www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

OUTLIER DETECTION BASED ON SIMILAR FLOCKING MODEL IN WIRELESS SENSOR NETWORKS

Cheng Chunling[1,2,3], Wu Hao[1], Yu Zhihu[1], Zhang Dengyin[1,3], Xu Xiaolong[1,2,3]
[1]College of Computer
Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu, China
[2]Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks
Nanjing, Jiangsu, China
[3]Key Lab of Broadband Wireless Communication and Sensor Network Technology
(Nanjing University of Posts and Telecommunications)
Ministry of Education Jiangsu Province
Nanjing, Jiangsu, China
Emails: chengcl@njupt.edu.cn; zhangdy@njupt.edu.cn

Submitted: Aug. 4, 2012 ***Accepted:*** Jan. 15, 2013 ***Published: Feb. 20, 2013***

Abstract- Outlier detection plays a crucial role in secure monitoring in Wireless Sensor Networks (WSN). Moreover, outlier detection techniques in WSN face the problem of limited resources of transmission bandwidth, energy consumption and storage capacity. In this paper, similar flocking model is proposed and a cluster algorithm based on similar flocking model (CASFM) is put forward to

detect outliers in real-time stream data collected by sensor nodes. The similar flocking model improves the Vicsek model by introducing the similarity between individuals and velocity updating rule, which causes similar objects to cluster quickly. In order to save energy, CASFM algorithm preprocesses similar data on the sending sensors first, which greatly reduces the transmission of similar data. So the communication overhead is decreased. With the characteristics of self-organization and fast convergence of flocking model, stream data can be clustered quickly. The experimental results show that the proposed algorithm can detect outliers effectively with less energy consumption.

Index terms*: *Wireless sensor network, Outlier detection, Stream clustering, Flocking model.

I. INTRODUCTION

Wireless Sensor Network(WSN)，which is listed as one of the most influential technology in the twenty-first century, has been used in many fields, such as military control, environmental monitoring and forecasting, health caring, intelligent home, urban transport and security monitoring [1]. In most applications, the accuracy and reliability of data collected by sensors make great contribution to the decision-making. The distributed environments of wireless sensors are different, and some even work in the harsh environment, so that the value of collected data may be deviated. Typically, the data sets which are obviously inconsistent with other data observations are called outliers. In traditional research, outliers are often ignored or treated as noise. However, these "noise" data may convey some important abnormal signal, for example, in environmental monitoring, outliers may indicate impending weather disasters; in military control, the outlier data may mean dangerous invasion; in health caring, the outlier data may indicate bad health conditions; in urban traffic monitoring, outlier data may mean traffic accidents. If we discard outlier without any analysis, it would result in the loss of important information. Therefore, outlier detection is a key task in the WSN and has become a hot issue in the research of WSN.

The resources of sensors are very limited in WSN, and the data from sensors form a large number of distributed real-time data stream, therefore, most traditional data mining methods are not directly applicable to WSN. An outlier detection approach based on similar flocking model in WSN is proposed in this paper. Firstly, we introduce flocking model into WSN. Inspired by the phenomenon of rapid aggregation of individuals in the same population, we propose a similar flocking model. Then, a stream clustering algorithm based on similar flocking model (CASFM) is presented to detect outliers by clustering the real-time data stream collected by sensors. In order to save the resources, especially the energy of sensors, the presented algorithm reduces the transmission of similar data on data collection sensors by using the feedback cluster information from cluster heads. In addition, the cluster algorithm can aggregate stream data rapidly by introducing the similar neighbors and the velocity updating rule in similar flocking model to let similar individuals cluster quickly, so that the computational energy can be saved. Finally, the experiments are carried out. The results show that CASFM can not only achieve a high precision of outlier detection, but also save the energy.

The rest of this paper is organized as follows. Related work on outlier detection techniques based on clustering is presented in Section II. Similar flocking model we proposed is described in Section III. Our outlier detection model in WSN and the stream cluster algorithm based on similar flocking model are elaborated in Section IV. Experimental results and the performance evaluation are reported in Section V. Finally, the paper is concluded in Section VI.

II. RELATED WORK

The outlier detection methods in WSN can be roughly divided into the following categories: statistics based method, nearest neighbor based method, clustering based method, classification based method and spectral analysis based method [2]. Among these methods, clustering based method does not need prior knowledge and has low computational complexity, so it has been proven to be an effective approach to provide better data aggregation and conserve the limited energy resources for large WSNs [3].

Korjani MM, et al. proposed a distributed detection model in WSN, which detected outliers by classifying sensor data through genetic-fuzzy clustering algorithm [4]. Amirhosein T, et al. proposed a communication efficient distributed clustering algorithm for WSN by improving distributed k-means algorithm [5]. In order to reduce the communication overhead, only summarized information was transmitted and computations were performed locally in clusters as much as possible [5]. This algorithm keeps the clustering quality and reduces communication overhead. However, clustering heads still require a large amount of data processing when the number of sensors is relatively high. Hakkilo S, et al. proposed a distributed WSN data stream clustering algorithm based on fuzzy clustering(SUBFCM) [6]. To minimize energy consumption of sensor nodes, SUBFCM followed the strategy of trading-off communication for computation through distributed clustering and successive transmission of local clusters. Masud et al. proposed a distributed anomaly detection algorithm by clustering ellipsoids in WSNs [7]. This algorithm learned an ellipsoid boundary for normal data at each sensor, then clustered these ellipsoids at a global level to model normal behavior and detected unusual events. Rajasegarar et al. proposed a distributed global outlier detection algorithm, in which the goals of detecting outlier and reducing energy consumption were realized by clustering and merging clustering results [8].

Currently, most clustering algorithms in WSN focus on designing better network topologies or routing protocols to extend network lifetime by clustering sensor nodes [9,10], while few algorithms pay attention to clustering real-time data stream collected by sensor nodes to monitor and detect outlier. In the network environment of limited resource, a well-designed algorithm should not only ensure the accuracy of detection, but also meet the requirement of saving network resources. This paper will introduce flocking model to WSN and propose the similar flocking model. Furthermore, the clustering algorithm based on similar flocking model is presented to find outliers in real-time data stream collected by sensor nodes and to realize the anomaly detection and monitoring while controlling the energy consumption.

III. SIMILAR FLOCKIING MODEL

Flocking is a cooperative behavior of social animals such as birds and fish. Through the interaction between individuals, these social animals can automatically line up in groups without centralized coordination and global information, which shows the swarm intelligence. The flocking model simulates the animals’ distribute-controlled aggregation movement to achieve complex functions. The representative models are biod model [11] proposed by Reynolds, Vicsek model [12], leader-follower model [13], and so on. In these models, Vicsek model is a basic model of multi-individual system. It aims to research the clustering behavior in non-equilibrium system and it has some key characteristics of a complex multi-individual system [12]. Flocking model has been used for text clustering by Cui, et al, and the experimental results show that the clustering algorithm based on flocking model has higher accuracy, faster convergence speed than k-means and ant colony algorithm. Moreover, it can cluster dataset for any shape, size and density and the clustering result is stable [14].

The Vicsek model is simple and has the advantage of fast convergence with the flocking essence, so it is very suitable for a resource-limited distributed environment, like WSN. In this paper, we improve the Vicsek model and propose a similar flocking model. By improving the heading rule, similar individuals will have similar moving direction gradually, and by introducing the speed updating rule, similar individuals will cluster quickly. Furthermore, we propose an outlier detection approach based on similar flocking model to detect outliers in WSN by clustering the real-time data stream collected by wireless sensor nodes. Meanwhile, in order to save the energy of sensor nodes, the nodes process the data stream locally and reduce the transmission of similar data.

a. Vicsek Model

Vicsek model [12] is a discrete-time system consisting of multiple autonomous individuals called particles. Each particle moves in a two-dimensional plane at a constant velocity, and its heading angel is the vector average of heading of its neighbors. The initial position of each particle is distributed in the plane randomly, and the initial heading is distributed in the interval $[-\frac{\pi}{2},\frac{\pi}{2}]$. Assume that current position of particle *i* is ($x_i(t)$, $y_i(t)$) at time *t*, and the velocity is *v*. The position updating rule is：

$$\begin{cases} x_i(t+1) = x_i(t) + v\cos\theta_i(t), \\ y_i(t+1) = y_i(t) + v\sin\theta_i(t), \end{cases} \quad i = 1, 2, \ldots, N \tag{1}$$

where $\theta_i(t)$ is the heading of individual *i* at time *t*. The linear form of the heading updating rule is:

$$\theta_i(t+1) = \frac{1}{n_i(t)} \sum_{j \in N_i(t)} \theta_j(t) \tag{2}$$

where $N_i(t)$ denotes the set of neighbors of individual *i* at time *t*, shown as (3). And *n(i)* is the number of individuals in $N_i(t)$.

$$N_i(t) = \{ j \mid d_{ij}(t) < r \} \ , \ r > 0 \tag{3}$$

where $d_{ij} = \sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}$, r is the neighborhood radius·

b. Similar Flocking Model

Vicsek model researches the clustering behavior in multi-individual system, but it does not take into account the difference between species or individuals and its effect on aggregation. Therefore, in order to let the individuals belonging to same species gather quickly and different kinds of individuals separate rapidly, similar flocking model is proposed, in which similarity between individuals and velocity updating rule are introduced. Correspondingly, the heading and the position updating rules are improved. Using similar flocking model to cluster the real-time data stream collected by sensor nodes, data will be treated as individuals and each individual moves according to the position and heading of the similar and nearby individuals. The improved rules are as follows:

Similar neighbors:

$$SN_i(t) = \{ j \mid d_{ij}(t) < r \wedge sim(i, j) > \delta \} \ , \ r > 0, \ 0 < \delta < 1 \qquad (4)$$

where δ is similarity threshold, *sim(i,j)* is the similarity between individual *i* and *j*. It can be described by the value of cosine angle of characteristic vector P_i and P_j: $sim(i, j) = \frac{(P_i, P_j)}{\| P_i \| \cdot \| P_i \|}$.

The heading updating rule is：

$$\theta_i(t+1) = \frac{1}{sn_i(t)} \sum_{j \in SN_i(t)} \theta_j(t) \qquad (5)$$

The heading angle is updated by similar neighbors instead of neighbors, which makes the similar individuals attain a common heading angle quickly. The velocity updating rule:

$$v(t+1) = P_{sim} \cdot v(t) \qquad (6)$$

here, P_{sim} is the similar characteristic factor, $P_{sim} = \frac{\sum_{C \in SN_i(t)} sim(P_i, P_C)}{sn_i(t)}$, $SN_i(t)$ is the set of similar neighbors, and *sn(i)* is the number of similar neighbors.

The position updating rule is：

$$\begin{cases} x_i(t+1) = x_i(t) + v(t+1) \cdot \cos \theta_i(t+1) \Delta t, \\ y_i(t+1) = y_i(t) + v(t+1) \cdot \sin \theta_i(t+1) \Delta t, \end{cases} \quad i = 1,2,\ldots,N \qquad (7)$$

This rule indicates the movement trace of aggregated individuals in space. Δt is the time step, $\theta_i(t+1)$ denotes the heading of next time $t+1$, and $(x_i(t), y_i(t))$ is the position of individual i at time t.

Similar characteristic factor P_{sim} is the mean value of similarity between individual i and its similar neighbors. It is used to control the moving velocity and the position change of i. If the similarity between individuals is high, the velocity is fast and the position updating step becomes larger. In addition, the heading among similar individuals will be consistent gradually. The velocity updating rule is used to regulate the speed of individuals to aggregate similar individuals and separate different individuals quickly. The heading updating rule is used to calculate the next direction of individual i through headings of its similar and nearby individuals, so the similar individuals are getting closer. According to the improved rules, the position of individual at time *(t+1)* is updated through the heading at time *(t+1)*, which makes the individuals' positions be synchronized faster than through the direction at time t in (2). So the similarity between individuals is considered through the similar characteristic factor P_{sim}. Different species of individuals will show big difference in position and heading when they are moving, so that they can be divided into different clusters, and the individuals with higher similarity will be more likely to be in the same cluster.

IV. OUTLIER DETECTION BASED ON SIMILAR FLOCKING MODEL IN WSN

a. Outlier Detection Model in WSN

The outlier detection model proposed in this paper is based on similar flocking model described above and detects outliers by clustering stream data collected from sensor nodes. It can be used in the applications of forest fire prevention, environmental monitoring, and so on.

In wireless sensor networks, hierarchical topology is a typical network topology structure, which is widely used in monitoring systems. The model in this paper is applicable to hierarchical WSN, as shown in Figure 1.

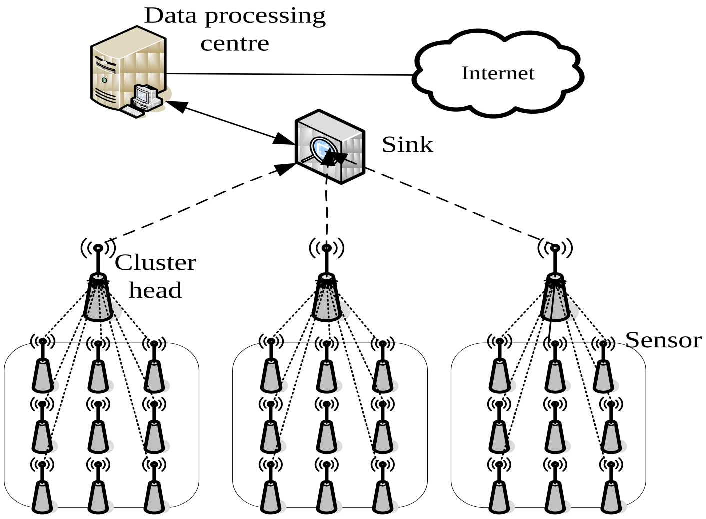

Figure 1. Hierarchical topology structure of WSN

The sensor nodes are responsible for data acquisition, and each sensor node will preprocess the raw stream data locally and send necessary information to its cluster head. After receiving the data sent by the sensors within its region and storing them in cache, cluster head clusters the data stream locally and distinguishes the local outliers. Then it discards the original data from sensors; Afterwards, cluster heads send the local clustering results and local outliers to the sink node. The sink node will gather all local results and cluster them globally to produce global outliers. The software model of outlier detection is shown in Figure 2. The local preprocessing module running on the sensor nodes preprocesses the raw stream data to reduce the amount of data sent to cluster head. The stream clustering algorithm based on similar flocking model is deployed on cluster heads and sink node to detect local outliers and global outliers respectively.

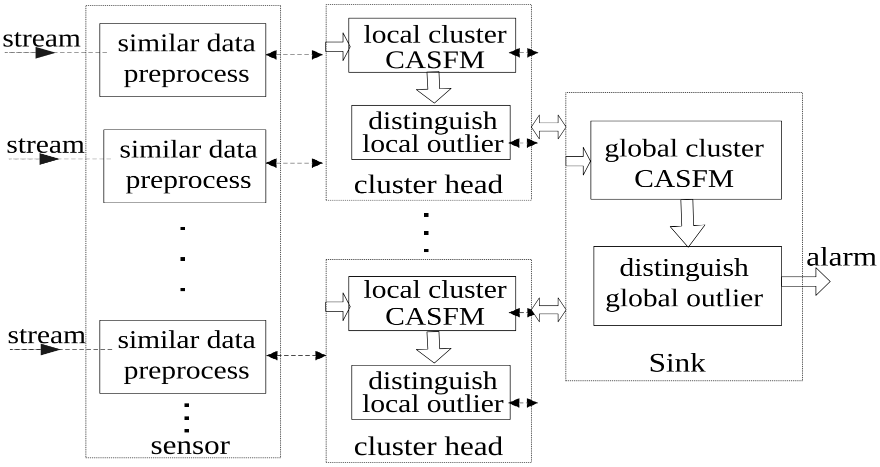

Figure 2. Outlier detection model in WSN

As the energy, processing power and storage capacity of sensor nodes are very limited, in order to prolong the life cycle of the entire network, not only high detection rate and low false alarm rate should be achieved, but also minimum consumption of network resources should be kept, especially the energy consumption. In the model proposed in this paper, the stream clustering algorithm based on similar flocking model running on cluster headers and sink node is simple, effective and fast convergent, so the computational energy consumption will be controlled. By local preprocessing on the sensor nodes and the local clustering on cluster headers, the amount of data transferred between nodes is minimized and the energy consumption of data transmission is reduced.

b. Distinguishing Outliers

There is no rigid mathematical definition of outliers at present, and different detection techniques have different ways to distinguish outliers. Since the method proposed in this paper detects outliers by clustering, a data point or a small cluster which is distant from rest clusters will be defined as an outlier [15]. Relevant definitions are given below:

Definition 1. Cluster characteristic tuple: is summarized information of a cluster. It can be expressed by a tuple consisting of five elements: $CF=<E, d_{total}, d_{max}, d_{min}, \lambda>$.

where E is the center of a cluster, $E = \frac{1}{n}\sum_{i}^{n} P_i$, P_i represents the characteristic vector of data object i in a cluster; d_{total} is the total distance from each data object to cluster's center E; d_{max}, d_{min} are the maximum radius and minimum radius from each data object to the cluster center respectively; λ is the number of data objects in a cluster.

Definition 2. Clustering compact degree: is a compact degree of data objects in a cluster, and its value can be calculated by clustering characteristic tuple CF_c of cluster c, that is:

$$CluCom(c) = \frac{CF_c.d_{\max} - \dfrac{CF_c.d_{total}}{CF_c.\lambda}}{\dfrac{CF_c.d_{total}}{CF_c.\lambda} - CF_c.d_{\min}} \tag{8}$$

here, $\frac{CF_c.d_{total}}{CF_c.\lambda}$ is the average distance from each data object to its cluster center. $CluCom(c)$ reflects the distribution degree of data points in a cluster. If the data points in a cluster are distributed evenly, the value of cluster compact degree is close to 1; On the contrary, the higher the value is, the more irregular data points distribute.

Definition 3. Outlier: If cluster *c* satisfies following conditions, it is considered as an outlier.

(a) $c.\lambda < \xi$

(b) $dist(c) \geq \frac{CF_c.d_{total}}{CF_c.\lambda} + \frac{CF^*.d_{total}}{CF^*.\lambda}$ (9)

(c) $CluCom(c) > 1+\varepsilon,\ \varepsilon > 0$

$dist(c)$ is defined as the exception factor of a cluster. It represents the degree of difference between cluster *c* and other clusters. It can be calculated through the distance between them; $\frac{CF_c.d_{total}}{CF_c.\lambda}$ is the average distance from each data point in cluster *c* to cluster center of *c*; $\frac{CF^*.d_{total}}{CF^*.\lambda}$ is the average distance from the cluster center of *c* to that of other clusters. Condition (a) indicates that the data points in cluster *c* are very few; condition (b) indicates the dissimilar degree of cluster *c*; condition (c) indicates the distribution degree of data points in cluster.

c. Stream Clustering Algorithm based on Similar Flocking Model (CASFM)

CASFM is deployed in the cluster heads and sink nodes. Data points in data stream act as autonomous individuals. They move in the virtual space according to the information of themselves and their similar neighbors, and then evolve into clusters. In the process of stream clustering, each data point will be mapped to a two-dimensional space first. Then it will move according to the position and heading updating rules, meanwhile its moving speed will change based on the velocity updating rule defined in similar flocking model. After certain times of iteration, the data points will gradually be formed to different clusters. If the cluster satisfies the definition of outlier, it would be treated as an outlier; else it would be a potential cluster. Although the behavior of each data point is only influenced by neighbor data, these local rules will result in global behavior of data stream clustering after certain times of iteration, which can produce the final clustering results and generate outliers.

The algorithm process is roughly divided into two phases: cluster initialization and cluster maintenance. During the cluster initialization phase, each sensor sends the real-time stream data collected in the initial cycle to its cluster head, and then cluster head performs an initial clustering process: data points are assigned to a two-dimensional space randomly and then move according to the rules in similar flocking model. After many times of iterations, data points with similar characteristics will be grouped together. So the normal clusters and local abnormal clusters will be produced. Finally, the initial data stream will be discarded to release the storage space.

In the cluster maintenance phase, each sensor preprocesses the raw stream data and then sends the result to the cluster head. Cluster head compares the similarity between received data and the initial clustering results, and then decide whether the points should be added to the known clusters. If they are similar, the algorithm will update the cluster characteristic tuples. Otherwise, the data points will be mapped to the two-dimensional space and move iteratively to get a new clustering result. Finally, the cluster results will be fed back to each sensor.

The cluster head locally clusters data stream collected by the sensor nodes within its field, and then transmits the results to the sink node. The sink node compares the local clustering information coming from the cluster heads, and then clusters the local results again to get global clusters. At last, the algorithm distinguishes outliers according to the clustering results and outputs outliers. The algorithm is described as follows:

Algorithm 1：stream clustering algorithm based on similar flocking model (CASFM)

Input：

X：data stream sent from sensor or cluster head $\{\{x_1^1, x_2^1, x_3^1, \dots, x_m^1\}, \dots, \{x_1^j, x_2^j, x_3^j, \dots\}, \dots\}$,
x_k^j represents the data point from sensor *j* at time *k*.

Output：

CF：the set of cluster characteristic tuples$\{CF_1, CF_2, \dots, CF_{num}\}$; initial value of *CF* is null;

OT：the set of outlier clusters;

OWID：the set of sensors collecting abnormal data;

```
Begin
   if ( firstflag )          // the data points are mapped to the two-dimensional plane at the first run
   {
      MaptoPlane(X);
      Firstflag = false;
   }
   for i=1 to M do {    //Clustering and cluster maintenance; M is the number of iteration
```

```
        for ($x_k^j \in$ X ) {    //for each data point
          move($x_k^j$);        //move by updating rules in similar flocking model
          if type($x_k^j$) = cluster    //if $x_k^j$ is cluster characteristic tuple after local processing
          {
             c = GetMostSimilarCluster($x_k^j$, CF);
             if (c) {                  // c exists
               d = E-distance($x_k^j$, c );
               if( d < $\varepsilon$ )     //if the similarity between $x_k^j$ and cluster c is less than threshold $\varepsilon$
                   form-micro-cluster(CF);
                         //form macro cluster and update the cluster characteristic tuple
             }
             endif  //if(c)
          else        // $x_k^j$ is a real-time collected data point
          {
             ComputerE-distance($x_k^j$, CF);
                //calculate the distance from data point $x_k^j$ to the center of all clusters
             c = GetNearestCluster($x_k^j$, CF);
             if ( c)                  // $x_k^j$ can be merged with cluster c
                merge-cluster($x_k^j$, c); //merge $x_k^j$ into cluster c
             updateCF(c);
          }
          end if
        end for  // $x_k^j$
      end for      // i
      Formcluster(CF);              //produce new cluster characteristic tuples
      for($CF_i \in CF$)  //for each cluster, distinguish whether it is outlier according to the definition 3
         judgeOutlier(OT, OWID);
      Output OT, OWID;
    End
```

d. Data preprocessing on data sending sensors

As data transmission consumes much more energy than computing and data sensing in wireless sensor networks [16], in order to reduce the energy consumption for data transmission, CASFM preprocesses the real-time data stream locally on sensor nodes. According to the clustering results feeding back from cluster head, sensor nodes reduce the transmission of similar data greatly.

The sensors send the raw stream data to the cluster head only in the first cycle, and the cluster head clusters the data and then feeds back the results (the value of cluster center and cluster radius) to the corresponding sensors. The subsequent data collected by the sensors will be compared with the local cluster results fed back from cluster head. If the distance is less than the radius threshold, sensors will not send the raw stream data to the head node. It just updates the cluster characteristic information which will be sent to cluster head periodically. If the distance is greater than the radius threshold, the data will be sent to the cluster head and taken part in the local clustering. As the environment of sensors is stable, the values of collected data are very close. Only when an abnormal event occurs, will the value of collected data change significantly. Therefore, sensor nodes can effectively reduce the amount of data by a similar data processing on collected data stream. This process is also suitable for cluster heads before cluster heads send data to sink node.

e. Analysis of energy consumption

The outlier detection model in this paper is developed for WSN of hierarchical topology, and the total energy consumption includes the following aspects: data collection, preprocessing, transmission and reception on the sensor nodes; data reception, transmission, and algorithm execution on cluster heads; data reception and algorithm execution on the sink node.

The energy consumption model of each sensor node is the first order radio model proposed by Heinzelmann W R [17]. In this model, the energy consumption of sending *k* bits at a distance of *d* meters is:

$$E_T(k,d) = kE_{elec} + k\xi_{fs}d^2 \qquad (10)$$

The energy consumption of receiving *k* bits is:

$$E_R(k) = kE_{elec} \qquad (11)$$

where, E_{elec} is transceiver circuitry energy, and ξ_{fs} is transmitter amplifier energy. As shown above, when the distance is a constant, the amount of transmitted data is the key factor which influences the energy consumption.

Total energy consumption can be calculated by the following equation:

$$E_{total} = E_{comp} + E_{transc} + E_{sen\sin g} + E_{sleep} \tag{12}$$

where $E_{comp} = E_{sensors} + E_{cluster} + E_{sink}$ is the total energy consumption for the algorithm execution. $E_{sensors}$ is the computational energy consumption for similar data processing on sensors. $E_{cluster}$ and E_{sink} denote the energy consumption of cluster heads and sink node for the algorithm execution respectively. Energy consumption of algorithm execution can be simulated by computer.

$E_{transc} = E_{Ss} + E_{Rs} + E_{Sch} + E_{Rch} + E_{Rsink}$ is the total energy consumption of communication among sensors. E_{Ss} and E_{Rs} are the energy consumption for data sending and data receiving on sensor nodes respectively; Likewise, E_{Sch} and E_{Rch} represent the energy consumption for data sending and receiving on cluster heads; E_{Rsink} is the energy consumption for receiving data on sink node. Here, energy consumption for data transmission is calculated according to equation (10) and (11).

$E_{sensing}$ is the energy consumption for detection; E_{sleep} is the energy consumption of sensors on sleep mode.

From equation (12), we can know that when the transmission links and the distance are determined, the communication energy depends on the amount of transmitted data. In order to reduce the communication overhead, the similar data processing algorithm on sensor nodes will reduce the transmission of similar data by only sending quite different data or summarized information to cluster head instead of raw stream data. So the amount of data sent by sensors and received by cluster head is very small. Especially when the nodes in sensor network are distributed densely and the observing areas are overlaid, the collected data tend to have high similarity and redundancy. Therefore, removing these similar, duplicate data can greatly reduce the amount of data transmitted, which can effectively reduce the energy consumption. In addition, the algorithm in this paper detects outliers by clustering. It can be seen that the algorithm will send the updated cluster characteristic tuple instead of raw stream data during the time cycle ΔT of similar data preprocess on sensor nodes, when the multiple data objects collected by a sensor are in a known cluster radius. Therefore, it can greatly reduce the energy consumption of communication.

On cluster heads, similar flocking model keeps the nature of Vicsek model, such as simplicity and fast convergence. The clustering algorithm based on similar flocking model can cluster similar objects in a self-organization way as soon as possible. It can also cluster data distributed with any shape at one time rather than traditional two-phase framework. To a certain extent, this algorithm can reduce the computational energy consumption.

V. EXPERIMENTAL RESULTS AND EVALUATION

a. Experimental environment

Several experiments were performed to evaluate the performance of CASFM. Three algorithms were compared, i.e., CASFM, SUBFCM [6] and the anomaly detection algorithm by clustering ellipsoids [7], which was called as ADCE for simplicity. The experimental data used in this paper came from the laboratory of Berkeley University, which was collected by 54 sensors deployed in the Intel Berkeley Research lab from 28 February to April 5 in 2004 [18]. The experimental data included the indoor information of humidity, temperature, light intensity, voltage and topology information, etc. Each sensor produced a reading every second. Experiments took the values of temperature and humidity as input data, and the topological information was used to establish spatial neighbor relationship. The time cycle of processing data on cluster head node ΔT=1min. The algorithm was implemented on Matlab R2009a, and the machine was configured with Intel dual-core 2.0GHzCPU, 3G RAM, Windows XP Operating System.

In our experiments, 8 sensors formed one group according to their successive number to simulate hierarchical topology, therefore, 7 groups were produced and there are 6 sensors in the last group. Each group chose a sensor as cluster head dynamically according to the remaining energy. A virtual sensor was added as the sink node, whose number is 0.

The energy consumption parameters were set according to SUBFCM, E_{elec} =50nJ/bit, ξ_{fs} = 100pJ/bit/m2. Other parameters were selected by large number of experiments as follows: the initial velocity v = 1000, ε = 0.24, ζ = 10, iteration number M = 100.

During the initialization of the algorithm, the data initial position in a two-dimensional space that data point mapped to is based on the sensor's position which the data point belongs to. The position of sensors is defined by two attributes, x and y, which are the coordinates in the plane. The locations of sensors have been specified in experimental data set.

b. Experimental Results and Evaluation

The experiments evaluated the algorithm from detection rate, false alarm rate and energy consumption. As SUBFCM is only a stream clustering algorithm, we distinguished outlier using definition 3 in this paper to compare the effects of outlier detection.

b.i Detection rate and false alarm rate

Table 1 shows the detection rate and false alarm rate of three algorithms. It can be seen that the detection rates of three algorithms are nearly 100%. But CASFM has less false alarm rate than other two algorithms. ADCE focuses on the local detecting results of sensors and ignores the global characteristic of sensor network. Because SUBFCM depends on the parameters setting of clustering, it can't guarantee low false alarm rate. In CASFM, few parameters need to be set in advance. Moreover, each data point is autonomous, and through the self-organization of similar flocking model, data move and cluster by themselves. So the false alarm rate is low. The results show that CASFM is a better algorithm for outlier detection.

Table 1: Comparison of the detection rate and false alarm rate

Algorithm	Detection Rate	False Alarm Rate
ACDE	99%	1.2%
SUBFCM	100%	0.88%
CASFM	100%	0.74%

b.ii Energy consumption

On the basis of original data set, the experiments are performed to evaluate the energy consumption by changing the distance between sensors and adding some sensors, whose data are part of original data. Figure 3 shows the total energy consumption of all sensors. It can be seen that when the distance is no longer than 40 meters, there is little difference about the energy consumption of three algorithms. When the distance is getting longer gradually, the energy consumption of CASFM is lower than ADCE and SUBFCM and the gap is increasing. This is owing to the similar data processing on sending sensors. When the environment is normal, the data collected by sensors are very similar. Large amount of raw data do not need to be sent, which cut down the communication overhead and energy consumption greatly.

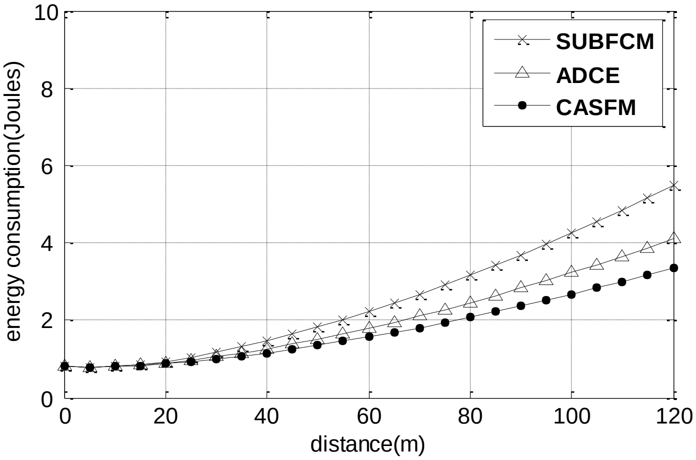

Figure 3. Comparison of energy consumption while changing the distance between sensors

Figure 4 shows the influence on energy consumption when the number of nodes increases. The added sensors are the same as original deployed sensors, for we duplicate some data of different time from original data set as the observations of added sensors. The distance between every two sensors are set to 100m, other parameters are not changed. It can be seen that when the number of nodes increases, CASFM consumes the lowest energy in three algorithms. It is due to the self-organization of similar flocking model and similar data processing on sensors. SUBFCM has the highest energy consumption when there are more sensors. The reason is that the information exchange between sensor and cluster head is more frequent during clustering when the number of sensors rises. ADCE transmits data locally, so its energy consumption is between that of SUBFCM and CASFM. The results show that CASFM can save more energy than the other two algorithms, especially in large-scale WSN.

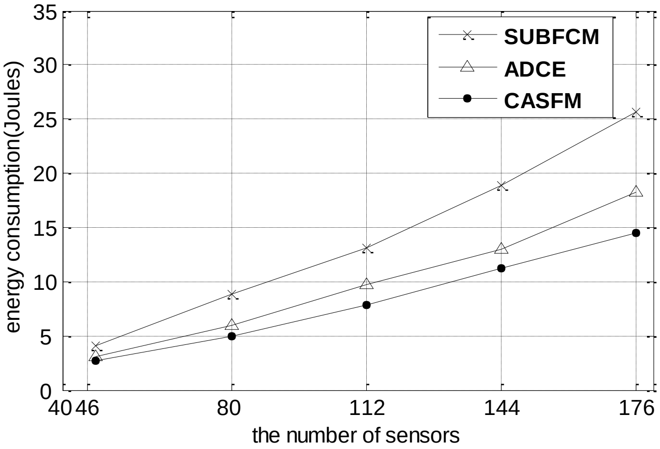

Figure 4. Comparison of energy consumption for different number of sensors

VI. CONCLUSIONS

In this paper, similar flocking model and a stream clustering algorithm based on similar flocking model (CASFM) are proposed to detect outliers in WSN. Through the self-organization property of the similar flocking model, CASFM can cluster data with any shape quickly and reduce the algorithm complexity. Considering the limited resources of wireless sensors, sensors only send necessary data according to the cluster characteristic information fed back from cluster head or sink node to decrease the transmission of similar data significantly. So the energy consumption can be decreased greatly. The experimental results show the effectiveness of CASFM in outlier detection and energy saving.

The algorithm proposed in this paper only considers the case that sensors are in a homogeneous network and there is no loss of data. When the sensor nodes are in a heterogeneous or unstable environment, how to effectively detect outliers is one of our future works. In addition, we are investigating other detection techniques and their applications.

ACKNOWLEDGMENTS

This work was partially supported by Swedish Research Links [No.348-2008-6212]; Research and Innovation Projects for Graduates of Jiangsu Province [CXZZ12_0483, CXLX12_0481]; the National Natural Science Foundation of China [61071093]; China’s Project 863

[2010AA701202], SRF for ROCS, SEM [NJ209002], and the Priority Academic Program Development of Jiangsu Higher Education Institutions [yx002001].

REFERENCES

[1] Xie Yingxin, Chen Xiangguang, Yu Xiangming et al. “Fast SVDD-based outlier detection approach in wireless sensor networks”, Journal of Scientific Instrument, Vol. 32, No. 1, pp 46-51, January 2011.

[2] Zhang, Y., N. Meratnia, and P. Havinga, “Outlier Detection Techniques for Wireless Sensor Networks: A Survey”. IEEE Communications Surveys and Tutorials, Vol. 12, No. 2, pp. 159-170, February 2010.

[3] Boyinbode Olutayo, Le Hanh, Mbogho Audrey, et al. “A survey on clustering algorithms for wireless sensor networks”. Proc. of 13th International Conference on Network-Based Information Systems, NBiS 2010, pp. 358-364, Japan, September 14-16, 2010.

[4] Korjani Mohammad Mehdi, Afshar Ahmad, Menhaj Mohammad Bagher, et al., “Optimal model detection in distributed sensor networks using genetic-fuzzy clustering”. Proc. of 2007 IEEE Congress on Evolutionary Computation, pp. 2817-2821, Singapore, September 25-28, 2007.

[5] Amirhosein T,Reza M,Frank E. “A Communication-Efficient Distributed Clustering Algorithm for Sensor Networks”. Proc. of 22nd International Conference on Advanced Information Networking and Applications – Workshops, AINAW 2008, pp.634-638, Japan, March 25-28, 2008.

[6] Hakilo S, Adnan A, Hamid G. “Distributed WSN Data Stream Mining based on Fuzzy Clustering”. Proc. of Symposia and Workshops on Ubiquitous,Autonomic and Trusted Computing, 2009, UIC-ATC '09.pp.395-400, Brisbane, Australia, July 7-9, 2009.

[7] Masud M, Sutharshan R, Christopher L, et al. “Anomaly Detection by Clustering Ellipsoids in Wireless Sensor Networks”. Proc. of 2009 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp.331-336, Melbourne,Australian, December 7-10, 2009.

[8] S. Rajasegarar, C. Leckie, M. Palaniswami and J.C. Bezdek. “Distributed Anomaly Detection in Wireless Sensor Networks”. Proc. of 10th IEEE Singapore International Conference on Communication systems, 2006. ICCS 2006, pp.1-5, Singapore, October 3-5 , 2006.

[9] Hao Jutao, Chen Qingkui, Huo Huan, et al. “Energy Efficient Clustering Algorithm for Data Gathering in Wireless Sensor Networks”. Journal of Networks, Vol.6, No.3, pp. 490-497, March 2011.

[10] Katiyar Vivek, Chand Narottam and Soni Surender. “Energy-efficient multilevel clustering in heterogeneous wireless sensor networks”. Communications in Computer and Information Science, Vol. 125, pp. 293-299, 2011.

[11] Reynolds C. “Flocks, herds, and schools: A distributed behavioral model. Comput Graph”, ACM SIGGRAPH Computer Graphics, Vol. 21, No.4, pp. 25-34, 1987.

[12] Vicsek T, Czirok A, Ben-Jacob E, et al. “Novel type of phase transition in a system of self-driven particles”. Phys. Rev. Lett., Vol. 75, No.6, pp. 1226-1229, August 1995.

[13] Li Qiang, He Yan and Jiang Jing ping. “A New Clustering Algorithm Based on Flocking with Virtual Leaders”. Journal of Electronics& Information Technology, Vol. 31, No. 8, pp. 1846-1851, August 2009.

[14] Xiaohui Cui, Jinzhu Gao and Thomas E. Potok. “A flocking based algorithm for document clustering analysis”. Journal of Systems Architecture, Vol. 52, No. 8, pp. 505-515, August 2006.

[15] TAN Pang-ning, STEINBACH M and KUMAR V, “Introduction to data mining”. Boston：Pearson Addison Wesley Education Inc, 2006.

[16] M.Hassani,E. Muller and T. Seidl. “EDISKCO:Energy Efficient Distributed In-Sensor-Network K-center Clustering with Outliers”. Proc. of the Third International Workshop on Knowledge Discovery from Sensor Data, SensorKDD’09, pp. 39-48, Paris,France, June 28**,** 2009.

[17] Heinzelman W R, Chandrakasan A and Balakrishnan H. “Energy efficient communication protocol for wireless microsensor networks”. Proc. of the 33rd Annual Hawaii International Conference on System Sciences. pp. 8020-8029, Maui, Hawaii, January 4-7, 2000.

[18] http://db.csail.mit.edu/labdata/labdata.html