimension attributes of each event obey normal distribution $N(\mu,\sigma^2)$, in which μ is the expect value and σ^2 is the standard deviation. In all, there are 12 kinds of

independent distribution functions $N(\mu_i^d,(\sigma_i^d)^2)$, in which event type i=1,2,3,4 and attribute dimension d=1,2,3. μ_i^d and σ_i^d are randomly assigned in interval [1,10] separately.

Different types of events will occur randomly in R at every moment during the simulation. The perceived radius R of sensor nodes is 10m and the sensor reading $\bar{m}_{s_i}(t)$ changes with the distance λ between sensor node s_i and the event location as $\bar{m}_{s_i}(t)=e^{-\lambda/2R}\cdot\bar{m}(t)$, in which $\bar{m}(t)$ is the actual attribute value of the event. If actor nodes move to the location of the happening events, they will process the events. However, the happening events will exist till the end of the simulation if no actor nodes come. Considering the existence of error and disturbance when sensor nodes work, the errors of sensor readings obey normal distribution $N(0,1)$ independently.

The current researches on event detection algorithms mostly focus on WSN applications. The cooperative detection algorithms applied in WSAN are not presented yet. The simulation experiments aim at analyzing the performance of effectiveness, real-time, mobility and communication traffic for 3-CEDF, as well as the influence of different key parameters to its performance. To compare with the proposed detection methods in this paper, the simulation constructs a centralized event detection algorithm, in which cluster-head nodes send the data perceived by sensor nodes to sink node and corresponding actor nodes are called by sink node to handle the events.

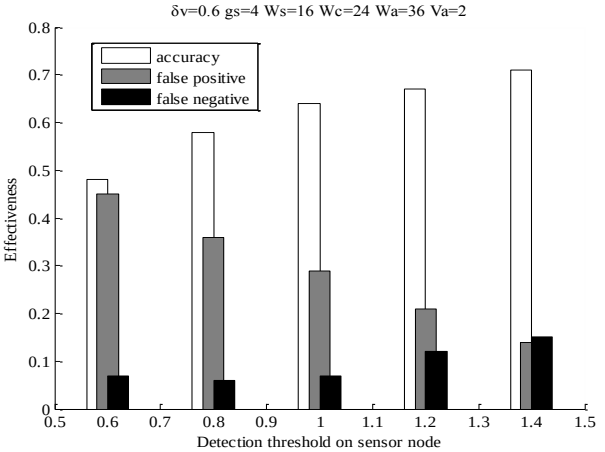


Fig. 2. Effectiveness of 3-CEDF affected by the detection threshold on sensor node.

A. Effectiveness

The simulation examines the effectiveness of the proposed event detection method in this paper, including accurate rate and false rate. The accurate rate can be calculated as the times of detecting the right event type by cluster-head nodes divided by the total number of the events reported by cluster-head nodes. The false rate includes false positive rate and false negative rate. The false positive rate can be calculated as the times of detecting the types of some events wrong and of detecting the types of non-existing events divided by the total number of the events reported by cluster-head nodes. The false negative rate refers to the undetected events, which actually happen. The simulation also analyzes the impact of parameters δ_s and δ_v on the effectiveness of the algorithms. The simulation results are displayed in Fig. 2 and Fig. 3.

As can be seen from Fig. 2, the accuracy of the algorithm increases from 48% to 71% when parameter δ_s increases from 0.6 to 1.4. This is because the number of events perceived by normal sensor nodes decreases gradually as δ_s increases and the total number of events reported by cluster-head nodes decreases correspondingly.

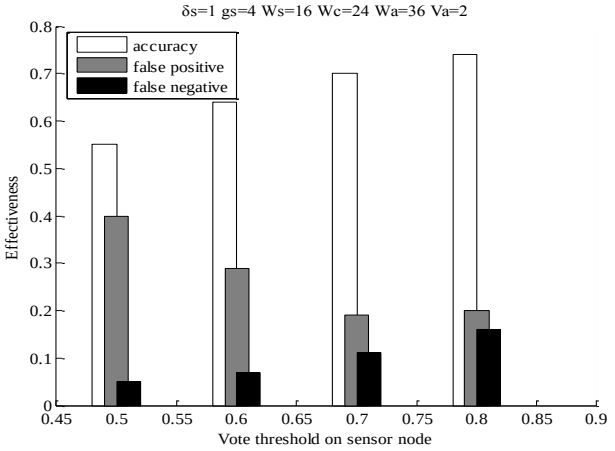


Fig. 3. Effectiveness of 3-CEDF affected by the vote threshold on sensor node

At the same time, the influence of the noise on sensor readings weakens gradually, and the times of detecting the event type correctly by cluster-head nodes increase correspondingly. So the accuracy of the algorithm will be improved with the increase of δ_s . In Fig. 2, the false positive rate decreases from 45% to 14%. This is because as δ_s increases, the times of detecting event types wrong by cluster-head nodes decrease gradually and the influence of the noise on sensor readings lessens correspondingly. Moreover, the times of detecting the types of non-existing events decrease. So the false positive rate will be reduced gradually with the increase of δ_s . In the simulation, the false negative rate of the algorithm increase by 8% with δ_s . As can be seen from Fig. 3, the accuracy of the algorithm increases from 55% to 74% when parameter δ_v increases from 0.5 to 0.8. This is because with the increase of δ_v , normal sensor nodes which report event data need to be closer to the event sources and the influence of the distance on sensor readings attenuation weakens correspondingly. At the same time, the times of detecting event types correctly by cluster-head nodes increase and the number of the events reported by normal sensor nodes reduces. Therefore, the accuracy of the algorithm will be improved as δ_v increases. In Fig. 3, the false positive rate decreases from

40% to 20%. This is because as δ_v increases, the times of detecting event types wrong by cluster-head nodes decrease gradually and the influence of the distance on sensor readings attenuation weakens correspondingly. At the same time, the times of detecting the types of non-existing events decrease. So the false positive rate of the algorithm will reduce gradually with the increase of δ_v. In the simulation, the false negative rate increases to 11% with δ_v.

B. Real-Time

The simulation evaluates the influence of sliding window size on the real-time of the cooperative event detection method proposed in this paper. The following figures show the average delay from events occurrence to receiving processing requests by actor nodes with different values of W_c and W_a.

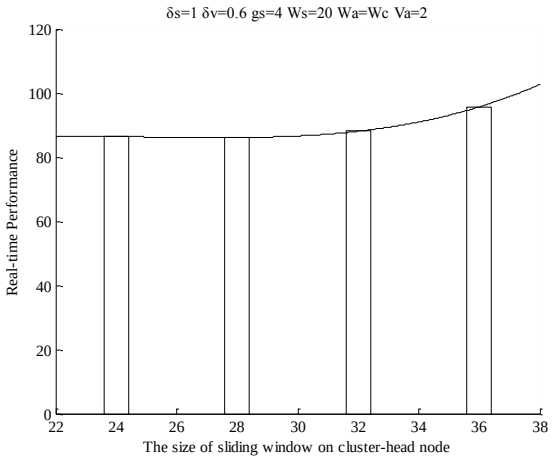


Fig. 4. Real-time performance of 3-CEDF affected by the size of sliding window on cluster-head node.

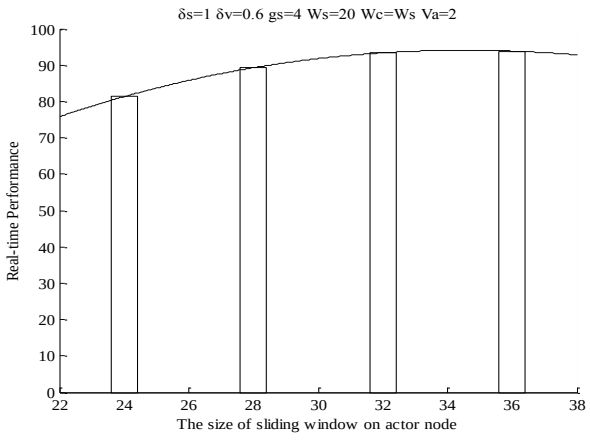


Fig. 5. Real-time performance of 3-CEDF affected by the size of sliding window on actor node.

As can be seen from Fig. 4, the average delay increases from 86s to 95.69s and the amplitude also increases gradually with the increase of W_c and W_a. However, considering the change of sliding windows, the increase amplitude of the average delay is smaller than the one of sliding window. This is because cluster-head nodes can collect more event-related sequences and the type matching of center points is more accurate when the sliding window size of cluster-head nodes increases gradually. Therefore, the event types can be detected and reported to actor nodes more quickly. In Fig. 5, the average delay increases from 81.32s to 93.84s and the amplitude changes uniformly when W_c doesn't change and W_a increases gradually. However, considering the change of W_a, the increase amplitude of average delay is similar to the one of sliding window. This is because actor nodes only judge whether the reported events are prior to process when receiving new event data sent by cluster-head nodes, and the change of average delay is slightly affected.

C. Mobility

The movement speed of actor nodes and parameters W_c, W_a can affect the priority judgment of event handing. The simulation estimates the influence of movement speed and W_c, W_a on the controllability of event area with the cooperation between all kinds of nodes in WSAN. In the simulation, the duration of all events happened in R is set as T_a, when actor nodes process events. If there is no actor node coming to handle events, the lasting time of all events in R is set as T. Moreover, the ratio between T_a and T can be used to evaluate the controllability for WSAN with the proposed event detection algorithm.

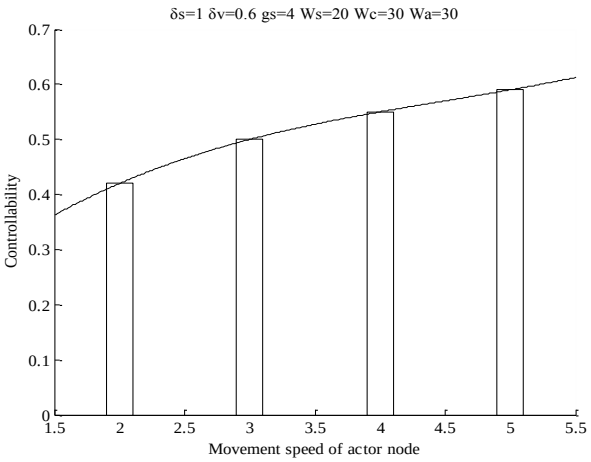


Fig. 6. Controllability of 3-CEDF affected by the movement speed of actor node when the size of sliding window on cluster-head node equal to the one on actor node.

As can be seen from Fig. 6 and Fig. 7, the controllability of event area is stronger with the increase of movement speed of actor nodes. This is because actor nodes may receive new processing requests in their way to handle previous events. If the actor node does not arrive at the position of previous event and the processing priority of new event is higher, the actor node will move to the location of new event. Faster the movement speed is, smaller the probability of the above problem happening is, and stronger the controllability of event area is.

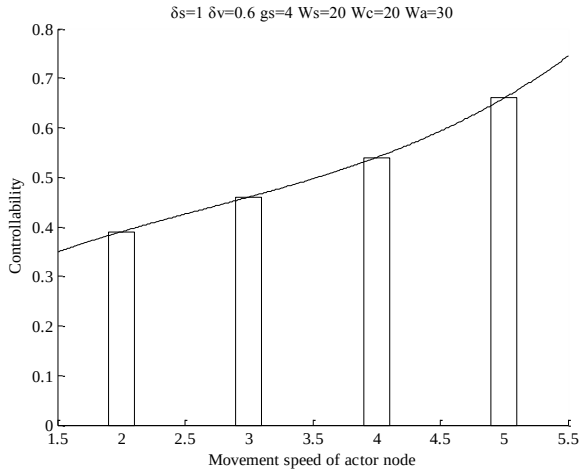


Fig. 7. Controllability of 3-CEDF affected by the movement speed of actor node when the size of sliding window on sensor node equal to the one on cluster-head node.

In Fig. 7, the frequency of making decision by actor nodes is high when W_c is small, and the controllability for WSAN increases greatly with the increase of the movement speed of actor nodes. However, when the speed of actor nodes is slow, the probability of the above problem happening increases, and the controllability for WSAN declines quickly. In the simulation, there are only 4 actor nodes which process different types of events separately. If there are more actor nodes in the event area, the controllability for WSAN will increase greatly.

D. Communication Traffic

WSAN nodes need communicate to cooperate with each other. Frequent communication will lead to faster energy consumption of nodes. The simulation examines the influence of parameters g_s and W_s on the communication traffic of the cooperative event detection method proposed in this paper. The average number of exchanged messages per unit of simulation time is used to estimate the communication traffic in WSAN. Besides, the traffic of the centralized event detection algorithm is also tested in the simulation.

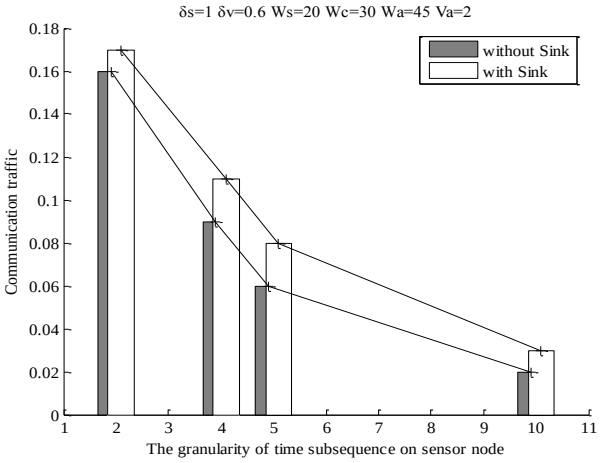


Fig. 8. Communication traffic of 3-CEDF affected by the granularity of time subsequence on sensor node.

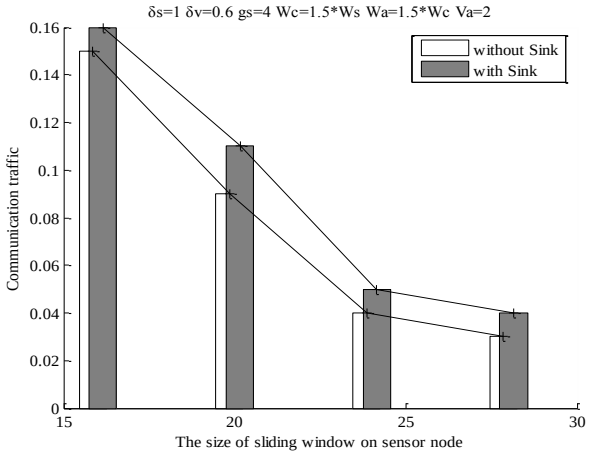


Fig. 9. Communication traffic of 3-CEDF affected by the size of sliding window on sensor node.

As can be seen from Fig. 8, the communication traffic of the event detection algorithm without sink node decreases from $0.16s^{-1}$ to $0.02\ s^{-1}$ gradually with the increase of g_s. This is because the information contained by the sequences decrease gradually as g_s increases and the probability of detecting event types correctly also decreases. Therefore, the traffic of reporting event data and mobilizing actor nodes will decreases. However, the above problem will interfere the effectiveness of the event detection algorithm. In Fig. 8, the traffic of the centralized event detection algorithm is slightly higher than the algorithm proposed in this paper. This is because the reporting of event data and the scheduling of actor nodes both need to communicate with sink node. As can be seen from Fig. 9, the traffic of the event detection algorithm without sink node decreases from $0.15s^{-1}$ to $0.03s^{-1}$ gradually with the increase of W_s. This is because the information contained by event data reported from sensor nodes increases and the number of event reports decreases as W_s increases. In Fig. 9, the traffic of the centralized event detection algorithm is still slightly higher than the algorithm proposed in this paper.

IX. CONCLUSIONS

In this paper, we propose a three-layer cooperative event detection framework for WSAN. Normal sensor nodes use PAA to do isometric segmentation on the received multidimensional vector time sequences. If the proposed detection algorithm uses non-isometric segmentation [30], the loss of key characteristic information can be avoided, and the time series are contractile in the time axis. However, it has higher complexity and spends too much time, and there are not good algorithms for similarity measure between the sequences with unequal length.

The similarity of two sequences can be indicated by the distance. The smaller the distance is, the higher the similarity is, and vice versa. This paper used the simplest and the most widely used similarity measure method

based on the Euclidean distance [31]. Statistical feature extraction methods [32] or model-based methods [33] can also be used to improve the similarity measurement. However, the former has very strong application dependency and needs manual intervention, the latter needs longer time series to evaluate better model parameters, and it can hardly deal with short time series correctly[35].

The proposed 3-CEDF in this paper used modified *K*-means algorithm to process clustering analysis. Other clustering methods, such as hierarchical method [36], density-based method [37], partitional method [38] and grid-based method [39], also can be used in the event detection framework. However, the above methods need to be selected according to the application data models. Besides, how to deal with large-scale high-dimensional data set is one of the hot and difficult issues in data mining.

In the future application of WSAN, the data received by sensor nodes include audio, video and other media data rather than regular scalar data. Actor nodes will analyze multimedia data and implement complex control strategies. Therefore, multimedia data mining [40] is one of the further research directions. Besides, various needs of WSAN applications also put forward higher requests to the performance of event detection algorithms. The algorithm design should take into account the task allocation mechanisms of actor nodes in WSAN. Therefore, the detection algorithms need to provide further classification of event characteristics. Due to network delays, faulty readings and vandalism, the design of robust event detection methods should be one of the future research directions.

ACKNOWLEDGEMENT

This work is supported by Fundamental Research Funds for the Central Universities (No.KYZ201421); The Agricultural Project for New Variety, New Technology and New Model of Jiangsu Province (SXGC[2014]309); Government sponsored project under the science and technology development initiative : Research and demonstration of electronic food safety traceability (2015BAK36B05).

REFERENCES

[1] E. Canete, J. Chen, M. Diaz, L. Llopis, and B. Rubio, "A service-oriented approach to facilitate WSAN application development," *Ad Hoc Networks*, vol. 9, p. 3, 2011.

[2] W. J. Guo and W. Zhang, "A survey on intelligent routing protocols in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 38, pp. 185-201, Feb. 2014.

[3] I. F. Akyildiz, W. Su, Y. Sankarsubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, pp. 393-422, 2004.

[4] R. D. Li, J. Li, and H. Asaeda, "A hybrid trust framework for wireless sensor and actuator networks in cyber-physical systems," *IEICE Transactions on Information and Systems*, vol. 10, pp. 2586-2596, 2010.

[5] Y. Wang, S. X. Ding, and D. M. Xu, "An H-Infinity fault estimation scheme of wireless networked control systems for industrial real-time applications," *IEEE Transactions on Control Systems Technology*, vol. 22, pp. 2073-2086, Nov. 2012.

[6] S. Cherrier, Y. M. Ghamri-Doudane, S. Lohier, and G. Roussel, "Services collaboration in wireless sensor and actuator networks: Orchestration versus choreography," in *Proc. ISCC*, 2012.

[7] H. Nakayama, Z. M. Fadlullah, N. Ansari, and N. Kato, "A novel scheme for WSAN sink mobility based clustering and set packing techniques," *IEEE Transactions on Automatic Control*, vol. 56, p. 10, 2011.

[8] M. Okhowat, M. Sharifi, and H. Momeni, "Task allocation to actors in wireless sensor actor networks: An energy and time aware technique," in *Proc. WCIT*, 2010.

[9] S. Bang and J. Myoungshic, "Weighted support vector machine using-means clustering," *Comuunications in Statistics-Simulation and Computation*, vol. 43, pp. 2307-2324, Nov. 26, 2012.

[10] S. Z. Erdogan, T. T. Bilgin, and J. Cho, "Fall detection by using k-nearest neighbor algorithm on WSN data," *IEEE GLOBECOM*, 2010.

[11] M. Ohsaki, H. Abe, and T. Yamaguchi, "Numerical time-series pattern extraction based on irregular piecewise aggregate approximation and gradient specification," *New Generation Computing*, vol. 25, 2007.

[12] C. Llas, D. Rosner, R. Tataroiu, and A. Surpateanu, "Wireless sensors and actuators networks: Localization for medical and robotics applications," *Control Engineering and Applied Informatics*, vol. 14, pp. 34-39, 2012.

[13] F. Losilla, A. J. Garcia-Sanchez, F. Garcia-Sanchez, J. Garcia-Haro, and Z. J. Haas, "A comprehensive approach to WSN-based ITS applications: A survey," *Sensors*, vol. 11, 2011.

[14] E. Ruiz-lbarra and L. Villasenor-Gonzalez, "Cooperation mechanism taxonomy for wireless sensor and actor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 7, 2009.

[15] H. Momeni, M. Sharifia, and M. Okhovvat, "Distributed assignment of real-time tasks in wireless sensor actor networks," *IEICE Electronics Express*, vol. 8, p. 7, 2011.

[16] Z. Zinon, C. Chrysostomos, and V. Vasos, "Wireless sensor networks mobility management using fuzzy logic," *Ad Hoc Networks*, vol. 16, pp. 70-87 , May 2014.

[17] X. Liu, K. M. Hou, and D. V. Christophemiros, "A hybrid real-time energy-efficient operating system for the resource-constrained wireless sensor nodes," *Sensors*, vol. 14, pp. 17621-17654, Sep. 2014.

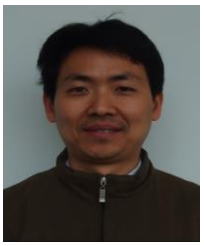
[18] M. Kushal, G. Shalabh, and R. Asok, "Statistical-mechanics-inspired optimization of sensor field configuration for detection of mobile targets," *IEEE Transactions on Systems Man and Cybernetics Part b-Cybernetics*, vol. 41, pp. 783-791, Jun. 2011.

[19] M. Moshtaghi, T. C. Havens, J. C. Bezdek, L. Park, C. Leckie, S. Rajasegarar, J. M. Keller, and M. Palaniswami,

"Clustering ellipses for anomaly detection," *Pattern Recognition*, vol. 44, pp. 44-69, 2011.
[20] A. Abdegawad, M. Bayoumi, "Data fusion in WSN," in *Resource-Aware Data Fusion Algorithms for Wireless Sensor Networks*, 2012, vol. 118, pp. 17-35.
[21] K. S. Sedighian and S. Mohsen, "Connectivity weakness impacts on coordination in wireless sensor and actor networks," *IEEE Communications Surveys and Tutorials*, vol. 15, pp. 145-166, 2013.
[22] M. Xie, J. K. Hu, and S. Han, "scalable hypergrid k-nn-based online anomaly detection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24 pp. 1661-1670, Aug. 2013.
[23] W. Wang, R. H. Zhang, and Z. W. Chen, "Normal vector based fault-tolerant event boundary detection in wireless sensor network," in *Proc. IEEE International Conference on Trust Security and Privacy in Computing and Communication*, vol. 16, pp. 1339-1344, 2013.
[24] A. Ngaopitakkul and C. Pothisarn, "The protection of transmission network systems using discrete wavelet transforms," *Internation Journal of Innovative Computing Information and Control*, vol. 8, pp. 6491-6502, 2012.
[25] Y. L. Jiang and H. B. Chen, "Application of general orthogonal polynomials to fast simulation of nonlinear descriptor systems through piecewise-linear approximation," *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, vol. 31, pp. 804-808, 2012.
[26] H. L. Li and C. H. Guo, "Symbolic aggregate approximation based on shape features," *Pattern Recognition and Artificial Intelligence*, vol. 24, pp. 665-672, 2011.
[27] M. Cullen, S. Brian, and L. Andrew, "Boundary identification in EBSD data with a generalization of fast multi-scale clustering," *Ultramicroscopy*, vol. 133, pp. 16-25, Oct. 2013.
[28] D. B. Dai, Y. Xiong, and Y. Y. Zhu, "Efficient algorithm for sequence similarity search based on reference indexing," *Journal of Software*, vol. 21, pp. 718-731, 2010.
[29] E. Rasul and S. R. Abdul, "Similarity measure using hausdorff distance in 2D shape recognition system," *Advances in Intelligent Systems Research*, vol. 68, pp. 194-197, 2013.
[30] G. Tomasz and L. Maciej, "Non-isometric transforms in time series classification using DTW," *Knowledge-Based Systems*, vol. 61, pp. 98-108, May 2014.
[31] C. Ivan, A. Ignacio, and K. Natalio "Blind optimisation problem instance classification via enhanced universal similarity metric," *Memetic Computing*, vol. 6, pp. 263-276, Dec. 2014.
[32] H. Yuan and X. P. Zhang, "Statistical modeling in the wavelet domain for compact feature extraction and similarity measure of images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 3, pp. 439-445, 2010.
[33] J. A. Rodriguez-Serrano and F. Perronnin, "A model-based sequence similarity with application to handwritten word spotting," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2108-2120, 2012.
[34] Y. F. Cheng, H. Tang, and X. Q. Chen, "Research and improvement on HMM-Based face recognition," *Mechanical Design and Power Engineering*, vol. 490-491 pp. 1338-1341, 2014.
[35] S. K. Jain, S. N. Singh, and J. G. Singh, "An adaptive time –efficient technique for harmonic estimation of nonstationary signals," *IEEE Transactions on Industrial Electronics*, vol. 60, pp. 3295-3303, Aug. 2013.
[36] M. Naraghi-Pour and G. C. Rojas, "A novel algorithm for distributed localization in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 11, Nov. 2014.
[37] S. Shahaboddin, A. Amineh, and A. N. Badrul, "D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks," *Measurement*, vol. 55, pp. 212-226, Sep. 2014.
[38] A. Helena and F. Ana, "Statistical modeling of dissimilarity increments for d-dimensional data: Application in partitional clustering," *Pattern Recognition*, vol. 45, pp. 3061-3071, Sep. 2012.
[39] T. Kotiswaran and K. Murugan, "Grid-Based clustering with predefined path mobility for mobile sink data collection to extend network lifetime in wireless sensor networks," *IETE Technical Review*, vol. 29, pp. 133-147, Mar.-Apr. 2012.
[40] J. X. Felix and S. Marios, "Subspace-Based discrete transform encoded local binary patterns representations for robust periocular matching on NIST's face recognition grand challenge," *IEEE Transactions on Image Processing*, vol. 23, pp. 3490-3505, Aug. 2014.

Hao-Yun Wang received the Ph.D. degree in information network from Nanjing University of Posts and Telecommunications in 2010. He also received the M.E. degree in communication and information systems from Kunming University of Sciences and Technology in 2005 and the B.E. degree in communication engineering from Huazhong University of Science and Technology in 2002, respectively. Currently, he is working at the College of Information Science and Technology in Nanjing Agricultural University. His current research interests are in the area of WSN, WSAN and CPS.

Yu-Han Fei was born in Jiangsu Province, China, in 1992. He received the B.E. degree from the Nanjing University of Posts and Telecommunications (NUPT) in 2014, Currently, he is a postgraduate student who Study in Nanjing Agricultural University, major in computer science and technology. His research interests include metaheurisitic algorithms, parameter identification and WSAN.



Shou-Gang Ren received the Ph.D. degree in Mechanical and Electronic Engineering from Nanjing University of Aeronautics and Astronautics in 2006. He also received the M.E. degree in Mechanical and Electronic Engineering from Nanjing University of Aeronautics and Astronautics in 2002 and the B.E. degree in Mechanical and manufacturing engineering from Nanjing University of Aeronautics and Astronautics in 1999, respectively. Currently, he is working at the College of Information Science and Technology in Nanjing Agricultural University. His current research interests are in the area of CPS and Agricultural informatization.

Huan-Liang Xu received his Ph.D. degree from Nanjing University of Aeronautics and Astronautics. He is currently a professor and the vice-president of College of Information Science and Technology at Nanjing Agriculture University. He is also the senior member of China Computer Federation, the vice-chairman of CG&CAD Committee of Jiangsu Computer Federation, and the standing member of Computer Application Committee of Jiangsu Computer Federation and the one of Jiangsu IoT Special Committee. His current research interests include IoT of agriculture, agricultural informatization, and farm product traceability system.