Hindawi Publishing Corporation
Journal of Sensors
Volume 2015, Article ID 641235, 11 pages
http://dx.doi.org/10.1155/2015/641235

Research Article

A Cluster-Based Fuzzy Fusion Algorithm for Event Detection in Heterogeneous Wireless Sensor Networks

ZiQi Hao,[1] ZhenJiang Zhang,[1] and Han-Chieh Chao[2]

[1] *Key Laboratory of Communication and Information Systems, School of Electronic and Information Engineering, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China*
[2] *Department of Electronic Engineering, National Ilan University, Yilan 26047, Taiwan*

Correspondence should be addressed to ZhenJiang Zhang; zhjzhang1@bjtu.edu.cn

Received 2 November 2014; Accepted 22 December 2014

Academic Editor: Qing-An Zeng

As limited energy is one of the tough challenges in wireless sensor networks (WSN), energy saving becomes important in increasing the lifecycle of the network. Data fusion enables combining information from several sources thus to provide a unified scenario, which can significantly save sensor energy and enhance sensing data accuracy. In this paper, we propose a cluster-based data fusion algorithm for event detection. We use k-means algorithm to form the nodes into clusters, which can significantly reduce the energy consumption of intracluster communication. Distances between cluster heads and event and energy of clusters are fuzzified, thus to use a fuzzy logic to select the clusters that will participate in data uploading and fusion. Fuzzy logic method is also used by cluster heads for local decision, and then the local decision results are sent to the base station. Decision-level fusion for final decision of event is performed by base station according to the uploaded local decisions and fusion support degree of clusters calculated by fuzzy logic method. The effectiveness of this algorithm is demonstrated by simulation results.

1. Introduction

Recent advances of low-cost wireless sensor networks have brought about many applications such as military, environmental monitoring, and intelligent transportation system. Event detection has been widely studied as a typical application of WSN.

Heterogeneous wireless sensor networks (HWSN) are networks composed of different kinds of sensors, which are different in some aspects such as energy, computing power, and storage space. In HWSN such as a cluster-based network, cluster heads are more powerful than cluster members in all the resources like power, storage, communication, and processing data; this heterogeneity alleviates the overhead of cluster members for the fact that all the expensive computations can be performed by cluster headers [1]. Therefore, the load-balance and lifetime of network can be significantly improved.

Data fusion is a technology that enables combining information from several sources in order to form a unified picture, and it is widely used in various areas such as sensor networks, robotics, and video and image processing [2]. As an efficient method for collaborative decision making of multiple sensors, data fusion has many advantages in WSN. Using multiple sensors to detect the same event can largely eliminate data ambiguity which may be caused by only one sensor, thus to enhance data reliability and ability of fault tolerance effectively. Moreover, as WSN is energy limited network, the sensors are generally battery powered and once deployed it is hard to be recharged; therefore energy saving becomes an important factor for expanding the lifetime of WSN. Concerning the fact that energy consumption caused by communication is considerably larger than that of data processing, when data fusion is performed, sensor data are fused and only the result is forwarded; thus the number of messages is reduced, which can significantly avoid collisions and save communication energy.

Clustering is a frequently used physical architecture of data fusion; it groups sensor nodes into several clusters in order to achieve the network scalability objective [3]. Every cluster has a cluster head (CH) for executing data fusion and serving as a relay; as a consequence CH consumes more energy than ordinary sensor nodes; therefore a more powerful sensor is more likely to be selected as a cluster head.

Fuzzy reasoning is a theoretical reasoning scheme of data fusion; it introduces the novel notion of membership degree which enables dealing with imperfect data appropriately [4]. Generally, the sensing data of a single sensor may be vague and partial; thus it is difficult to obtain the final fusion decision result via precisely quantitative calculation of these imperfect data. Fuzzy logic uses the membership degree to fuzzify the partial data and then combines them with fuzzy rules thus to produce fuzzy output, which is an efficient solution to deal with the uncertainty of data.

Previous researches mainly focus on designing or improving the fusion or clustering algorithms independently; few works have combined the two technologies together to perform monitoring task. The authors of [5] have proposed a cluster based multisensor data fusion algorithm in WSN using fuzzy logic for event detection; the method adopted fuzzy logic approach to handle the uncertainty and vagueness present in the environment data in the local decision period of cluster head. However, the proposed fuzzy logic fusion method only considers data level fusion of intracluster, without designing the specific clustering algorithm and decision-level fusion method.

In this paper, we proposed a cluster-based fusion method for event detection. We consider a heterogeneous wireless sensor networks deployment environment, where two kinds of sensor nodes exist; one is ordinary sensors, which we assume to be same with nodes in [5], where each sensor node is equipped with diverse sensors (temperature, humidity light, and Carbon Monoxide); thus data of different attributes can be gained. The other kind of sensor is mobile nodes, which have the properties of mobility and high energy. Mobile nodes serve as CHs, which will move from random deployed initial position to the corresponding destinations calculated by k-means algorithm. The proposed k-means based clustering algorithm can achieve minimum energy consumption of intracluster communication. The function of CH is performing intracluster fusion and local decision, and it also serves as delay for sending the local decision results to the base station. In order to enhance the accuracy of event detection, we propose a *fusion support degree* for each cluster, which means the probability for a cluster to participate in the decision-level fusion for final decision of emergency event. The clusters of which the value of fusion support degree is less than a predefined threshold will be rejected from the data fusion, while the value serves as weight of local decisions when performing decision-level fusion. The fusion support degree is determined by distance between CH and event centre and remaining energy of CH and cluster members. Finally, the base station will perform decision-level fusion according to local decisions and *fusion support degrees* uploaded by CHs of clusters will join the data fusion to make final decision and make corresponding warning alarms.

The rest of paper is organized as follows: we present our related work in Section 2. Section 3 is some preliminaries before we introduce our cluster based fuzzy decision fusion algorithm (CFDF). Section 4 gives the CFDF overview and detailed descriptions. In Section 5, we evaluate the performance of the proposed algorithm. We conclude this paper in Section 6.

2. Related Work

Numerous researches have been done in the field of data fusion and clustering. In this section, we give some review of the related work.

Khaleghi et al. [2] give a critical review of data fusion state-of-the-art methodologies; a new data centric taxonomy of data fusion methodologies was introduced and the challenging aspects and existing algorithms in each category are discussed. The date-related data fusion algorithms can be divided into dealing with data imperfection, data correlation, data inconsistency, and data disparateness. Generally, researches and algorithms about data fusion mainly focus on three levels [6]. The lowest level fusion is data-level fusion, of which the processing is to merge the observed raw data. Raw sensor data can be directly combined if the sensor data are commensurate. Techniques for raw data fusion typically involve classic detection and estimation methods such as averaging method. However, if the data are uncorrelated, the fusion should be performed at a feature or a decision level. Feature-level fusion can be performed after data-level fusion or directly performed by a single sensor. In feature-level fusion, features are extracted from multiple sensor observations and combined into a single feature vector for decision-level fusion. Decision-level fusion is the highest level fusion, of which the main idea is fusion of local decisions of each sensor; the local decision is a preliminary determination of an entity's location, attributes, identity, and so on. The most commonly used decision-level fusion methods include voting techniques, Bayesian inference, and Dempster-Shafer's method [6].

Gók et al. in [7] proposed a decision-level fusion algorithm based on fuzzy logic used for single target classification. The research adopts kNN to make local decisions, which might cost a lot of time because it needs quite a few number of training samples to train classifiers. At each node, a fusion chance will be calculated according to the distance and energy of received information using a fuzzy method. The proposed decision-level fusion method achieves high efficiency on the occasion that the number of sensors is quite small. However, when a large number of sensor nodes are needed in a given sensing field, using every sensor to conduct local decision and uploading decision results will result in massive extra information processing and communication energy consumption. Grouping the sensors into several clusters could be a good way to solve the above problem, in that the majority of data processing and transmission could be done at CHs.

Clustering algorithms in the literature vary in their objectives; often the objective is related to application requirements [3]. For example, if the application is sensitive to data delay, the connectivity and length of data routing paths are usually considered as critical problems. Some other popular problems such as load balancing, fault-tolerance, minimal cluster count, and maximal network longevity are also concerned. In the field of distributed clustering algorithms, each sensor executes the algorithm independently and their states are based on the cluster membership decisions on their own states. Low Energy Adaptive Clustering Hierarchy (LEACH) [8] is the most popular distributed clustering algorithm; a dynamic adaptive probability-based CH selecting mechanism was proposed. Unlike the LEACH of which the selection of CHs is random, hybrid energy-efficient distributed clustering (HEED) [9] proposes a CH selecting method considering a hybrid of energy and communication cost. DWEHC in [10] proposed by Ding et al. can achieve more aggressive goals than those of HEED, the weight of CH is based on its residual energy and distance to its neighbors, and the largest weight node in a neighborhood may become a CH. Simulation results illustrated that DWEHC can do better in load-balance and has $O(1)$ time complexity.

Manjunatha et al. in [5] suggested a data fusion method based on clustering for environment monitoring. In the proposed method, each sensor is equipped with diverse sensors, and the processing and fusion of these diverse sensor signals are carried out by CHs using proposed fuzzy rule based system. The reliability and accuracy of environmental detection are gained by multiple data fusion. However, this paper did not give a complete data fusion processing for it only considered fusion of intracluster and multiple sensing variables, but the decision-level fusion of intercluster was not mentioned.

3. Preliminaries

In this section, we will give a brief introduction to some mathematical models used in our proposed cluster-based fusion model and also give some assumptions under the monitoring environment of this paper.

3.1. A Brief Introduction to k-Means Algorithm. k-means clustering [11] is one of the simplest unsupervised learning algorithms that solve the clustering problem. The specific objective of k-means is to classify a given data set into a certain number of clusters (assume k clusters). The main idea is to define k centroids randomly or in a cunning way beforehand and then assign each data point to the closest centroid to be a member of the current cluster; the first round is completed after all the data points are assigned. By now k clusters are constructed and next the second round will be performed by recalculating k new centroids as barycenters of the clusters resulting from the previous round, and then a new binding has to be done between the same data set points and the nearest new centroids. And then, do this loop until centroids do not move any more.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2, \tag{1}$$

where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres. The specific steps of k-means are described as follows [12]:

(1) Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.

(2) Assign each object to the group that has the closest centroid.

(3) When all objects have been assigned, recalculate the positions of the k centroids.

(4) Repeat Steps (2) and (3) until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

3.2. The Fuzzy Theory Introduction. The concept of fuzzy set was proposed by Zadeh in 1965 [4], and the introduction of fuzzy theory is to deal with problems involving knowledge expressed in vague or linguistic terms especially when the boundary of the set contained in the universe is ambiguous, vague, or fuzzy [5]. In general crisp set, the relation between an element and a given set is that the element is either belonging to the set or not. However, in a fuzzy set, the element is not definitely belonging to a set or not; instead some ambiguity or vagueness may exist; thus elements belong to a fuzzy set to a certain degree represented by a number in interval $[0, 1]$. The number is called membership degree; the larger the value of membership degree is, the greater the degree of the element belonging to the set is. When the values of all the elements are either 0 or 1, the fuzzy set will degenerate into a crisp set.

Considering a set X whose element is represented by x, the membership function $\mu_{\widetilde{A}}(x)$ is a function that associates with each x in X with a membership degree in $[0, 1]$. A fuzzy set A in X is a set of ordered pairs and is given as $\widetilde{A} = \{x_i, \mu_{\widetilde{A}}(x)\}$, $x_i \in X$, where x_i represents the element in a fuzzy set A.

Fuzzy logic inference is a process using the fuzzy input and fuzzy rules to obtain the fuzzy output, of which the substance is to map the input space into a given output space by fuzzy roles. The fuzzy roles normally use the IF/THEN statement, the format of which is IF x is A Then y is B, where x, y are the input linguistic variable and output linguistic variable, and A, B are the linguistic value of fuzzy set before reasoning and after reasoning, respectively. Mamdani fuzzy inference system proposed by Ebrabim Mamdani in 1975 is a widely used fuzzy inference method, which can implement the reasoning computation from input to output effectively through a serious of predefined fuzzy roles.

3.3. Assumptions. In this paper, we consider a cluster-based data fusion algorithm for event detection; our research is based on the following assumptions:

(1) To simplify the complexity of the problem, we assume the sensing field is a flat two-dimensional region; no barriers exist; thus the mobile nodes can smoothly move inside the sensing area. And also the nodes are

uniformly deployed in the sensing filed and cannot be recharged after deployment.

(2) In consideration of load-balance problem, we adopt a heterogeneous WSN composed of two kinds of sensor nodes, general static nodes and mobile nodes; the mobile nodes will act as CHs and the energy of which is preset higher than that of general nodes, because CHs will consume more energy on data transmission and processing.

(3) Each sensor node can achieve its location by a GPS module or some other techniques such as the method proposed in [13]. The locations will be forwarded to BS only once and stored by BS for the subsequent clustering and fusion calculation.

(4) Like LCA [14], a single-hop intracluster topology is established by our proposed k-means based clustering algorithm, and TDMA is used for intracluster communication. Intercluster uses multiple-hop routing to arrive at the sink node.

4. Cluster Based Fuzzy Fusion Algorithm (CBFFA)

In this section, we will introduce our proposed cluster-based fuzzy fusion algorithm (CBFFA). Considering the time complexity of clustering may decline the efficiency of event detection, we proposed a centralized none-event driven clustering method; namely, the clustering is performed before some event occurs. Algorithm 1 illustrated the pseudocode of CBFFA.

Our proposed CBFFA includes three phases, the initial clustering phase, event detection phase, and finally the event warning phase. In initial phase, the sensor nodes (including static nodes and mobile nodes) are randomly deployed in the two-dimensional sensing filed, and each sensor sends its position coordinate to the BS via multihop routing. Therewith the k-means based clustering algorithm (kBCA) will be conducted. We assume that the number of mobile nodes will be larger than that of clusters, and the remaining mobile nodes which are not selected to be the CH will serve as spare nodes. The detailed description of the proposed k-means based clustering algorithm (kBCA) is introduced in Section 4.1. After dividing the nodes in sensing filed into k clusters, the event detection will be performed. When an emergency arises, it will be detected by the sensors around immediately. In this paper, we assume the location of the event centre to be the position coordinates of the sensor which first detects the event.

It is obvious that a considerable number of sensors in each cluster can achieve the sensing data for event detection. In this paper, our objective is to select the clusters of which the sensing accuracy is relatively high to carry out data fusion, thus to improve accuracy of event detection. Meanwhile, for prolonging the lifetime of the whole WSN, the clusters having relatively low energy are not recommended to join in the data fusion. Instead, they will switch into sleeping mode.

The accuracy of the monitoring data may vary with the distance between sensor and the event centre; when the distance increases the accuracy will decline. In addition, the remaining energy of sensors also has an influence to the accuracy of the event detection, considering both sending and receiving information will consume energy of sensors, when a sensor has little energy and cannot be recharged during transmitting, the sensing data may fail to be sent; thus the transmission reliability is closely related to the remaining energy of sensors. We proposed a fuzzy theory based method to decide the chance (fusion support degree) of a cluster to participate in the data fusion; the fusion support degree is calculated in the BS according to the uploaded distance value between CH and event centre and average remaining energy value of a cluster. The specific calculation process of fusion support degree is described in Section 4.2. When the fusion support degree is less than a predefined threshold, which means the cluster is either quite far from the event centre or lower in energy, as explained before the sensing data accuracy of these clusters will be very low. Therefore, these clusters will be rejected from data fusion and enter into sleeping mode. The clusters of which fusion support degree is larger than threshold will conduct data transmission and fusion. Here, the fusion support degree is a value that explains the credibility of a cluster's local decision; the higher the value is, the more contribution the local decision will bring to the final decision-level fusion result. Finally, the final decision-level fusion will be conducted at the BS and afterwards the corresponding early warning information will be made.

4.1. k-Means Based Clustering. In this section, we will introduce the k-means based clustering algorithm (kBCA). The k-means clustering algorithm can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers. And for sensor nodes, the transmission energy consumption is closely dependent on transmission distance, which is proportional to squared Euclidean distance or quadruplicate Euclidean distance [15]. Therefore, kBCA will have good performance for decreasing intracluster energy consumption. However, the initialized centroids and the value of k are not specified in the algorithm. Usually, the initial centroids are determined by random way and k is a man-made value. In this paper, we adopt a cunning way to determine the value of k and the centroids' initial position; that is, we selected the least possible value of k to guarantee the network connectivity and initial positions of centroids are suggested to be uniformity distributed and far from each other. Algorithm 2 illustrated the pseudocode of kBCA.

First, we should determine the number of clusters according to size of a cluster. We adopt the mean of numbers of static nodes inside the circle centred by each of N_k mobile nodes to be the size of a cluster. In consideration of connectivity intercluster, the radius of the circle is suggested to be $R_c/4$. In order to increase the convergence speed of k-means, the initial centroids should be far from each other; thus we approximately divide the sensing field into k number of squares and let centres of them be initial centroids. After running k-means algorithm, the convergent centroids are achieved, and

```
Notations:
BS: the base station
$k_i$: the number of static nodes in cluster $i$
$E_{\mathrm{CM}}$, $E_{\mathrm{CH}}$: initial energy of cluster members and CH respectively
$\overline{E_{\mathrm{CM}_i}}$: average remaining energy of members in cluster $i$
$p_{\mathrm{CM}}$, $p_{\mathrm{CH}}$: weighting factors of cluster member and CH, satisfying $p_{\mathrm{CM}} + p_{\mathrm{CH}} = 1$
$\overline{E_{C_i}}$: average remaining energy of cluster $i$
$P_i$: a predefined threshold to determine a cluster will join fusion or not
$k$meansBasedClustering(nodes): the clustering algorithm based on $k$-means proposed in this paper
FusionSupportDegree(CH$_i$): the chance of a cluster to participate in the final decision-level fusion
LocalDecision(CH$_i$): intra-cluster fusion and local decision performed by CH$_i$
DecisionLevelFusion(CHs): the process of decision-level fusion conducted by BS
Initialization clustering phase:
Deploy all the nodes (both the static nodes and mobile nodes) randomly in the sensing filed.
For each static node do
    Send its location $(x_i, y_i)$ to BS via multi-hop routing
    Vector_staticNodes($i$) $\leftarrow (x_i, y_i)$
End for
For each mobile node do
    Send its location $(x_{ci}, y_{ci})$ to BS via multi-hop routing
    Vector_mobileNodes($i$) $\leftarrow (x_{ci}, y_{ci})$
End for
$k$meansBasedClustering(matrix_nodes);
Event detection phase:
If an event occurs
    Then the first node that detect the event sent its location $(x_{\mathrm{event}}, y_{\mathrm{event}})$ to BS var multi-hop routing
    For each cluster head CH$_i$
    $d_{ci} \leftarrow \sqrt{(x_{ci} - x_{\mathrm{event}})^2 + (y_{ci} - y_{\mathrm{event}})^2}$
    End for
    For each CM$_j$ in $C_i$
        Send its remaining energy $E_j$ to $C_i$
    End for
    For each $C_i$
    $\overline{E_{\mathrm{CM}_i}} \leftarrow \frac{1}{k_i}\sum_{j=0}^{k_i} E_j$,   $\overline{E_{C_i}} \leftarrow p_{\mathrm{CM}} \frac{\overline{E_{\mathrm{CM}_i}}}{E_{\mathrm{CM}}} + p_{\mathrm{CH}} \frac{E_{\mathrm{CH_i}}}{E_{\mathrm{CH}}}$
      Send $\overline{E_{C_i}}$ to BS
   End for
   For each cluster head CH$_i$
      $P_i \leftarrow$ FusionSupportDegree(CH$_i$);
      If $P_i > P_s$
        Then
          $D_{C_i} \leftarrow$ LocalDecision(CH$_i$);
          $D_{C_i}$ is sent to BS
      Else
          $C_i$ turns on sleep mode
      End If
   End For
   $P_{\mathrm{final}} \leftarrow$ DecisionLevelFusion(CHs);
End If
Event alert phase:
BS makes the corresponding early warning information according to decision-level fusion result.
```

Algorithm 1: Pseudocode of cluster based fuzzy fusion algorithm (CBFFA).

next the mobile nodes server as CHs should move to these centroids thus to ensure minimizing the energy consumption of intracluster communication. A variety of methods can be used to choose CHs and match them to the locations of centroids; the most common one is greedy method; for each of the k centroids, the nearest mobile node will be chosen as a CH and move to location of the current centroid. To achieve global optimum, Hungarian method can be used to obtain

```
Notations:
$R_c$: communication range of the CH
$N_k$: a selected number of mobile nodes to determine number of clusters $k$
Static_nodes: number of static nodes
Input: Coordinates of static nodes Vector_staticNodes
Determine value of $k$:
(1) For $i = 1 : N_k$
(2)   For all static nodes $(x_j, y_j)$ If
(3)     $\sqrt{(x_j - x_{ci})^2 + (y_j - y_{ci})^2} \le \frac{R_c}{2}$
(4)     Static_Number($i$)(++);
(5)   End for
(6) End for
(7) cluster_size $\leftarrow \frac{1}{N_k}\sum_{i=1}^{N_k}$Static_Number($i$)
(8) $k \leftarrow \frac{\text{static\_nodes}}{\text{cluster\_size}}$
(9) Divide the sensing field into $k$ squares, and order the centres of squares to be initial positions of centroids.
Determine converged centroids using $k$-means:
(1) Let initial initial positions of centroids to be $m_1, m_2, \ldots, m_k$
(2) Until there are no changes in any mean
(3)   Use the estimated means to classify the samples into clusters
(4)   For $i$ from 1 to $k$
(5)     Replace $m_i$ with the mean of all of the samples for cluster $i$
(6)   End For
(7) End Until
Motion period for CHs:
(1) Select $k$ mobile nodes and dispatch them to locations of $m_1, m_2, \ldots, m_k$ using greedy method or Hungarian method.
```

Algorithm 2: Pseudocode of k-means based clustering algorithm (kBCA).

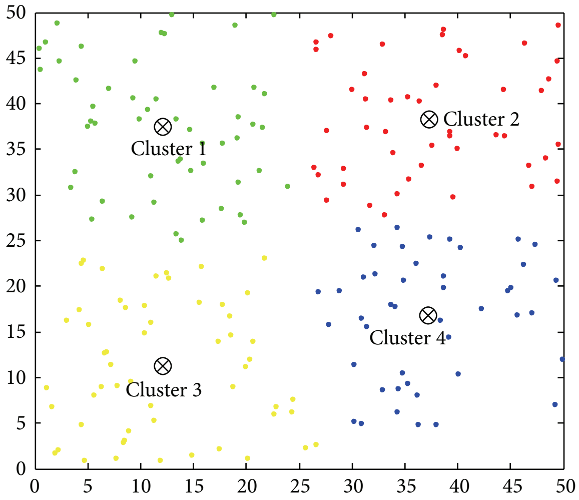


Figure 1: Clustering result of kBCA.

the optimal matching which can minimize the overall moving distance of mobile sensors. Figure 1 shows the clustering result of kBCA.

4.2. The Fusion Support Degree. The fusion support degree is a value that measures the reliability for the measured data of a cluster. As discussed before, the accuracy of event detection is closely related to distance between the sensor and event and the remaining energy of a sensor; thus this paper uses distance and remaining energy to decide the fusion support degree. Generally, we use *far* or *near* to describe a distance and use *high* or *low* to describe energy. However, the linguistic variables suchlike *far* or *high* may include some level of uncertainty and vagueness. Therefore, the final decision result of event may be imprecise if crisp sets of distance and energy are used to deal with data fusion. As fuzzy logic can deal with the knowledge of which the boundaries are not clear through membership function, for this reason we adopt fuzzy logic method to calculate fusion support degree. We adopt a similar method as proposed in [7], using distance and remaining energy as two fuzzy input variables, and fusion support degree as fuzzy output. Figure 2 illustrated the structure of fuzzy logic method, where the distance and energy are crisp input variables and fusion support degree is crisp output after defuzzification.

To simplify the analysis for problem, trapezoidal and triangular membership function is defined to describe fuzzy input variables *distance* and *energy*. The membership functions and corresponding linguistic states of distance and energy are shown in Figures 3(a) and 3(b).

Figure 4 shows the membership functions and corresponding linguistic states of fuzzy output variable, namely, fusion support degree. We introduce the same membership

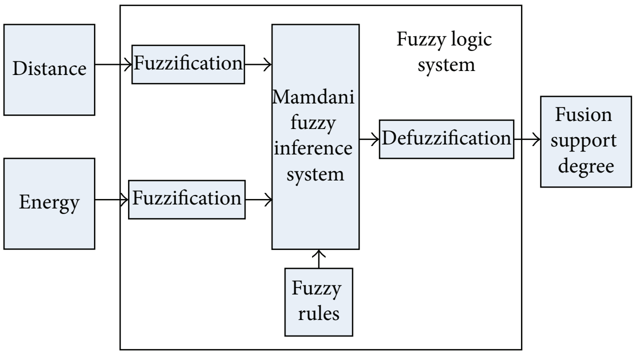


Figure 2: Structure of fuzzy logic method.

Table 1: Fuzzy rules for our fuzzy algorithm.

Rule number	Distance	Energy	Fusion support degree
1	Far	High	MM
2	Far	Medium	ML
3	Far	Low	LL
4	Medium	High	HL
5	Medium	Medium	MH
6	Medium	Low	LM
7	Near	High	HH
8	Near	Medium	HM
9	Near	Low	LH

functions used in [7]. The fusion support degree is divided into nine levels, namely, LL (Low in Low), LM (Medium in Low), LH (High in Low), ML (Low in Medium), MM (Medium in Medium), MH (Medium in High), HL (Low in High), HM (Medium in High), and HH (High in High), and the fusion support degree they represent is successively larger.

Table 1 illustrated fuzzy rules for our fuzzy algorithm. In research [7], the authors consider a sensing environment with obstacles; thus even the sensor is relatively near from the target; the sensing accuracy may be low due to obstacles. Therefore, they give more weight on energy fuzzy variable than distance when dealing with fuzzy output. In this paper, we assume no obstacles exist, so the accuracy is largely related to distance; consequently in most cases we give fuzzy input variable distance more weight. However, under the circumstances that remaining energy of cluster is quite low, to prolong the lifetime of networks, the cluster having low energy is not suggested to join the fusion; thus we give more weight on energy when the remaining energy is low.

After fuzzy variables and fuzzy rules are designed, a fuzzy inference method should be used to achieve fuzzy output, thus to get the final system outputs (FusionSupportDegree(CHi) in Algorithm 1) through defuzzification. In this paper, we use the most commonly used fuzzy inference system, namely, Mamdani method, to obtain fuzzy output.

4.3. Local Decision. We take fire detection as an example of event detection. In this scenario, fire centre can be modeled as a fixed coordinate point in the sensing filed. Each sensor has an integrated multiple sensors module which can collect environmental information such as temperature, humidity, light intensity, and the amount of Carbon Monoxide (CO) [5]; thus the sensing data structure of a sensor can be described as a multidimensional vector $(v_1, v_2, \ldots, v_n)$, where v_i $(1 \leq i \leq n)$ stands for the value of ith environmental information variable.

After fusion support degrees of each cluster are calculated by BS, the clusters whose fusion support degrees are less than the threshold will be eliminated from fusion processing, which means they will not conduct data uploading and data fusion, while the clusters whose fusion support degrees are larger than the threshold will perform local decision and upload decision results thus to join the final decision-level fusion. The local decision of a cluster includes two steps. Firstly, all sensors inside a cluster send their sensing data to CH; then intracluster fusion is performed by CH. Secondly, according to the fusion results, the CH will make the local decision, which reflects evaluation of emergency level for the current cluster to event.

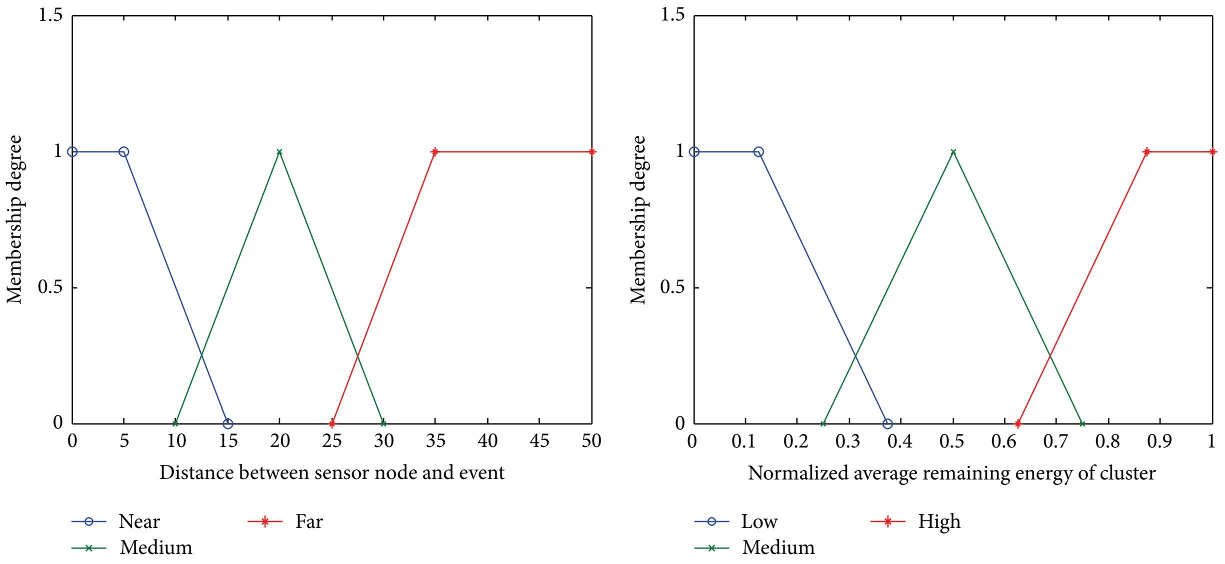
When all sensors inside a cluster have sent their sensing data to the CH, intracluster data fusion is conducted. We use the mean of sensing values of each dimension to be intracluster fusion result. Considering that some sensors might have a possibility of being damaged during deploying, which will reduce the accuracy of sensing data, thus these singular sensors should be eliminated before data fusion.

Note that the multidimensional vector of cluster i after fusion is represented by $(\overline{v_{i1}}, \overline{v_{i2}}, \ldots, \overline{v_{in}})$. The mean of these n variables will be used to conduct local decision; however it is not easy to achieve a quantitative equation for local decision result; therefore we adopt the fuzzy logic method same as proposed in Section 4.2 to get local decision result by fuzzifying each dimension of $(\overline{v_{i1}}, \overline{v_{i2}}, \ldots, \overline{v_{in}})$ to be fuzzy input and defining decision result as fuzzy output variable. This paper did not design the specific fuzzy sets and rules for fuzzy input and output, so that the designing of fuzzy sets is closely dependant on the type of events; when event changes, fuzzy input and fuzzy rules will change as well.

4.4. Decision-Level Fusion. The result of local decision discussed in Section 4.3 is a percentage value calculated by a single cluster which illustrated the possibility an event will happen or the degree of emergency. The decision-level fusion means combining local decisions from several clusters to get a final evaluation of the event. The fusion support degree and local decision result introduced before are prepared for decision-level fusion, and (2) shows calculation of final decision-level fusion result:

$$P_{\text{final}} = \frac{\sum_{p_i > p_s} P_i D_{ci}}{\sum_{p_i > p_s} P_i}, \tag{2}$$

where P_i denotes fusion support degree of cluster i and D_{ci} denotes local decision result of cluster i. If P_{final} is larger than a threshold, BS will make the corresponding early warning information according to fusion result.



(a)

(b)

Figure 3: (a) Fuzzy sets for distance between sensor and event. (b) Fuzzy sets for remaining energy.

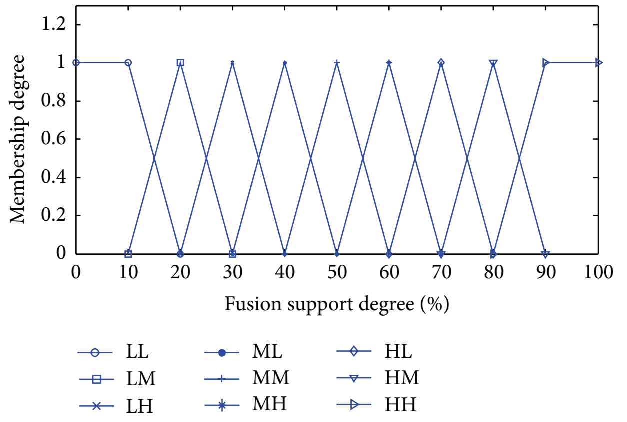


Figure 4: Fuzzy sets for fuzzy output variable fusion support degree.

5. Results and Discussion

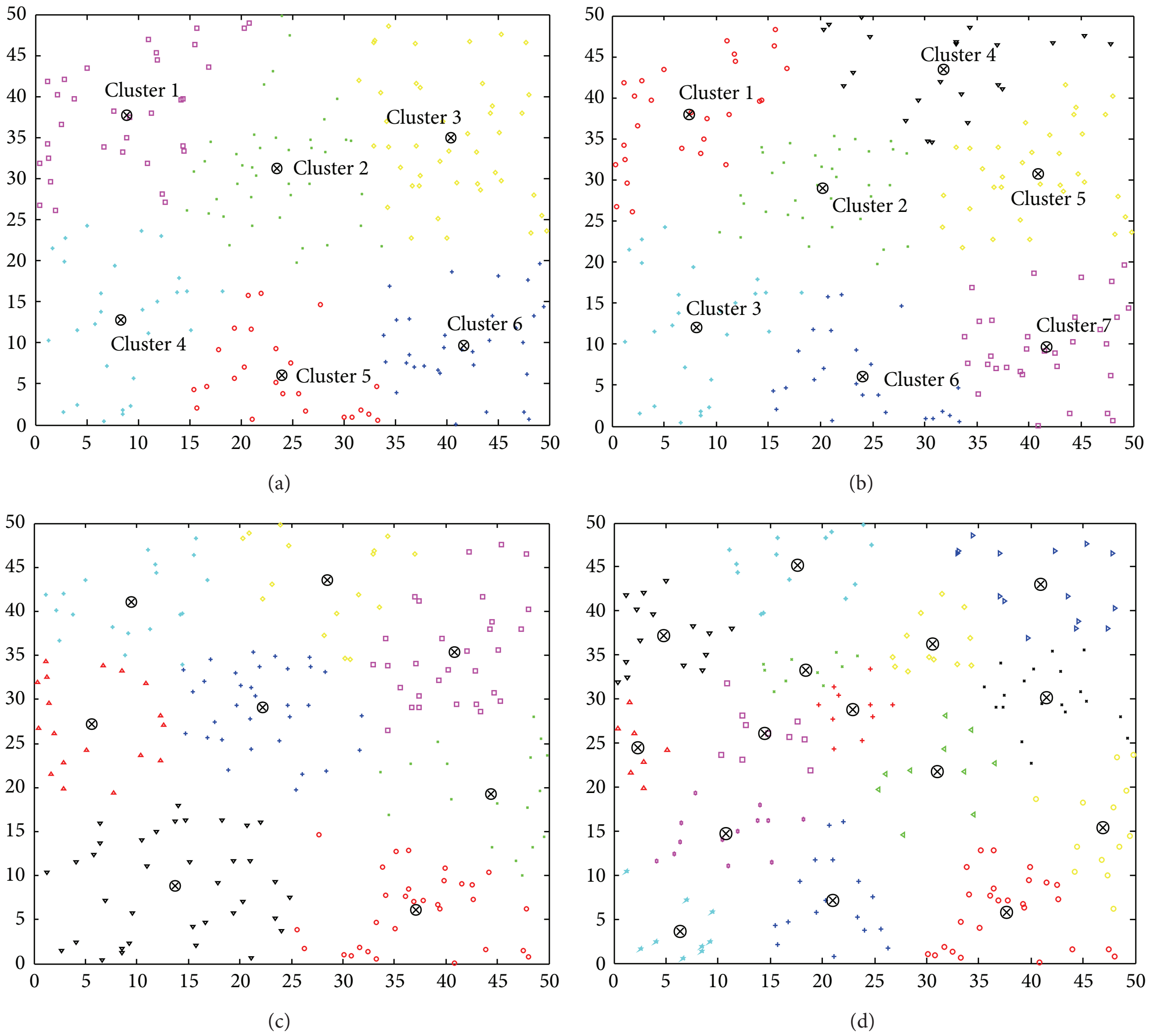
In this section, we will evaluate the performance of our proposed cluster-based fuzzy decision fusion algorithm (CFDFA). The simulation is implemented in Matlab. We assume the sensing area to be a flat two-dimensional square region of which the size is 50 m × 50 m, 200 static sensor nodes and 15 mobile nodes are randomly deployed on the sensing area, and the communication radius (sensors can communicate with each other when their distance is less than communication radius) is assumed to be 30 m. We take fire detection as an example of event detection, and the sensing data structure of a sensor is defined as a multidimensional vector (temperature, light intensity, and CO density).

5.1. Analysis of k-Means Based Clustering Algorithm (kBCA). Firstly, we give a simulation of the initial clustering algorithm kBCA. According to **Algorithm 2**, we order $N_k = 5$; then the cluster size is estimated by (3), of which the value is 35.5. The value means the maximum size a cluster could be so as to meet the connectivity of the network. Consequently, the number of clusters should be more than the value of 200/35.5 = 5.6:

$$\text{cluster\_size} \longleftarrow \frac{1}{N_k}\sum_{i=1}^{N_k}\text{Static\_Number}(i). \qquad (3)$$

Figure 5 shows the clustering results when the number of cluster changes. It is obvious that, with the increasing of cluster number, the number of members in a cluster reduces consequently. Therefore, the average communication energy consumption of a cluster will reduce.

In order to analyze lifetime of the whole network, **Figure 6** gives a quantitative analysis of overall communication distance and variance of cluster size when number of clusters changes. When the cluster number changes from 5 to 15, there is a declining trend of the overall distance, which illustrates that the larger the cluster number is, the more energy-efficient the network will be. Generally, the lifecycle of whole network in a clustering environment is not only related to the overall communication consumption, but also related to balance of cluster size. A balanced clustering means that clusters have approximate sizes, thus to guarantee balance of energy consumption between different clusters. As shown in **Figure 6**, the variance of cluster sizes changes irregularly when the cluster number is less than 10; however, when cluster number is larger than 10, the variance has a decline trend. Above analysis concludes that number of clusters will definitely influence the lifecycle of whole network. In some certain circumstance, the more the cluster number is, the more energy-efficient and load-balanced the network will be. However, increasing of cluster number will cause a corresponding increasing of



(a)

(b)

(c)

(d)

FIGURE 5: Clustering results of *k*BCA when cluster number *k* changes. (a) $k = 6$. (b) $k = 7$. (c) $k = 8$. (d) $k = 15$.

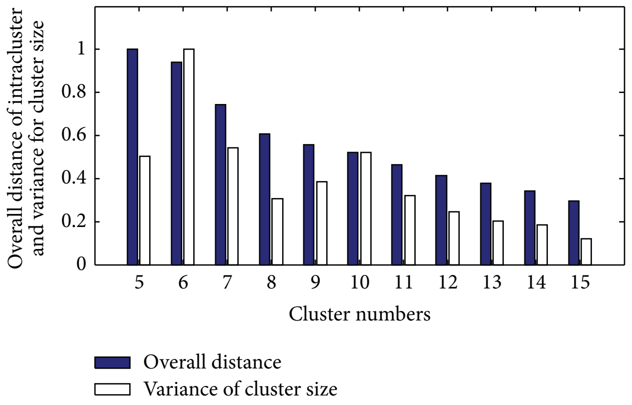


FIGURE 6: Overall distance of intracluster and variance of cluster size when cluster number *k* changes.

mobile nodes; generally mobile nodes are more expensive than ordinary nodes; thus with the increasing of cluster number, the cost of whole network will be increased either. So there is a tradeoff between lifetime and cost of the network; the number of cluster should be designed according to the actual situation.

5.2. Simulation for Event Detection. To make the compromise between the lifecycle and expense of network, the cluster number is proposed to be 12. Figure 7 shows the assumed event detection scenario, 12 clusters are formed by *k*BCA, and a random event occurs with a location of (28,18). Next, the fusion support degree of each cluster is calculated by fuzzy method introduced in Section 4.2. We use the fuzzy logic toolbox on Matlab to simulate the fuzzy output. Table 2 illustrates the fusion support degree results of 12 clusters, in which the distance means distance between fire centre and CH and energy means the average residual energy of a cluster. We define the threshold of fusion support degree to be 40%. Therefore, cluster 5, cluster 8, and cluster 10 will be rejected from data fusion and enter into sleeping mode; the data

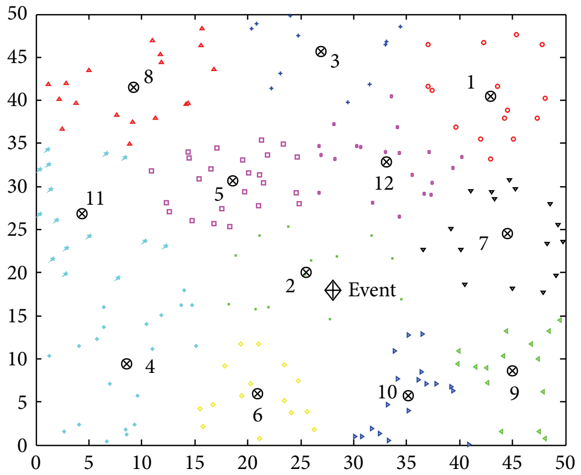


Figure 7: Event detection scenario when cluster number is 12.

Table 2: Fusion support degree of clusters.

Cluster number	Distance	Energy (%)	Fusion support degree (P_i) (%)
1	27.03	0.83	61.6
2	3.17	0.35	68.6
3	27.72	0.73	57.3
4	21.24	0.66	62.7
5	15.76	0.13	20
6	14.00	0.85	74.9
7	17.74	0.69	65.2
8	30.08	0.35	32.7
9	19.41	0.45	60
10	14.27	0.23	21.4
11	25.27	0.48	58.7
12	15.69	0.95	70

fusion and event detection processing are performed by the remaining 9 clusters.

In the simulation of local decision, we use the same fuzzy sets of inputs and output variables as proposed in [5]. $3^3 = 27$ rules are designed; some of the example rules are as follows:

> IF Temperature is low and light intensity is low and CO is low THEN fire probability is vary_low;
>
> IF Temperature is high and light intensity is high and CO is high THEN fire probability is vary_high;
>
> IF Temperature is medium and light intensity is high and CO is high THEN fire probability is high.

Table 3 is the local decisions for clusters, which means the fire probability evaluated by the current cluster.

Table 3: Local decisions for clusters.

Rule number	Fusion support degree	Intracluster fusion result	Fire probability (D_{ci}) (%)
1	61.6	(21, 322, 52)	31.5
2	68.6	(80, 875, 85)	92.2
3	57.3	(30, 300, 38)	44.5
4	62.7	(23, 225, 41)	32
6	74.9	(60, 810, 80)	91.6
7	65.2	(40, 355, 81)	75
9	60	(35, 450, 60)	62.5
11	58.7	(20, 268, 50)	25
12	70	(38, 303, 71)	75

Finally, the final decision-level fusion is calculated by formula (2); the result is $P_{\text{final}} = 60.5\%$, meaning that there is a probability of 60.5% that a fire disaster could happen. The final result and corresponding early warning information will be sent to users for disaster prevention.

5.3. Algorithm Complexity Analysis. At last, we will give the computation complexity evaluation of the proposed CFDFA. It has been calculated that the time complexity of k-means is $O(tkdn)$, where t is number of iterations, k is number of clusters, n is number of data objects, and d is the dimension (equals 2 in this paper). The time complexity of event detection is $O(n + 4k + 2m + 1)$, where m is number of clusters taking part in data fusion. Therefore, the total time complexity is $O(n)$, which means the proposed algorithm is completed in linear time; thus it is efficient and enables scalability when dealing with large data sets. Moreover, as the clustering is performed once the sensors are deployed before event occurs, the time consumption of clustering can be ignored when analyzing the efficiency of event detection; thus the proposed algorithm is efficient enough for detecting and reporting sudden events.

6. Conclusions

In order to enhance the accuracy of event detection, this paper proposed a cluster-based data fusion algorithm using fuzzy logic method (CFDFA) for event detection in a heterogeneous wireless sensor network. In terms of the energy-efficient problem in WSN, we suggested a k-means based clustering algorithm (kBCA); the clustering algorithm can efficiently reduce energy consumption of intracluster, and simulation results illustrated that when a proper cluster number is chosen, well load-balance of cluster size can also be gained; thus the lifetime of network can be prolonged. A crisp form of possibility of event occurrence can be achieved by base station after CFDFA is performed; we take fire detection as an example to simulate our event detection algorithm and get crisp value of the fire probability.

In the future, we may consider more complicated scenarios where the event is no longer static; thus an event-driven dynamic clustering algorithm will be designed. Moreover, we may work on more details about the efficient intracluster data fusion algorithms suitable for our event detection algorithm, where singular point detection may be a key point to be considered.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by National Natural Science Foundation under Grant 61371071, Beijing Natural Science Foundation under Grant 4132057, and Academic Discipline and Postgraduate Education Project of Beijing Municipal Commission of Education.

References

[1] B. Premamayudu, V. Rao, and S. Varma, "A key management & establishment scheme in heterogeneous wireless sensor networks (HWSN)," *International Journal of Applied Engineering Research*, vol. 9, no. 20, pp. 6813–6821, 2014.

[2] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: a review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[3] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[4] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[5] P. Manjunatha, A. K. Verma, and A. Srividya, "Multi-sensor data fusion in cluster based wireless sensor networks using fuzzy logic method," in *Proceedings of the IEEE Region 10 and the 3rd International Conference on Industrial and Information Systems (ICIIS '08)*, pp. 1–6, IEEE, Kharagpur, India, December 2008.

[6] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[7] S. Gók, A. Yazici, A. Coşar, and R. George, "Fuzzy decision fusion for single target classification in wireless sensor networks," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ '10)*, pp. 1–8, Barcelona, Spain, July 2010.

[8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[9] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[10] P. Ding, J. A. Holliday, and A. Celik, "Distributed energy-efficient hierarchical clustering for wireless sensor networks," in *Distributed Computing in Sensor Systems*, vol. 3560 of *Lecture Notes in Computer Science*, pp. 322–339, Springer, Berlin, Germany, 2005.

[11] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, University of California Press, Berkeley, Calif, USA, 1967.

[12] http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html.

[13] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM '03)*, vol. 3, pp. 1734–1743, 2003.

[14] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[15] P. Ding, J. A. Holliday, and A. Celik, "Distributed energy-efficient hierarchical clustering for wireless sensor networks," in *Distributed Computing in Sensor Systems*, Lecture Notes in Computer Science, pp. 322–339, Springer, Berlin, Germany, 2005.

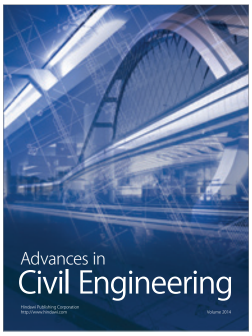
Journal of
Engineering

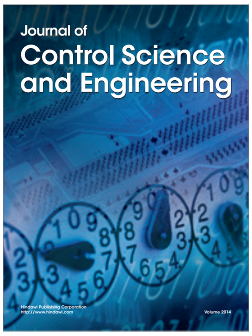
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

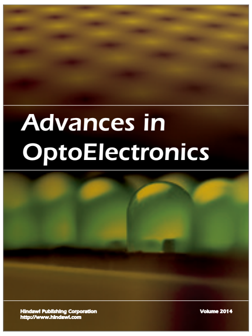
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

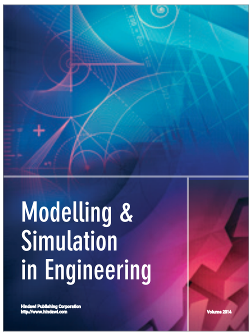
Journal of
Robotics

Journal of
Electrical and Computer Engineering

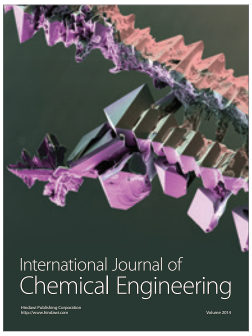

Advances in
OptoElectronics

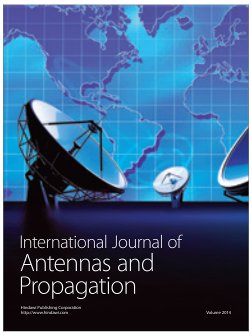
VLSI Design

International Journal of
Navigation and Observation

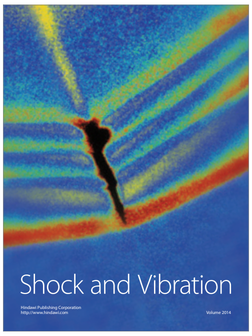

Modelling & Simulation in Engineering

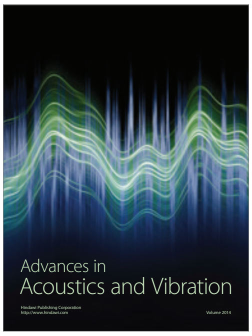
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration