JAIDM
Journal of AI and Data Mining
Vol. 1, No. 1, 2013, 1–11.

Outlier detection in wireless sensor networks using distributed principal component analysis

A. Ahmadi Livani, M. Abadi*, M. Alikhani

Faculty of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran

Received 08 January 2013; accepted 28 January 2013
* *Corresponding author: abadi@modares.ac.ir (M. Abadi)*

Abstract
Outlier detection is an important task for intrusion detection and fault diagnosis in wireless sensor networks (WSNs). Outliers in sensed data may be caused due to compromised or malfunctioning sensor nodes. In this paper, we propose a centralized and a distributed approach based on the principal component analysis (PCA) for outlier detection in WSNs. In the distributed approach, we partition the network into multiple groups of sensor nodes. Each group has a group head and several member nodes. Every member node uses a fixed-width clustering algorithm and sends a description of its local sensed data to the group head. The group head then applies a distributed PCA to establish a global normal pattern and detect outliers. This pattern is periodically updated using weighted coefficients. We compare the performance of the centralized and distributed approaches based on the real sensed data collected by 54 Mica2Dot sensors deployed in Intel Berkeley Research Lab. The experimental results show that the distributed approach reduces both communication overhead and energy consumption, while achieving comparable accuracy.

Keywords: *Wireless sensor network; Outlier detection; Principal component analysis.*

1. Introduction
Wireless sensor networks (WSNs) are composed of a large number of tiny sensor nodes deployed in an environment for monitoring and tracking purposes. Sensor nodes use ad-hoc communications and collaborate with each other to sense different phenomena that may vary in time and space, and send the sensed data to a central node for further processing and analysis [1]. WSNs are applied to various applications, ranging from military to civilian fields.

The data sensed and collected by sensor nodes are often unreliable. The quality of the sensed data may be affected by noise or missing values. The low cost and low quality sensor nodes have limitations in power supply, memory, computational capabilities, and communication bandwidth [1]. These limitations make the sensed data unreliable and inaccurate. Particularly, when power supply is exhausted, the probability of generating erroneous data will grow rapidly [2]. On the other hand, the operations of sensor nodes are frequently susceptible to environmental effects. The vision of large scale and high density WSNs is to randomly deploy a large number of sensor nodes in harsh and unattended environments [3]. Since events occurred in the real world (*e.g.*, forest fire or earthquake) cannot be accurately detected using erroneous data, it is extremely important to ensure the reliability and accuracy of the sensed data [4], [5].

An outlier is an observation (or a set of observations) in a data set, which appears to be inconsistent with the remainder of that data set [6]. The term outlier, also known as anomaly, originally stems from the field of statistics [7]. Outlier detection, also known as anomaly detection, is one of the fundamental tasks of data mining along with predictive modeling, cluster analysis, and association analysis [8].

In WSNs, outliers can be defined as those data that have significant deviations from the normal

pattern of the sensed data [9]. Potential sources of outliers include noise, actual events, or malicious attacks [8]. A straightforward approach for outlier detection in WSNs is to establish a normal pattern of the sensed data and detect data that deviate significantly from the established pattern as outliers. As environmental conditions may change over time, a predefined normal pattern will not be sufficiently representative for future outlier detection. Thus, a key challenge here is to dynamically detect outliers with acceptable accuracy while minimizing communication overhead and energy consumption.

In WSNs, the energy consumption in the radio communication is significantly greater than of that in the computation [10]-[12]. For example, in Sensoria sensors and Berkeley motes, the ratio between communication and computation energy consumption ranges from 10^3 to 10^4 [13]. Hence, we can take this advantage to prolong the network lifetime through increasing computational cost in order to reduce communication overheads.

Principal component analysis (PCA) is a powerful technique for analyzing and identifying patterns in data [14]. It finds the most important axis to express the scattering of data [15]. By using PCA, the first principal component is calculated, which reflects the approximate distribution of data.

In this paper, we propose a centralized and a distributed PCA-based approach for outlier detection in WSNs. We partition the network into groups of sensor nodes. Each group has a group head and several member nodes. In the centralized approach, every member node sends its local sensed data to the group head. The group head then applies PCA to establish a global normal pattern and detect outliers. In the distributed approach rather than sending all sensed data, every member node uses a fixed-width clustering (FWC) algorithm and sends a description of its local sensed data to the group head. The group head then applies a distributed PCA (DPCA) to establish the global normal pattern. In these two approaches, the established normal pattern is periodically updated using weighted coefficients. We compare the performance of the centralized and distributed approaches based on real sensed data collected from 54 Mica2Dot sensors deployed in Intel Berkeley Research Lab. In comparison to the centralized approach, we show that the distributed approach can achieve significant reductions in communication overhead and energy consumption, while achieving comparable accuracy.

The rest of this paper is organized as follows: Section 2 briefly reviews some related work. Section 3 formally introduces the problem of outlier detection in WSNs. Sections 4 and 5 describe the centralized and distributed outlier detection approaches, respectively, and Section 6 analyzes the communication overhead and computational cost of them. Section 7 reports the experimental results and finally Section 8 draws some conclusions.

2. Related work

In monitoring WSNs, due to the critical nature of applications in many cases, sensed data collected from various sensor nodes should be analyzed dynamically and compared to an established normal pattern in order to detect potential outliers.

Janakiram *et al.* [16] proposed a technique based on Bayesian belief networks (BBNs) for outlier detection in the sensed data. The technique uses BBNs to capture the spatio-temporal correlations among the observations of sensor nodes and the conditional dependencies among the observations of sensor features. Each node trains a BBN to detect outliers based on its neighbors' sensed data as well as its own sensed data. An observation is considered as outlier if it falls beyond the range of the expected class. Accuracy of a BBN depends on how the conditional dependencies among the observations of sensor features exist. This technique does not work well when the resources are limited and the network topology changes dynamically.

Rajasegarar *et al.* [17], [18] proposed two distributed outlier detection approaches. The first approach is based on clustering. In this approach, sensor nodes have a hierarchal topology. At the end of each time window, every sensor node clusters its sensed data and sends the statistics of the clusters to its immediate parent node. The parent node then merges its own clusters with the clusters collected from its intermediate children nodes and sends the statistics of the merged clusters to its immediate parent node. This process continues recursively up to the gateway node, where an outlier detection algorithm is applied on its merged clusters to detect outlier clusters. An outlier cluster can be determined in the gateway node, if the cluster's average inter-cluster distance is larger than one threshold value of the set of inter-cluster distances. Determining the parameter used to compute the average inter-cluster distance is not always easy. The second approach is based on one-class quarter sphere SVM. Every sensor node runs the one-class quarter-sphere SVM on its sensed data and sends its local radius to its parent node. The parent node then combines its own local radius with radii collected from its children

nodes and sends the global radius to its children nodes. The children nodes use the global radius to locally detect outliers. The sensed data that lies outside the global quarter sphere are considered as outliers. This approach ignores spatial correlations of neighboring sensor nodes, which makes the results of local outliers inaccurate.

Chatzigiannakis *et al.* [5] proposed a centralized outlier detection approach in which PCA is applied on the sensed data of all sensor nodes in order to reduce the dimensionality of them. The first few most important derived principal components are then selected to be used in the subspace method. The goal of this method is to divide the current sensed data into normal and anomalous spaces. However, this approach has several drawbacks. It uses squared prediction error (SPE) to perform outlier detection in the residual space. Since SPE is sensitive to modeling errors, it may increase the false alarm rate. Also, sending all sensed data to a central node leads to a high communication overhead, which is a major source of energy consumption for sensor nodes.

Sheng *et al.* [19] proposed a histogram-based technique to detect global outliers over the sensed data. This technique attempts to reduce communication overhead by collecting hints in the form of a histogram rather than collecting all sensed data in a central node. The central node uses the hints to extract the data distribution in the network and detect the potential outliers. However, this technique does not consider the inter-feature dependencies of multi-dimensional sensed data.

Ahmadi Livani *et al.* [14] proposed an energy-efficient approach for detecting outliers in the sensed data. The outlier detection procedure is comprised of two phases: *training* and *outlier detection*. In the training phase, every sensor node computes a description of its local sensed data and sends it to its group head. After receiving descriptive data from all member nodes, the group head applies the approximate global PCA (AGPCA) to establish a global normal pattern and sends it to all member nodes. In the outlier detection phase, every member node detects outliers based on their projection distances from the global first principal component.

3. Problem definition

We consider a WSN composed of a set of sensor nodes deployed in a homogenous environment. The sensor nodes are synchronized and their sensed data belong to the same unknown distribution. We partition the network into multiple groups of sensor nodes. Each group has a group head and several member nodes. The sensor nodes within the same group are physically close to each other and sense a similar phenomenon. The partitioning can be static or dynamic [20]. In the dynamic partitioning, the network may be rearranged periodically, if the environmental conditions change.

Let $G = \{s_i : i = 1 \dots s\}$ be a group of sensor nodes. At each time interval Δt_k, every member node $s_i \in G$ senses a data vector x_k^i. Each data vector is composed of features x_{kj}^i:

$$x_k^i = (x_{k1}^i, x_{k2}^i, \cdots, x_{kd}^i), \quad x_k^i \in \Re^d \ . \tag{1}$$

During each time window t, s_i senses a set of data vectors $X_i(t) = \{x_k^i(t) : k = 1 \dots n_i\}$. An outlier in a set of data vectors is defined as a data vector that has significant deviation from the other data vectors. Our aim is to detect outliers in data vectors sensed by the member nodes.

4. Centralized outlier detection approach

In this section, we propose a centralized approach, for outlier detection in WSNs. It consists of three phases: *training*, *outlier detection*, and *updating*.

4.1. Training phase

The training phase involves modeling the distribution of a given set of normal data vectors. Let G be a group of sensor nodes. In this approach, every member node $s_i \in G$ sends its sensed data vectors to the group head s_G. After receiving the data vectors from all member nodes, s_G combines its own data vectors with them and forms a set of data vectors $X(0)$:

$$X(0) = \begin{bmatrix} X_1(0) \\ X_2(0) \\ \vdots \\ X_s(0) \end{bmatrix} , \tag{2}$$

where $X_i(0)$ is an $n_i \times d$ matrix of data vectors sensed by the member node s_i, $i = 1 \dots s$. So, $X(0)$ is an $n \times d$ matrix, whose rows are the data vectors and columns are the features.

$$n = \sum_{i=1}^{s} n_i \ . \tag{3}$$

s_G first normalizes the matrix $X(0)$ to a range of [0,1]. It then computes the global column means $\bar{x}(0)$ of $X(0)$ and the global covariance matrix $S(0)$ of $X(0)$:

$$S(0) = \frac{1}{n} X^T(0)(I - \frac{1}{n} e_n e_n^T) X(0) \ , \tag{4}$$

where $e_n \equiv (1,1,\cdots,1)^T$ is a vector of length n.

To establish a normal pattern, s_G computes the global first principal component $\varphi(0)$. The principal components of $X(0)$ are given by a singular value decomposition (SVD) [21] of $nS(0)$:

$$nS(0) = V(0)\Sigma^2(0)V^T(0) \ , \tag{5}$$

where $V(0)$ is the matrix of principal components of $X(0)$ and $\Sigma^2(0) = \text{diag}(\lambda_1^2(0), \lambda_2^2(0), \cdots, \lambda_d^2(0))$ is the diagonal matrix of eigenvalues ordered from largest to smallest. Note that often $n-1$ is used instead of n in the above equations when the data are a sample from some larger population.
After that, as shown in Figure 1, s_G calculates the projection distance of each data vector $x_k^i(0) \in X(0)$ from $\varphi(0)$ as

$$d_p(x_k^i(0), \varphi(0)) = (\left\| x_k^i(0) - \bar{x}(0) \right\|^2 - (\varphi^T(0) \cdot (x_k^i(0) - \bar{x}(0)))^2)^{\frac{1}{2}} \ . \tag{6}$$

The maximum projection distance of all data vectors from $\varphi(0)$ is then calculated as

$$d_{\max} = \max_{1 \le k \le n} d_p(x_k^i(0), \varphi(0)) \tag{7}$$

and the triple $(\bar{x}(0), \varphi(0), d_{\max})$ is used to establish the global normal pattern $P(0)$.

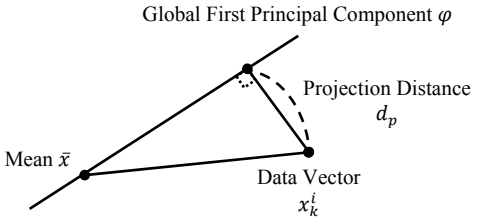


Figure 1. The projection distance of a data vector x_k^i from the global first principal component φ.

4.2. Outlier detection phase

To detect outlier data vectors, during each time window t, the group head s_G first calculates the projection distance of each data vector $x_k^i(t) \in X(t)$ from $\varphi(t-1) \in P(t-1)$. It then classifies $x_k^i(t)$ as outlier, if the calculated projection distance is greater than $d_{\max}$:

$$\begin{cases} d_p(x_k^i(t), \varphi(t-1)) > d_{\max} & : \text{Outlier} \\ d_p(x_k^i(t), \varphi(t-1)) \le d_{\max} & : \text{Normal} \end{cases} . \tag{8}$$

4.3. Updating phase

There might be changes over time in the conditions of the environment in which a WSN is deployed. Therefore, it is necessary to update the global normal pattern.

Let t be the current time window. To update the global normal pattern $P(t)$, the group head s_G first calculates the global column means and global covariance matrix of normal data vectors at the ρ previous time windows (see Figure 2):

$$\bar{x}_\rho(t) = \sum_{\tau=t-\rho+1}^{t} w(\tau)\bar{x}(\tau) \ , \tag{9}$$

$$S_\rho(t) = \sum_{\tau=t-\rho+1}^{t} w(\tau)S(\tau) \ , \tag{10}$$

where $\bar{x}(\tau)$ and $S(\tau)$ are the global column means and global covariance matrix at time window τ, respectively. $w(\tau)$ is a weighted coefficient assigned to the normal data vectors at time window τ. The Ebbinghaus' forgetting curve [15] is used to calculate the weighted coefficients. As shown in Figure 3, the purpose of using the forgetting curve is to reduce the importance of normal data vectors in the old time windows when updating the global normal pattern.
s_G then computes the global first principal component $\varphi(t)$ by a singular value decomposition of $S_\rho(t)$ and uses the triple $(\bar{x}_\rho(t), \varphi(t), d_{\max})$ to update the global normal pattern $P(t)$.

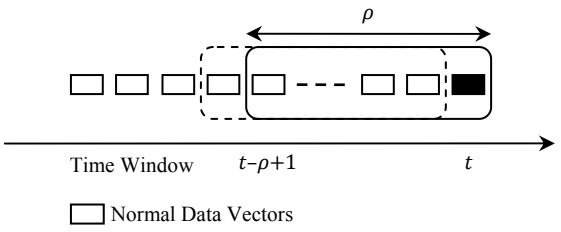


Figure 2. Updating the global normal pattern.

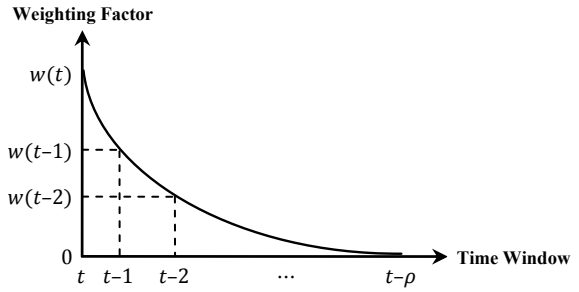


Figure 3. The Ebbinghaus' forgetting curve.

The centralized outlier detection approach has some major drawbacks. First, a large volume of data vectors should be transmitted over the network, which leads to significant decrease of the network lifetime. Second, a high communication load is imposed on the group head because all other member nodes send their data to it.

In the following section, we propose a distributed outlier detection approach employing in-network processing and sensor collaborations to prolong the network lifetime.

5. Distributed outlier detection approach

In this section, we propose a distributed approach for outlier detection in WSNs. The approach consists of three phases: *training*, *outlier detection*, and *updating*.

In the training phase, the distribution of normal data vectors is modeled. Let G be a group of sensor nodes. Every member node $s_i \in G$ first normalizes its local data vectors $X_i(0)$. It then applies the description procedure on $X_i(0)$ and sends the obtained descriptive data $D_i(0)$ to its group head s_G. Afterwards, s_G applies a distributed PCA (DPCA) [22] on the set of descriptive data $\mathcal{D}(0)$ received from all member nodes and then establishes a global normal pattern $P(0)$. Figure 4 shows the pseudo-code of the training phase.

In the outlier detection phase, during each time window t, s_G applies the detection procedure on the set of descriptive data $\mathcal{D}(t)$ received from all member nodes and detects outliers based upon the global normal pattern $P(t-1)$. Figure 5 shows the pseudo-code of the outlier detection phase.

In the updating phase, at the end of each time window t, s_G first applies DPCA on the set of normal descriptive data $\mathcal{D}^*(t) \subseteq \mathcal{D}(t)$ and computes the global covariance matrix $S(t)$. It then updates the global normal pattern $P(t)$ through calculating the global column means and global covariance matrix of normal data vectors at the ρ previous time windows.

Better load balancing is achieved by distributing the outlier detection process among all member nodes. Also, the communication overhead is reduced by sending the description of data vectors rather than the whole sensed data vectors. This helps to prolong the network lifetime.

5.1. Data normalization

Features of data vectors sensed by member nodes may have different ranges. Hence, when calculating the distance between data vectors, features with larger values will dominate those features with smaller values. Therefore, we normalize each feature into a range $[0,1]$ in order to ensure that all features have the same influence on the distance calculation.

In the centralized approach, all data vectors from all member nodes are sent and available at the group head. Therefore, the group head can normalize all data vectors. In the distributed approach, if every member node normalizes its local data vectors by using the local minimum and local maximum parameters, the resulting normalized data vectors will not be exactly the same as those of the centralized approach. Hence, we perform the following operations to find the global minimum and global maximum parameters in order to normalize the local data vectors.

After each time window t, every member node $s_i \in G$ computes two vectors $x^i_{\min}(t)$ and $x^i_{\max}(t)$ of minimum and maximum values for its local data vectors $X_i(t)$ and sends them to the group head s_G. After receiving above vectors from member nodes, s_G computes the global minimum and global maximum vectors $x^g_{\min}(t+1)$ and $x^g_{\max}(t+1)$, and sends them to all member nodes. Every member node s_i uses these global parameters to normalize its local data vectors.

procedure Training

input:
  A group of sensor nodes $G = \{s_i : i = 1 \dots s\}$
output:
  A global normal pattern $P(0)$
begin
  for all member nodes $s_i \in G$ **do**
    Sense a set of data vectors $X_i(0)$ and normalize it
    Apply the description procedure on $X_i(0)$ and send the descriptive data $D_i(0)$ to the group head s_G
  end for
  for group head s_G **do**
    $\mathcal{D}(0) \coloneqq \bigcup_{i=1}^{s} \{D_i(0)\}$
    Apply DPCA on the set of descriptive data $\mathcal{D}(0)$ and establish a global normal pattern $P(0)$
  end for
end procedure

Figure 4. The training phase.

procedure Outlier Detection

input:
  A group of sensor nodes $G = \{s_i : i = 1 \dots s\}$
  The global normal pattern $P(t-1)$
begin
  for each time window t **do**
    for all member nodes $s_i \in G$ **do**
      Sense a set of data vectors $X_i(t)$ and normalize it
      Apply the description procedure on $X_i(t)$ and send the descriptive data $D_i(t)$ to the group head s_G
    end for
    for group head s_G **do**
      Detect outliers based upon $P(t-1)$
    end for
  end for
end procedure

Figure 5. The outlier detection phase.

5.2. Data description

Every member node $s_i \in G$, at each time window t, computes a so-called descriptive data $D_i(t)$ of its normalized data vectors $X_i(t)$ and sends it to the group head s_G. $D_i(t)$ is represented as a ternary $D_i(t) = (\bar{x}_i(t), R_i(t), C_i(t))$, where $\bar{x}_i(t)$ is the column means of $X_i(t)$, $R_i(t)$ is the matrix obtained by the QR decomposition [22] of the column-centered matrix of $X_i(t)$, and $C_i(t)$ is the description of clusters formed by the clustering operation on $X_i(t)$. Our clustering algorithm is based on the fixed-width clustering (FWC) algorithm [23].

Figure 6 shows the pseudo-code of the algorithm FWC that takes $X_i(t)$ as input and groups its data vectors into a set of clusters $C_i(t)$ of fixed radius w_c. For each data vector $x_k^i(t) \in X_i(t)$, if $C_i(t)$ is empty, a new cluster $C_1^i(t)$ is created with $x_k^i(t)$ as its centroid. Otherwise if the distance between $x_k^i(t)$ and the centroid of $C_j^i(t)$ is less than or equal to w_c, $x_k^i(t)$ is added to the nearest cluster $C_{\min}^i(t)$ and the centroid of $C_{\min}^i(t)$ is adjusted to the mean of the data vectors it contains. Otherwise, a new cluster $C_j^i(t)$ is created with $x_k^i(t)$ as its centroid. This operation forms a set of disjoint clusters $C_i(t)$. Finally, the radius of each cluster $C_j^i(t) \in C_i(t)$ is set to the outermost data vector in the cluster.

5.3. Establishing global normal pattern

Let $\mathcal{D}(0)$ be the set of descriptive data received by the group head s_G.

$$\mathcal{D}(0) = \bigcup_{i=1}^{s} \{D_i(0)\} \ , \tag{11}$$

where $D_i(0) = (\bar{x}_i(0), R_i(0), C_i(0))$ is the descriptive data of the member node s_i.

In order to establish a global normal pattern, s_G first applies DPCA on $\mathcal{D}(0)$ to compute the global first principal component $\varphi(0)$. For this purpose, s_G first computes the global column means of $X(0)$:

$$\bar{x}(0) = \frac{1}{n}\sum_{i=1}^{s} n_i \bar{x}_i(0) \tag{12}$$

and then computes the QR decomposition of each pair of matrices $R_i(0)$ and $R_j(0)$ received from member nodes by using Givens rotations:

$$\begin{bmatrix} R_i(0) \\ R_j(0) \end{bmatrix} = Q_{(i,j)}(0) R_{(i,j)}(0) \ . \tag{13}$$

Next, s_G continues this operation until $\ell = |\log_2^s|$ steps to obtain $R_{(1,2,\dots,s)}(0)$ and computes the QR decomposition of the following upper-trapezoidal $(s+d) \times d$ matrix:

$$\begin{bmatrix} \sqrt{n_1}(\bar{x}_1(0) - \bar{x}(0)) \\ \sqrt{n_2}(\bar{x}_2(0) - \bar{x}(0)) \\ \vdots \\ \sqrt{n_s}(\bar{x}_s(0) - \bar{x}(0)) \\ R_{(1,2,\dots,s)}(0) \end{bmatrix} = Q(0)R(0) \ . \tag{14}$$

Next, s_G computes the global first principal component $\varphi(0)$ of $X(0)$ by a singular value decomposition of $R(0)$:

$$R(0) = U(0)\Sigma(0)V^T(0) \ . \tag{15}$$

Notice that the computed global principal components are exactly the same as those computed from the centralized approach.

We can easily calculate the global covariance matrix $S(0)$ as

$$S(0) = \frac{1}{n} R^T(0)R(0) \ , \tag{16}$$

procedure FWC

input:
  A set of data vectors $X_i(t) = \{x_k^i(t) : k = 1 \dots n_i\}$
  Cluster radius w_c
output:
  A set of clusters $C_i(t) = \{C_j^i(t) : j = 1 \dots l_i\}$
begin
  $C_i(t) \coloneqq \emptyset$
  for each data vector $x_k^i(t) \in X_i(t)$ **do**
    if $C_i(t) = \emptyset$ **then**
      Create a new cluster $C_1^i(t)$ with centroid $x_k^i(t)$ and radius w_c
      $C_i(t) \coloneqq \{C_1^i(t)\}$
    else
      Find the nearest cluster $C_{\min}^i(t) \in C_i(t)$ to $x_k^i(t)$
      if $d(x_k^i(t), C_{\min}^i(t)) \le w_c$ **then**
        Add $x_k^i(t)$ to $C_{\min}^i(t)$ and update its centroid
      else
        Create a new cluster $C_j^i(t)$ with centroid $x_k^i(t)$ and radius w_c
        $C_i(t) \coloneqq C_i(t) \cup \{C_j^i(t)\}$
      end if
    end if
  end for
  for each cluster $C_j^i(t) \in C_i(t)$ **do**
    Find the outermost data vector $x_k^i(t)$ in cluster $C_j^i(t)$
    Set the radius of cluster $C_j^i(t)$ to $d(x_k^i(t), C_j^i(t))$
  end for
end procedure

Figure 6. The FWC algorithm.

Finally, s_G calculates the distance of each cluster $C_j^i(0) \in \mathcal{D}(0)$ from $\varphi(0)$:

$$d(C_j^i(0), \varphi(0)) = d_p(c_j^i(0), \varphi(0)) + r_j^i(0) \quad , (17)$$

where $c_j^i(0)$ and $r_j^i(0)$ are the centroid and radius of $C_j^i(0)$, respectively. $d_p(c_j^i(0), \varphi(0))$ is the projection distance from $c_j^i(0)$ to $\varphi(0)$.
The triple $(\bar{x}(0), \varphi(0), d_{\max})$ is then used to establish the global normal pattern $P(0)$, where $d_{\max}$ is the maximum distance of all clusters in $\mathcal{D}(0)$ from $\varphi(0)$.

$$d_{\max} = \max_{1 \le i \le s, 1 \le j \le l_i} d(C_j^i(0), \varphi(0)) \ . \quad (18)$$

It should be mentioned that $d_{\max}$ is used in the outlier detection phase to detect data vectors that have significant deviation from the global normal pattern.

5.4. Outlier detection

Let $\mathcal{D}(t)$ be the set of descriptive data received by the group head s_G at time window t and $C_i(t) \in \mathcal{D}(t)$ be the description of clusters of the member node s_i. In order to detect outliers, s_G first calculates the distance of each cluster $C_j^i(t) \in C_i(t)$ from $\varphi(t-1) \in P(t-1)$:

$$d(C_j^i(t), \varphi(t-1)) = d_p(c_j^i(t), \varphi(t-1)) + r_j^i(t) \ . \quad (19)$$

It then classifies $C_j^i(t)$ as outlier, if the calculated distance is greater than $d_{\max}$:

$$\begin{cases} d(C_j^i(t), \varphi(t-1)) > d_{\max} & : \text{Outlier} \\ d(C_j^i(t), \varphi(t-1)) \le d_{\max} & : \text{Normal} \end{cases} . \quad (20)$$

If the number of outlier clusters received from a member node is greater than a threshold, the descriptive data received from that node will be discarded.

6. Complexity analysis

In this section, we analyze the communication overhead and computational cost of the centralized and distributed approaches in more detail.
In the centralized approach, at each time window, every member node s_i should communicate to the group head to send its local sensed data vectors. Hence, it incurs a communication overhead of $O(n_i d)$, where n_i is the number of data vectors sensed during the time window and d is the number of features of data vectors. Also, in order to establish or update the global normal pattern, first, the group head should calculate the global column means and global covariance matrix of normal data vectors for several previous time windows, which has a computational cost of $O(nd^2)$, where n is the number of received data vectors. Then, it should perform a singular value decomposition to compute the updated global first principal component, which has a computational cost of $O(d^3)$.
In the distributed approach, at each time window, in order to normalize the data vectors, every member node s_i should communicate to the group head to send a pair of vectors of minimum and maximum values for its local data vectors. The group head should communicate with all the member nodes to return to them the global minimum and maximum vectors. Also, in order to compute a description of normalized data vectors, every member node s_i should perform the QR decomposition and clustering operations. Hence, it incurs a communication overhead of $O(d^2)$ and a computational cost of $O({n_i}^2 d)$, where n_i is the number of data vectors sensed during the time window and d is the number of features of data vectors. Also, in order to establish or update the global normal pattern, first, the group head should apply DPCA on normal descriptive data and calculate the global column means and global covariance matrix of normal data vectors for several previous time windows, which has a computational cost of $O(d^3 \log_2 s)$, where s is the number of member nodes. Then, it should perform a singular value decomposition to compute the updated global first principal component, which has a computational cost of $O(d^3)$.
Table 1 shows the comparison between the communication overhead and computational costs of the centralized and distributed approaches.

Table 1. Comparing the centralized and distributed approaches for communication overhead and computational cost

	Computational Cost of the Group Head	Computational Cost of a Member Node	Communication Overhead of the Network
Centralized Approach	$O(nd^2)$	–	$O(nd)$
Distributed Approach	$O(d^3 \log_2 s)$	$O({n_i}^2 d)$	$O(sd^2)$

$(s \ll n, d \ll n, n_i \ll n)$

7. Experimental results

In this section, we compare the performance of the distributed outlier detection approach with that of the centralized approach.
We used the real sensed data collected from 54 Mica2Dot sensors deployed in Intel Berkeley Research Lab between February 28 and April 5, 2004. The sensed data included humidity, temper-

ature, light, and voltage values collected once in 31 seconds. In the experiments, we first partitioned the sensor network into eight groups of sensor nodes using the grouping algorithm in [20]. We then selected data from a group that included six nodes, namely nodes 37 to 42. We also randomly selected one of nodes and added some Gaussian noise to its sensed data to simulate the malfunctioning node. The amount of noise was measured by the signal-to-noise ratio (SNR). In the experiments, the length of time window was set to 52 minutes and the parameter SNR was set to 32 dB.

Cumulative percent variance (CPV) [24] is a measure of the percent variance captured by the first few principal components. It can be used to evaluate the importance of each principal component. Figure 7 shows the percent variance captured by the global first principal component $\varphi(t)$ for time windows 0 to 10, in the centralized and distributed approaches. As shown in the Figure 7, at each time window, the global first principal component captures at least 50 percent of the total variance of normal data vectors. Hence, we can use it to establish the global normal pattern at each time window.

Figure 8 compares the behavior of the malfunctioning node with a normal node during a time window of the outlier detection phase, in the centralized and distributed approaches. As can be seen in Figure 8, the malfunctioning node behaviors significantly different from the normal node and thus it can be easily detected by considering the projection distance.

We examined the effect of varying two parameters: The cluster radius w_c ranging from 0.01 to 0.90 and the signal-to-noise ratio parameter SNR ranging from 0 to 40 dB.

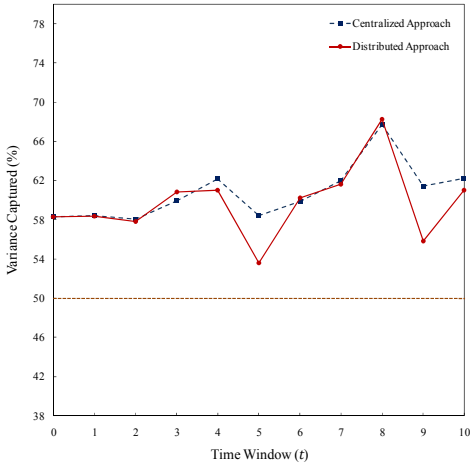
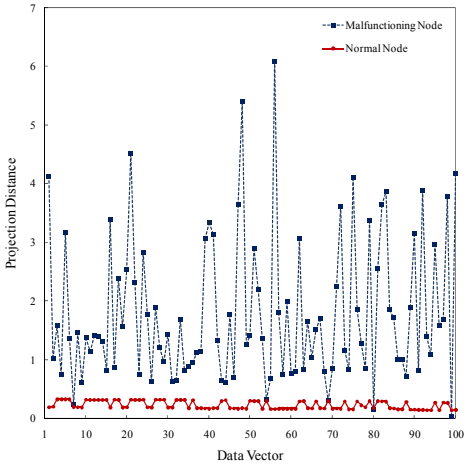
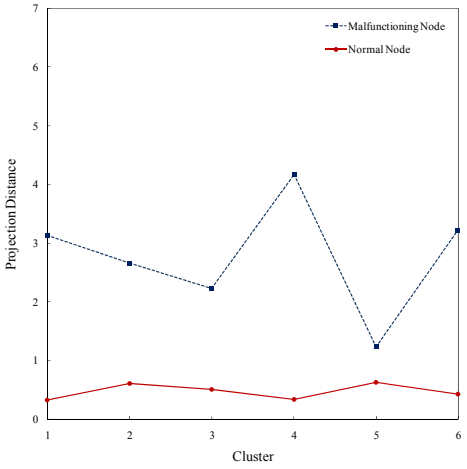


Figure 7. The percent variance captured by each principal component.



(a) Centralized approach



(b) Distributed approach

Figure 8. Projection distance from the global first principal component during a time window.

Table 2 compares the performance of the distributed approach for different values of w_c. For this sensed data, $w_c = 0.05$ is a better choice than other values by which the distributed approach can achieve a better trade-off between the detection rate (DR) and false alarm rate (FAR).

Table 2. Average detection and false alarm rates of the distributed approach for different values of w_c

w_c	Distributed Approach	
	Average DR	Average FAR
0.01	94.05	2.62
0.05	93.63	2.35
0.10	93.20	2.32
0.30	91.87	1.83
0.50	94.81	3.40
0.70	96.24	5.70
0.90	96.88	6.38

Table 3 compares the performance of the centralized and distributed approaches for different values of SNR. As can be seen in Table 3, the average detection and false alarm rates for the distributed approach are respectively 96.7% and 3.9%,

while for the centralized approach are respectively 96.4% and 2.7%. Hence, the distributed approach achieves a comparable performance to that of the centralized approach.

Table 3. Average detection and false alarm rates of the centralized and distributed approaches for different values of SNR

SNR	Centralized Approach		Distributed Approach	
	Average DR	Average FAR	Average DR	Average FAR
0	99.98	3.73	99.97	4.91
4	99.91	3.39	99.83	4.67
8	99.82	3.11	99.67	4.36
12	99.71	2.93	99.55	4.13
16	99.26	2.81	99.02	3.96
20	98.65	2.66	98.36	3.75
24	97.56	2.45	97.34	3.56
28	96.26	2.35	96.04	3.44
32	93.89	2.27	94.34	3.40
36	91.32	2.20	92.05	3.31
40	84.24	2.16	87.51	3.26

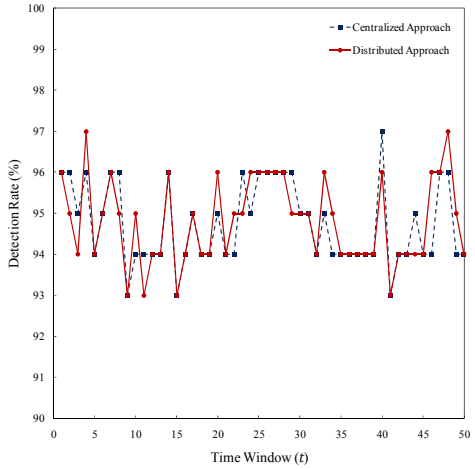


Figure 9. Detection rate of the centralized and distributed approaches during 50 time windows.

Figure 9 compares the detection rate of the centralized and distributed approaches during 50 time windows. As shown in Figure 9, the distributed approach is able to detect outlier data vectors with a rate similar to that of the centralized approach.

Table 4 compares performance of the centralized and distributed approaches for different lengths of time window, ΔT, ranging from 26 to 156 minutes. As shown in Table 4, the distributed approach is able to detect outlier data vectors with a rate similar to that of the centralized approach.

Figures 10 and 11 show reduction in communication overhead (RCO) [12] in the network for different values of w_c and ΔT, respectively. Reduction in communication overhead is calculated as

$$RCO = \frac{n - \tau}{n}, \quad (21)$$

where n and τ are the total number of data sent in the centralized and distributed approaches, respectively.

When compared to the centralized approach, the distributed approach achieves 68% to 95% reduction in communication overhead for w_c in the range of 0.01 to 0.90 and 92% to 96% reduction in communication overhead for ΔT in the range of 50 to 630 minutes.

Table 4. Average detection and false alarm rates of the centralized and distributed approaches for different values of ΔT

ΔT	Centralized Approach		Distributed Approach	
	Average DR	Average FAR	Average DR	Average FAR
26	90.92	1.65	91.43	2.22
52	93.89	2.27	94.34	3.40
78	94.48	4.70	94.53	4.89
104	94.00	5.90	94.77	5.48
130	94.46	5.96	95.49	5.94
156	95.15	6.01	95.50	6.05

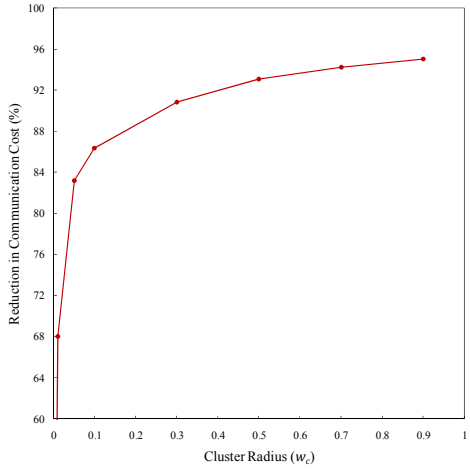


Figure 10. Reduction in communication overhead in the network for different values of w_c.

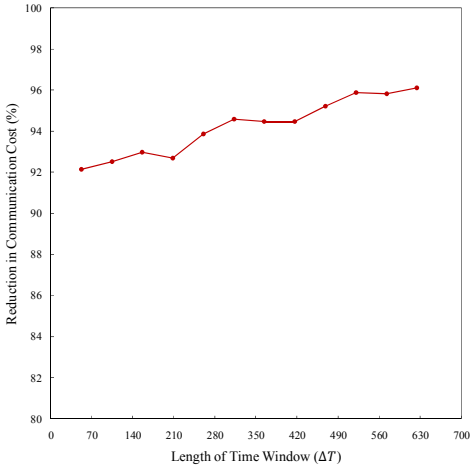


Figure 11. Reduction in communication overhead for different values of ΔT.

8. Conclusions

In this paper, we proposed a centralized and a distributed PCA-based approach to detect outliers in sensed data in WSNs. We partition the network into groups of sensor nodes. Each group has a group head and several member nodes. In the centralized approach, every member node sends its local sensed data to the group head. The group head then applies PCA to establish a global normal pattern and detect outliers. In the distributed approach, we partition the network into groups of sensor nodes. Each group has a group head and several member nodes. Rather than sending all sensed data, every member node uses fixed-width clustering (FWC) and sends a description of its sensed data to the group head. The group head then applies distributed principal component analysis (DPCA) in order to establish a global normal pattern and to detect outliers. The established normal pattern is periodically updated using a forgetting curve.

We compared the performance of the distributed approach with that of a centralized approach based on real sensed data collected from 54 Mica2Dot sensors deployed in Intel Berkeley Research Lab. The experimental results showed that the distributed approach achieves 93.09% reduction in communication overhead in comparison to the centralized approach, while achieving the similar detection and false alarm rates.

Acknowledgment

This work was supported in part by the Iran Telecommunication Research Center (ITRC).

References

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks, IEEE Communications Magazine. 40(8), 104–112.

[2] Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., and Gunopulos, D. (2006). Online outlier detection in sensor data using non-parametric models, in Proceedings of the 32nd International Conference on Very Large Databases, Seoul, Korea.

[3] Zhang, Y., Meratnia, N., and Havinga, P. (2008). Outlier detection techniques for wireless sensor networks: A survey, Journal of Communications Surveys & Tutorials. 12(2), 159–170.

[4] Martincic, F. and Schwiebert, L. (2006). Distributed event detection in sensor networks, in Proceedings of the International Conference on Systems and Networks Communications, Tahiti, French Polynesia.

[5] Chatzigiannakis, V. and Papavassiliou, S. (2007). Diagnosing anomalies and identifying faulty nodes in sensor networks, IEEE Sensors Journal. 7(5), 637–645.

[6] Barnett, V. and Lewis, T. (1994). Outliers in Statistical Data, New York: John Wiley Sons.

[7] Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies, Journal of Artificial Intelligence Review. 22(2), 85–126.

[8] Tan, P.-N., Steinbach, M., and Kumar, V. (2004). Introduction to Data Mining, New York: Addison-Wesley.

[9] Chandola, V., Banerjee, A., and Kumar, V. (2007). Outlier detection: A survey, Technical Report, University of Minnesota.

[10] Ilyas, M., Mahgoub, I., and Kelly, L. (2004). Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, London: CRC Press.

[11] Polastre, J., Szewczyk, R., and Culler, D. (2005). Telos: Enabling ultra-low power wireless research," in Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, CA, USA.

[12] Raghunathan, V., Schurgers, C., Park, S., and Srivastava, M. (2002). Energy aware wireless microsensor networks, IEEE Signal Processing Magazine. 19(2), 40–50.

[13] Zhao, F., Liu, J., Guibas, L., and Reich, J. (2003). Collaborative signal and information processing: An information-directed approach, Proceedings of the IEEE. 91(8), 1199–1209.

[14] Ahmadi Livani, M. and Abadi, M. (2010). An energy-efficient anomaly detection approach for wireless sensor networks, in Proceedings of the 5th International Symposium on Telecommunications, Tehran, Iran.

[15] Nakayama, H., Kurosawa, S., Jamalipour, A., Nemoto, Y., and Kato, N. (2009). Dynamic anomaly detection scheme for AODV-based mobile ad hoc networks, IEEE Transactions on Vehicular Technology. 58(5), 2471–2481.

[16] Janakiram, D., Mallikarjuna, A., Reddy, V., and Kumar, P. (2006). Outlier detection in wireless sensor networks using Bayesian belief networks, in Proceedings of 1st International Conference on Communication System Software and Middleware, New Delhi, India.

[17] Rajasegarar, S., Leckie, C., and Palaniswami, M. (2006). Distributed anomaly detection in wireless sensor networks, in Proceedings of the 10th IEEE International Conference on Communication Systems, Singapore.

[18] Rajasegarar, S., Leckie, C., Palaniswami, M., and Bezdek, J. C. (2007). Quarter sphere based distributed anomaly detection in wireless sensor networks, in Proceedings of the IEEE International Conference of Communication, Glasgow, UK.

[19] Sheng, B., Li, Q., Mao, W., and Jin, W. (2007). Outlier detection in sensor networks," in Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Montreal, Canada.

[20] Li, G., He, J., and Fu, Y. (2008). Group-based intrusion detection system in wireless sensor networks, Computer Communications, 31(18), 4324–4332.

[21] Golub, G. H. and Van Loan, C. F. (1996). Matrix Computations, Johns Hopkins University Press, Third Edition.

[22] Bai, Z.-J., Chan, R. H., and Luk, F. T. (2005). Principal component analysis for distributed data sets with updating, in Proceedings of International workshop on Advanced Parallel Processing Technologies, Singapore.

[23] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data, Applications of Data Mining in Computer Security, Kluwer Academic Publishers.

[24] Jolliffe, I. T. (2002). Principal Component Analysis, New York: Springer-Verlag.