Ensuring high sensor data quality through use of online outlier detection techniques

Yang Zhang*, Nirvana Meratnia and Paul J.M. Havinga

Pervasive Systems Group,
University of Twente,
Drienerlolaan 5,
7522 NB Enschede,
The Netherlands
Fax: +31 53 489 4590
Email: zhangy@cs.utwente.nl
Email: meratnia@cs.utwente.nl
Email: havinga@cs.utwente.nl
*Corresponding author

Abstract: Data collected by Wireless Sensor Networks (WSNs) are inherently unreliable. Therefore, to ensure high data quality, secure monitoring, and reliable detection of interesting and critical events, outlier detection mechanisms are needed to be in place. The constraint nature of resources available in WSNs necessities that unlike traditional outlier detection techniques performed off-line, outliers to be identified in an online manner. This means that outliers in distributed streaming data should be detected in (near) real time with a high accuracy while maintaining the resource consumption of the WSN to a minimum. In this paper, we propose outlier detection techniques based on one-class quarter-sphere support vector machine meeting constraints and requirements of WSNs. To reduce the false alarm rate while increasing the detection rate and to enable collaborative outliers detection, we take advantage of spatial and temporal correlations that exist between sensor data. Experiments with both synthetic and real data show that our distributed and online outlier detection techniques achieve better detection accuracy and lower false alarm compared to an earlier distributed, batch outlier detection technique designed for WSNs.

Keywords: WSNs; wireless sensor networks; outlier detection; SVM; support vector machine; spatial correlations; temporal correlations.

Reference to this paper should be made as follows: Zhang, Y., Meratnia, N. and Havinga, P.J.M. (2010) 'Ensuring high sensor data quality through use of online outlier detection techniques', *Int. J. Sensor Networks*, Vol. 7, No. 3, pp.141–151.

Biographical notes: Yang Zhang is currently a PhD student in the Pervasive System Group at the University of Twente in the Netherlands. He received his BS and MS degrees in Computer Science and Technology from the University of Jiangsu, China, in 2002 and 2004, respectively. He received his second MS degree in Telematics from the University of Twente, the Netherlands, in 2006. His research interests include wireless sensor networks, anomaly/outlier detection, event detection, data mining, machine learning and wireless/mobile communication. He is involved as publicity co-chair and reviewer for conferences and workshops.

Nirvana Meratnia is an Assistant Professor in the Pervasive System Group at the University of Twente. She received her PhD in 2005 from the Database Group at the same university. Her research interests are in the area of distributed data management in wireless sensor networks, smart and collaborative objects, ambient intelligence, and context-aware applications. She is actively involved as programme committee member and reviewer for conferences and workshops.

Paul J.M. Havinga is an Associate Professor in the Pervasive System Group at the University of Twente in the Netherlands. He received his PhD on the thesis entitled 'Mobile Multimedia Systems' in 2000 and was awarded with the 'DOW Dissertation Energy Award' for this work. His research interests include large-scale, heterogeneous wireless systems, sensor networks, energy-efficient architectures and protocols, ubiquitous computing, and personal communication systems. This research has resulted in over 180 scientific publications in journals and conferences. He is the editor of several journals and magazines and regularly serves as programme committee chair, member and reviewer for conferences and workshops.

1 Introduction

Tremendous advances in electronics and wireless communications technologies have enabled the reality of tiny and low-cost sensor devices, also known as *sensor nodes*, which are equipped with wireless transceivers, low-power microcontroller, energy sources and various types of sensors. A Wireless Sensor Network (WSN) consists of a large number of these nodes distributed over a large geographical area to cooperatively monitor a phenomenon. A wide variety of applications of WSNs ranges from personal spaces to scientific, industrial, business and military domains. In a typical application, a WSN collects continuous *streaming data*, performs in-network data processing and takes local decisions in (near) real time. Thus, providing a high-quality stream of sensor data is essential for the decision-making process in WSN applications.

Compared to wired networks, WSNs have strong resource constraints in terms of energy, memory, computational capacity and communication bandwidth. Moreover, the large-scale and dense vision of the WSN dictates that the network usually has to operate in a harsh and unattended environment. The resource constraints and environmental effects make a WSN more vulnerable to faults and malicious activities (e.g. denial of service attacks or black hole attacks), and cause unreliable and inaccurate sensor readings. To ensure a reasonable data quality, secure monitoring and reliable detection of interesting and critical events and to facilitate effective and correct decision-making using data collected by WSNs, identifying anomalous measurements in point of action is a must. In WSNs, *outliers*, also known as *anomalies*, are those measurements that do not conform to the normal behavioural pattern of the sensed data (Chandola et al., 2007).

Extracting knowledge from multiple distributed sensor data streams is not a trivial task (Gaber, 2007). Conventional outlier detection techniques (Hawkins, 1980; Barnett and Lewis, 1994; Knorr and Ng, 1998; Breunig et al., 2000) may not be suitable for the context of streams and distributed environments of WSNs. They have paid limited attention to reasonable availability of computational resources. They also assume the stationary data distribution, store all the data in memory at the same time, and perform data analysis in a centralised approach. In WSNs, an appropriate outlier detection technique should pay attention to limitations in terms of computing, communication and storage of the network and deal with the distributed data analysis. Thus, a key objective for outlier detection in WSNs is to identify outliers in the distributed streaming data in an online manner with a high accuracy while maintaining the resource consumption of the network to a minimum.

Based on one-class quarter-sphere Support Vector Machine (SVM) (Laskov et al., 2004), we propose distributed and Online Outlier Detection (OOD) techniques appropriate for resource-constrained WSNs. Since sensor data of adjacent nodes in a densely deployed WSN tend to be spatially and temporally correlated (Vuran et al., 2004), we take advantage of spatial and temporal correlations that exist in sensor data to cooperatively identify outliers and also distinguish between events and errors in real time. Experiments with both synthetic and real data collected by the Intel Berkeley Research Laboratory (IBRL, 2004) and the SensorScope System (SensorScope, 2007) show that our outlier detection techniques achieve better detection accuracy and lower false alarm compared to an earlier distributed Batch Outlier Detection (BOD) technique (Rajasegarar et al., 2007) designed for WSNs.

The contributions of this paper can be summarised as:

- extending existing offline and static techniques to be able to detect outliers in an online and adaptive manner to meet WSN requirements.
- proposing a decision function to determine whether every new measurement as normal or anomalous.
- taking advantage of spatiotemporal correlations by using information of neighbouring nodes to identify outliers. This in turn helps in better detection accuracy.
- presenting preliminary work on online distinction between event and error.

The remainder of this paper is organised as follows. Related work on one-class SVM-based outlier detection techniques is presented in Section 2. Fundamentals of the one-class centred quarter-sphere SVM are described in Section 3. Our proposed distributed and OOD techniques are explained in Section 4. Experimental results and performance evaluation are reported in Section 5. The paper is concluded in Section 6 with plans for future research.

2 Related work

The purpose of data mining is to find and extract hidden valuable information from a data set (Tan et al., 2006). Compared to the other three data mining tasks, i.e. predictive modelling, cluster analysis and association analysis, outlier detection is the closest task to the initial motivation behind data mining (Hodge and Austin, 2003). Outlier detection has been widely researched in various disciplines such as statistics, data mining, machine learning, information theory and spectral decomposition (Chandola et al., 2007). Generally speaking, outlier detection techniques can be categorised into statistical-based, nearest neighbour-based, clustering-based, classification-based and spectral decomposition-based approaches (Chandola et al., 2007; Zhang et al., 2008). Classification-based approaches are important systematic approaches in the data mining and machine learning communities. They learn a classification model using a set of data instances in the training phase and classify an unseen instance into one of the learned (normal/outlier) class in the testing phase. SVM-based techniques are from family of classification-based approaches and separate the data belonging to different classes by fitting a hyperplane that produces a maximal margin. They have the following three main advantages:

- do not require an explicit statistical model
- provide an optimum solution for classification by maximising the margin of the decision boundary;
- avoid the curse of dimensionality problem.

One of the challenges faced by SVM-based outlier detection techniques for WSNs is obtaining error-free or labelled data for training. One-class (*unsupervised*) SVM-based techniques address this challenge by modelling normal behaviour of the unlabelled data while automatically ignoring the anomalies existing in the training set. The main idea of one-class SVM-based outlier detection techniques is to use a non-linear function to map the data vectors collected from the *original space* to a higher dimensional space called *feature space.* Then a decision boundary of normal data is found, which encompasses the majority of the data vectors in the feature space. Those new unseen data vectors falling outside the boundary are classified as outliers. Scholkopf et al. (2001) have proposed a hyperplane-based one-class SVM, which identifies outliers by fitting a hyperplane from the origin. Tax and Duin (2004) have proposed a hypersphere-based one-class SVM, which identifies outliers by fitting a hypersphere with a minimal radius.

In addition to obtaining the labelled data, another challenge faced by SVM-based outlier detection techniques is inapplicability of their quadratic optimisation during the learning process for WSN applications. This process is extremely costly and not suitable for limited resources available in WSNs. Laskov et al. (2004) have extended work of Tax and Duin (2004) by proposing a one-class quarter-sphere SVM, which is formulated as a linear optimisation problem and thus reduces the effort and computational complexity. Rajasegarar et al. (2007) further exploit potential of the one-class quarter-sphere SVM of Laskov et al. (2004) for distributed outlier detection in WSNs. Their technique assumes that the history prior to the window does not influence current behaviour and it detects outliers at each node only after collecting a large number of data measurements in a time window. This technique ignores the temporal and spatial correlations that exist in sensor data and performs outlier detection in an offline and batch manner. Moreover, it causes too long detection delay and makes, therefore, the technique unsuitable for real-time applications of WSNs. We will use it for our comparison and performance evaluation in Section 5.

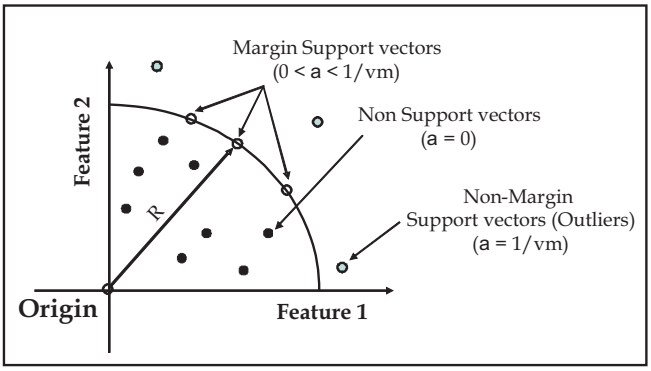
As mentioned before, one-class SVM-based outlier detection techniques build a model representing normal behaviour of the sensed data and identify an outlier as a sensor measurement that does not conform to this model. Due to the fact that sensor data is streaming data, i.e. an ordered sequence of unbounded real-time data records with a high data rate, a normal model will evolve over time and the defined normal model may not be sufficiently representative for future identification. Davy et al. (2006) have considered the change of the normal model over time and identified outliers using previous data vectors in a sliding time window. However, this technique is not applicable to WSNs since it has expensive computational effort and also ignores the spatial correlations in spatial data measurements of adjacent nodes in a densely deployed WSN.

In this paper, we extend the technique of Laskov et al. (2004), and first propose a distributed and OOD technique to identify every new measurement collected at each node as normal or anomalous in real time. Based on this technique, we then propose three Adaptive Outlier Detection (AOD) techniques, which take different strategies to sequentially update the model representing normal behaviour of the sensed data. As it will be shown in the next sections, they can be used to satisfy specific user requirements such as fast responsiveness, high detection and low computation.

3 The one-class quarter-sphere SVM

In our proposed techniques, we exploit the one-class centred quarter-sphere SVM of Laskov et al. (2004) to build the normal model of sensor measurements. They have converted the quadratic optimisation problem of the one-class SVM to a linear optimisation problem. The geometry of the one-class centred quarter-sphere SVM-based approach is shown in Figure 1.

Figure 1 Geometry of the quarter-sphere formulation of one-class SVM (Laskov et al., 2004)



The constrained optimisation problem of the one-class centred quarter-sphere SVM is formalised as follows:

$$\min_{R\in\Re,\xi\in\Re^m} \quad R^2+\frac{1}{vm}\sum_{i=1}^{m}\xi_i \tag{1}$$

$$\textit{subject to}:\ \left\|\phi(x_i)\right\|^2 \le R^2+\xi_i, \xi_i \ge 0, i=1,2,...m$$

where m denotes number of data vectors in the training set, and $\{\xi_i : i = 1, 2,.. . m\}$ are the slack variables that allow some of the data vectors to fall outside the quarter-sphere. The parameter $v \in (0,1)$ is a regularisation parameter that represents the fraction of data vectors that can be outliers. The Lagrange function for this optimisation is:

$$L = R^2-\sum_{i=1}^{m}\alpha_i\left(R^2-\left\|\phi(x_i)\right\|^2+\xi_i\right)-\sum_{i=1}^{m}\beta_i\xi_i+\frac{1}{vm}\sum_{i=1}^{m}\xi_i \tag{2}$$

where $\alpha_i \geq 0$, $\beta_i \geq 0$ for all i = 1,2,..., m are the Lagrangian multipliers. Taking the zero derivatives of L with respect to R and ξ_i result to:

$$\frac{\partial L}{\partial R} = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i = 1 \tag{3}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \Rightarrow \alpha_i = \frac{1}{vm} - \beta_i \tag{4}$$

From equation (4), we can obtain $0 \leq \alpha_i \leq \frac{1}{vm}$ using $\alpha_i \geq 0$, $\beta_i \geq 0$. Substituting equations (3) and (4) into equation (2) produces:

$$L = \sum_{i=1}^{m} \alpha_i \left(\left\| \phi(x_i) \right\|^2 \right) = \sum_{i=1}^{m} \alpha_i \left(\phi(x_i) \cdot \phi(x_i) \right) \tag{5}$$

where $\left\| \phi(x_i) \right\|^2$ is given by the dot product of $\phi(x_i) \cdot \phi(x_i)$, which indicates a measure of similarity between $\phi(x_i)$ and $\phi(x_i)$ in the feature space. A kernel function $k(x_i,x_i)$ is used to compute the similarity of any of two vectors in the feature space using the original attribute set (Tan et al., 2006). Hence, the dual formulation of equation (1) will become:

$$\min_{\alpha \in \Re^m} \; -\sum_{i=1}^{m} \alpha_i k(x_i, x_i) \tag{6}$$

$$\textit{subject to}: \sum_{i=1}^{m} \alpha_i = 1, 0 \leq \alpha_i \leq \frac{1}{vm}, i = 1, 2, ...m$$

Converting the dual problem of a quadratic optimisation to a linear optimisation problem effectively reduces the computational complexity. In order to fix the centre of the quarter-sphere at the origin, the mapped data vectors in the feature space need to be subtracted from the mean, i.e. $\mu = \frac{1}{vm} \sum_{i=1}^{m} \phi(x_i)$. The centred kernel matrix K_c can be obtained in terms of the kernel matrix $K = k(x_i, x_j) = \left(\phi(x_i) \cdot \phi(x_j) \right)$ using $K_c = K - 1_m K - K 1_m + 1_m K 1_m$, where 1_m is an $m \times m$ matrix with all values equal to $\frac{1}{m}$.

From equation (6), the $\{\alpha_i\}$ value can be easily obtained using some effective linear optimisation techniques (Nash and Sofer, 1996). The data vectors in the training set can be classified depending on the results of $\{\alpha_i\}$, as shown in Figure 1. The training data vectors with $0 \leq \alpha \leq \frac{1}{vm}$, which fall on the quarter-sphere, are called *margin support vectors*. Their distances to the origin indicate the minimal radius R of the quarter-sphere and can be used to identify any new unseen data vector as normal or anomalous. Those data vectors whose distances to the origin are larger than R are detected as outliers and these are the measurement we are interested in.

4 Outlier detection techniques for WSNs

In this section, we will describe our four distributed and online techniques. The first technique aims at identifying every new measurement collected at each node as normal or anomalous in real time. The latter three techniques take different strategies to sequentially update the normal model representing normal behaviour of the sensed data. The policies concerning updating the normal model in these techniques include updating (a) at each time interval, (b) at a fixed-size time window, and (c) depending on the previous decision results. These proposed techniques enable each sensor node in the network to exploit temporal correlations among its most recent sensor measurements to identify its new arriving measurement as normal or anomalous in real-time. Moreover, using high-degree spatial correlations that exist between sensor readings of adjacent nodes, each node has more information to cooperatively identify outliers. The whole detection process does not only depend on a node's own decision criterion learned from its temporal readings but also on the decision criteria learned from its spatially neighbouring nodes.

Before explaining the techniques in details, we draw reader's attention to our set of assumptions. We assume that sensor nodes are time synchronised and are densely deployed in a homogeneous WSN, where sensor data tends to be correlated in both time and space. A sensor sub-network consists of n sensors nodes S_1,S_2,... S_n, which are within radio transmission range of each other. It means that each node has n–1 spatially neighbouring nodes in the sub-network. At each time interval Δ_i, each sensor node in the sub-network measures a data vector. Let $x_1^i, x_2^i, ..., x_n^i$ denote the data vector measured at S_1, S_2,… S_n, respectively. Each data vector is composed of multiple attributes x_j^{il}, where $x_j^i = \left\{ x_j^{il} : j = 1...n, l = 1...d \right\}$ and $x_j^i \in \Re^d$. At time t, each sensor node has collected its own m measurements from time $t - m$ to time $t - 1$: $x_j^t = \left\{ x_j^{t-m}, ... x_j^{t-1} : j = 1...n \right\}$. From time t, each node identifies every new measurement as normal or outlier.

4.1 Online Outlier Detection (OOD) technique

As previously mentioned, a straightforward approach for outlier detection in WSNs is to build a model representing normal behaviour of the sensed data and identify an outlier as a sensor measurement that does not conform to this model. Our OOD technique is built on this principle and goes further.

Initially, each node learns the local radius of the quarter-sphere using its m sequential data measurements, which may include some anomalous data. The one-class quarter-sphere SVM can efficiently find a minimal radius R to enclose the majority of these mapped sensor measurements in the feature space. Each node then locally broadcasts the learned radius information to its spatially neighbouring

nodes. When receiving the radii from all of its neighbours, each node computes a median radius R_m of its neighbouring nodes and a median global radius R_g of all nodes in the sub-network. We use median because in estimating the 'centre' of a sample set, the median is more robust than the mean.

When a new sensor measurement x_i^t is collected at time t, node S_i first compares the distance of x_i^t from the origin with the radius R learned with respect to its m previous measurements $\{x_i^{t-m},...,x_i^{t-1}\}$ in a sliding window. According to the mean $\mu = \frac{i}{m}\sum_{i=1}^{m}\phi(x_i)$ and the kernel matrix $K = k(x_i, x_j) = (\phi(x_i)\cdot\phi(x_j))$, the distance of x_i^t from the origin in the feature space is formalised as follows:

$$\begin{aligned} d(x) &= \sqrt{\left\|\phi(x) - \frac{1}{m}\sum_{i=1}^{m}\phi(x_i)\right\|^2} \\ &= \sqrt{\|\phi(x)\|^2 + \frac{1}{m^2}\sum_{i=1}^{m}\|\phi(x_i)\|^2 - \frac{2}{m}\sum_{i=1}^{m}\phi(x)\cdot\phi(x_i)} \\ &= \sqrt{\phi(x)\cdot\phi(x) + \frac{1}{m^2}\sum_{i=1}^{m}\phi(x_i)\cdot\phi(x_i) - \frac{2}{m}\sum_{i=1}^{m}k(x, x_i)} \\ &= \sqrt{k(x, x_i) + \frac{1}{m^2}\sum_{i=1}^{m}k(x_i, x_j) - \frac{2}{m}\sum_{i=1}^{m}k(x, x_i)} \end{aligned} \quad (7)$$

Based on the fact that sensor data collected in a densely deployed WSN tends to be spatially and temporally correlated (Vuran et al., 2004), the data x_i^t will be classified as normal if $d(x) <= R$, which means that x_i^t falls on or inside the quarter-sphere at S_i. Otherwise if $d(x) > R$, S_i further compares $d(x)$ with the median radius R_m of its spatially neighbouring nodes. Then if $d(x) > R_m$, x_i^t will finally be classified as outlier in the sub-network. The decision function to declare a measurement as normal or outlier can be formulated as equation (8), where the sensor measurements with a negative value are classified as outlier.

$$f(x) = sgn(max(R - d(x), R_m - d(x))) \quad (8)$$

Identifying what has caused the outlier in sensor data is an important task. Potential sources of outliers in data collected by WSNs include noise and errors, actual events and malicious attacks. Our proposed technique makes distinction between events and errors based on the observation that erroneous measurements are likely to be spatially unrelated, while event measurements are likely to be spatially correlated (Krishnamachari and Iyengar, 2004). The main idea is that only when x_i^t is considered as outlier, S_i collects the distances of all of its neighbouring nodes currently sensing data from their own origin and computes a median distance $d(x)_m$. If an event occurs in the sub-network, $d(x)$ and $d(x)_m$ will be temporally different but a spatial consensus will be observed. This means that $d(x)$ and $d(x)_m$ will exceed their own radius of R and R_m, respectively. Moreover, they both will exceed the median global radius R_g of the sub-network. If this is not the case, the detected outlier may indicate an erroneous measurement. The pseudocode of the OOD is shown in Table 1.

Table 1 Pseudocode of the OOD

```
1   procedure LearningSVM()
2     each node collects m sensor measurements for learning
      its own radius R and locally broadcasts the radius to its
      spatially neighbouring nodes;
3     each node then computes R_m and R_g;
4     initiate OutlierDetectionProcess(R, R_m);
5     return;
6   procedure OutlierDetectionProcess(R, R_m)
7     when x^t arrives
8       compute d(x);
9       if (d(x) > R AND d(x) > R_m)
10        x^t indicates an outlier;
11        initiate SourceOfOutlierProcess(R, R_m, R_g,
          d(x));
12      else
13        x^t indicates a normal measurement;
14      endif;
15    return;
16  procedure SourceOfOutlierProcess(R, R_m, R_g, d(x))
17    collect the distances of all of its neighbouring nodes'
      currently sensing data from their own origin and
      compute d(x)_m;
18    if (d(x) > R AND d(x)_m > R_m)
19      if (d(x) > R_g AND d(x)_m > R_g)
20        x^t may indicate an event;
21    else
22        x^t may indicate an erroneous measurement;
23    endif;
24    else
25      x^t may indicate an erroneous measurement;
26    endif;
27    return;
```

4.2 Instant Outlier Detection (IOD) technique

After each node uses the normal model learned in the OOD to identify outliers and makes distinction between events and errors, the normal model may not be sufficiently representative for future identification of outliers and thus need to be updated. The simplest method of updating the normal model over time is to compute the minimal radius of one-class quarter-sphere for each training set, i.e. at each time interval. Each update step needs to add a current measurement and to remove the oldest measurement from the training set. This procedure is repeated for evolving training set of fixed size. Figure 2 illustrates the update policy of IOD's model. The corresponding pseudocode modification for the IOD is shown in Table 2.

Figure 2 Update policy of IOD's model. Circles represent sensor measurements. The 'sliding' training set is composed of the last m measurements. The black dot represents the measurement identified at current time t

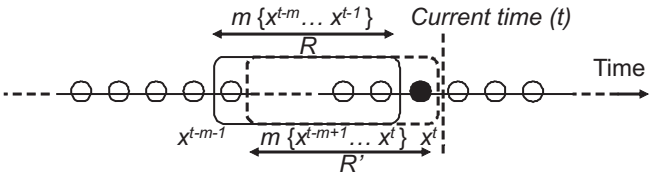


Table 2 Pseudocode of the IOD

1–5	procedure **LearningSVM()**
6–14	procedure **OutlierDetectionProcess(**R, R_m)
15	initiate **UpdatingModelProcess**(x^t);
16	set $t \leftarrow t + 1$;
17	return;
18	procedure **UpdatingModelProcess** (x^t)
19	update the training set: the oldest measurement x^{t-m} is removed and replaced by x^t;
20	recompute R using the updated training set;
21	locally broadcast R to its neighbouring nodes;
22	recompute R_m of its neighbouring nodes;
23	return;

Once the radius of a node is updated at each time interval, the node locally broadcasts the new radius R to its spatially neighbouring nodes. When receiving the radii from all of its neighbours, each node recomputes the median radius R_m of neighbouring nodes and the median global radius R_g of all nodes in the sub-network. The updated R, R_m and R_g are used to identify the next sensor measurement as normal or anomalous, and further distinguish between event and error.

4.3 Fixed-size Time Window-based Outlier Detection (FTWOD) technique

It is obvious that the IOD has expensive computational effort due to updating the normal model at each time interval. Thus, a slightly modified version of the IOD is to identify each sensor measurement upon being collected but update the normal model at a fixed-size time window. It means that the training set will be freezed to identify the next $n(n \ll m)$ measurements as normal or anomalous. This modification effectively reduces the times of recomputing the minimal radius of one-class quarter-sphere and prevents extremely high computational complexity. Furthermore, each of the n measurements upon arrival will be classified as normal or anomalous in real-time. This is to ensure that there is no delay in outlier detection itself and the training set is updated in periods of fixed-size time windows.

Each update step in this technique requires to add the previous n sensor measurements and to remove the oldest n measurements from the training set. Figure 3 illustrates the update policy of FTWOD. The corresponding pseudocode modification for the FTWOD technique is shown in Table 3. In general, n should be much smaller than m (m being the size of the training data set) because large n will result in missing small behavioural changes in the data set. Since environmental changes occur gradually, choosing a big n will lead to failure of the outlier detection techniques. On the contrary, if $n = 1$, the FTWOD becomes like the IOD, which is computationally expensive because of frequently updating the normal boundary at each time interval. Thus, we set the value of n to 5 in our experiment.

Figure 3 Update policy of FTWOD's model. The training set is updated at each n measurements

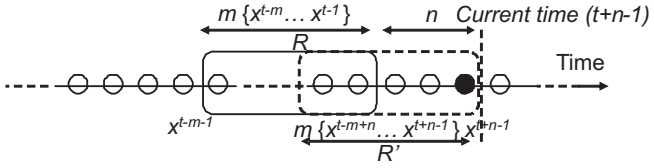


Table 3 Pseudocode of the FTWOD

……	
15	if (t % n = = 0)
15'	initiate **UpdatingModelProcess** $\left(x^{t-n+1}...x^t\right)$;
……	

4.4 Adaptive Outlier Detection (AOD) technique

The update policy of the above-mentioned techniques is updating the normal model either at each time interval or at n time intervals, with little consideration of the impact when a normal or anomalous measurement is incorporated into the sliding training set. Moreover, they introduce expensive complexity effort and communication load due to the fact that each node is required to frequently update the normal boundary and locally broadcast the updated R to its neighbouring nodes. Thus, for the sake of energy efficiency and computational simplicity, we introduce a third technique, which takes a new strategy to update the normal model depending on the previous decision results, i.e. only when a new measurement has a significant impact on the previous normal model.

As shown in Figure 1, the margin support vectors and outliers have non-zero α values so that the dual formulation of equation (1) will not be met if they are added into the existed training set. In order to meet the constraints specified in equation (6) and to find a minimal radius, when a current measurement is detected as margin support vector or outlier, we propose our AOD technique.

AOD updates the normal boundary when a current measurement is detected as outlier or support vector, and the radius R is updated based on the distance between the outlier measurement and support vectors. Then it adds all the previous n' measurements including the current measurement into the training set and removes the same amount of the oldest measurements from the training set. AOD identifies each new measurement upon being collected at each node as normal or anomalous in real time. Due to the fact that compared to normal data, outliers and margin support vectors

are very rare (Tax and Duin, 2004); this technique is more efficient in terms of energy and computational costs. Figure 4 illustrates the update policy of AOD. The corresponding pseudocode modification for the AOD technique is shown in Table 4.

Figure 4 Principle of the AOD. The black dot represents the measurement identified as a margin support vector or an outlier

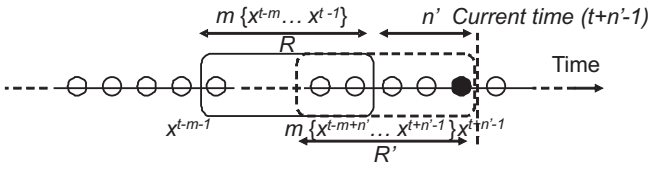


Table 4 Pseudocode of the AOD

……	
15	if (x^t is an outlier or a margin support vector
15'	initiate **UpdatingModelProcess** $\left(x^{t-n'+1}...x^t\right)$;
……	

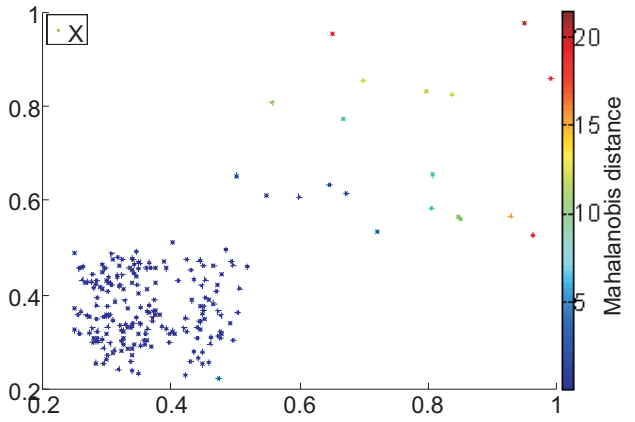
5 Experimental results and evaluation

This section describes the performance evaluation of our four techniques compared to the distributed, BOD technique presented earlier by Rajasegarar et al. (2007). In our experiments, we have used synthetic data as well as real data gathered from a deployment of WSN in the Intel Berkeley Research Laboratory (IBRL, 2004) and in the SensorScope System (SensorScope, 2007).

5.1 Synthetic data

For the simulation, we use MATLAB and consider a sensor sub-network consisting of seven sensor nodes, which are within radio transmission range of each other. It means that each node has six spatially neighbouring nodes in the sub-network. The 2-D synthetic data used for each node is composed of a mixture of three Gaussian distribution with uniform outliers; the mean is randomly selected from (0.3, 0.35, 0.45), and the standard deviation is selected as 0.03. Subsequently, 10% (of the normal data) anomalous data is introduced and uniformly distributed in the interval [0.5, 1]. The data values are normalised to fit in the [0, 1]. The BOD technique of Rajasegarar et al. (2007) identifies outliers at each node using a local radius and evaluates its performance using the original training data with added labels rather than testing data. To have a fair comparison, thus in the experiment we use the same data measurements in both techniques for evaluating the performance while the same amount of training data is used to learn the quarter-sphere SVM classifier of our techniques. The testing data used for each node comprise 200 normal and 20 anomalous data. Figure 5 illustrates data distribution of the synthetic data.

Figure 5 Plot for synthetic data (see online version for colours)



5.2 Real data from Intel Berkeley Research Laboratory (IBRL)

This data set is collected from a cluster of neighbouring sensor nodes from a WSN deployed in the Intel Berkeley Research Laboratory. Figure 6(a) illustrates the set-up. The sub-network consists of seven sensor nodes, namely nodes 1, 2, 33, 34, 35, 36, 37. The network recorded temperature, humidity, light and voltage measurements at 31 seconds intervals. In our experiments, we use a 6:00 am–6:00 pm period of data recorded on 5th March 2003 with two attributes: temperature and humidity for each data measurement. The data values are normalised to the range [0, 1]. The labels of data measurements are obtained using their degree of dissimilarity. As shown in Figure 6(b), the data vectors are labelled as anomalous if they are determined to be distant from other data vectors using the *Mahalanobis distance*. The same amount of testing data as the training data is used to evaluate the performance of our techniques.

5.3 Real data from SensorScope system

This data set is collected from a cluster of neighbouring sensor nodes from a WSN deployed in Grand-St-Bernard. Figure 7(a) illustrates the deployment. The sub-network consists of seven sensor nodes, namely nodes 2, 3, 4, 8, 12, 14, 20. The network recorded ambient temperature, relative humidity, soil moisture, solar radiation and watermark measurements at 2 minutes intervals. In our experiments, we use a 6:00 am–6:00 pm period of data recorded on 20th September 2007 with two attributes: ambient temperature and relative humidity for each sensor measurement. The data values are normalised to the range [0, 1]. The labels of data measurements are obtained using their degree of dissimilarity. As shown in Figure 7(b), the data vectors are labelled as anomalous if they are determined to be distant from other data vectors using the Mahalanobis distance. The same amount of testing data as the training data is used to evaluate the performance of our techniques.

Figure 6 (a) Sensor nodes deployed in Intel Berkeley Research Laboratory (IBRL, 2004); (b) Plot for real data from IBRL (see online version for colours)

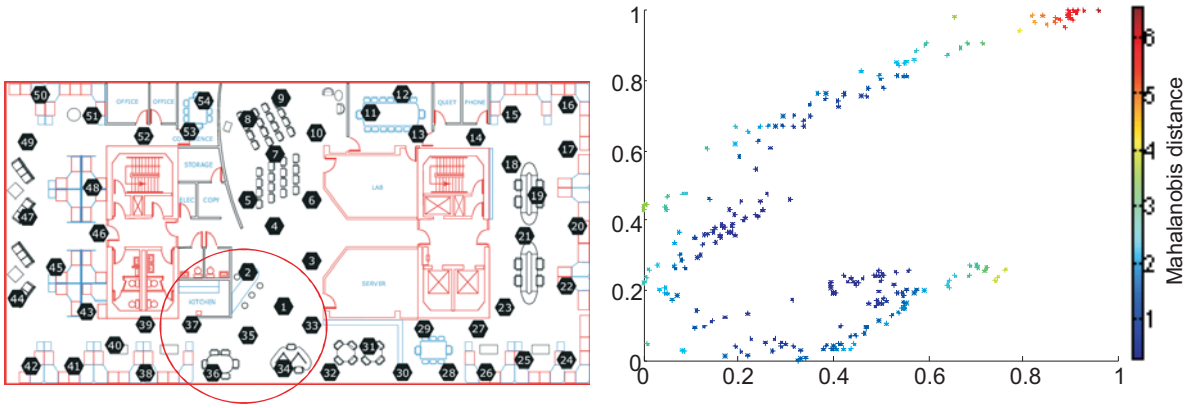
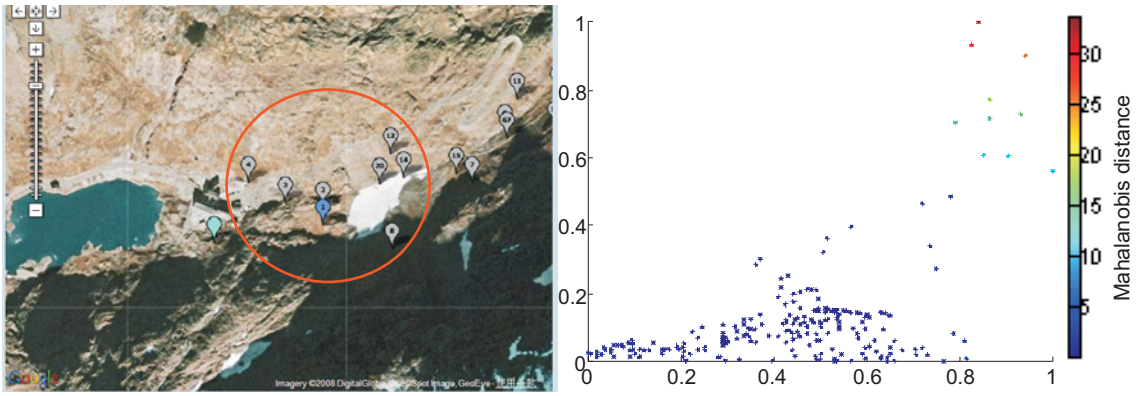


Figure 7 (a) Grand-St-Bernard deployment in SensorScope System(SensorScope, 2007); (b) Plot for real data from SensorScope System (see online version for colours)



5.4 Experimental results and evaluation

We have tested the following three kernel functions:

- Linear kernel function: $k_{Linear} = (x_1.x_2)$, where $\{x_1,x_2\}$ are the data vectors
- Radial Basis Function (RBF) kernel function: $k_{RBF} = \exp\left(-\|x_1 - x_2\|^2 / \sigma^2\right)$, where σ is the width parameter of the kernel function
- Polynomial kernel function: $k_{Polynomial} = (x_1 \cdot x_2 + 1)^r$, where r is the degree of the polynomial.

Kernel matrices generated using the above kernel functions were centred. We have evaluated two important performance metrics, the *detection rate*, which represents the percentage of anomalous data that are correctly considered as outliers, and the *false alarm rate*, also known as False Positive Rate (FPR), which represents the percentage of normal data that are incorrectly considered as outliers.

We have examined the effect of the regularisation parameter υ for our four online techniques and the BOD technique presented by Rajasegarar et al. (2007), using the linear, RBF and polynomial kernel functions. υ represents the fraction of data vectors that can be outliers. The higher value of the parameter υ can achieve the better detection accuracy; meanwhile it can also lead to the higher false alarm rate. So an appropriate value of the parameter υ should match exactly the anomaly ratio. In the case of no a priori knowledge about the anomaly ratio, varying υ can be used to evaluate the robustness of the techniques. A robust technique can achieve high accuracy rate while keeping a false alarm rate low with the parameter υ increases or decreases. In the experiments we have varied it in the range from 0.01 to 0.25 in intervals of 0.02, the kernel width parameter σ is set to 0.25, and the kernel degree parameter r is set to 3. A Receiver Operating Characteristics (ROC) curve is usually used to represent the trade-off between the detection rate and the false alarm rate. The larger the area under the ROC curve, the better the performance of the technique.

Figure 8 shows the ROC curves obtained for the five techniques using the linear, RBF and polynomial kernel functions for the synthetic data. The ROC curves shows that our four techniques achieve better performance than the BOD technique with different kernel functions. Due to the fact that the obtained results using the RBF kernel function effectively compare the performance of these techniques, Figure 9 shows the detection rate and the false alarm rate obtained for the five techniques using the RBF kernel function for real data from IBRL. Figure 10 shows the detection rate and the false alarm rate obtained for the five techniques using the RBF kernel function for real data from SensorScope System. These simulation results show that our four techniques achieve better accuracy in terms of parameter selection using different kernel functions compared to the BOD technique used by Rajasegarar et al. (2007).

Figure 8 (a) ROC curves with Linear kernel for synthetic data; (b) ROC curves with RBF kernel for synthetic data; (c) ROC curves with Polynomial kernel for synthetic data (see online version for colours)

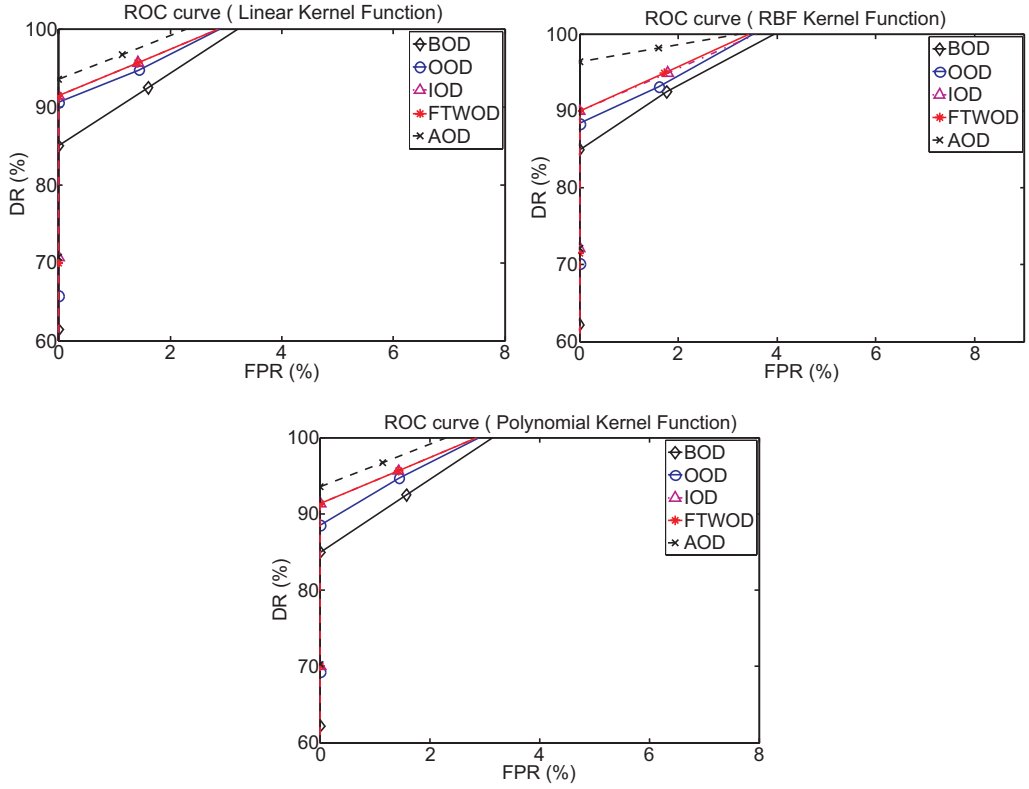


Figure 9 (a) Detection rate with RBF kernel for IBRL data; (b) False Alarm rate with RBF kernel for IBRL data (see online version for colours)

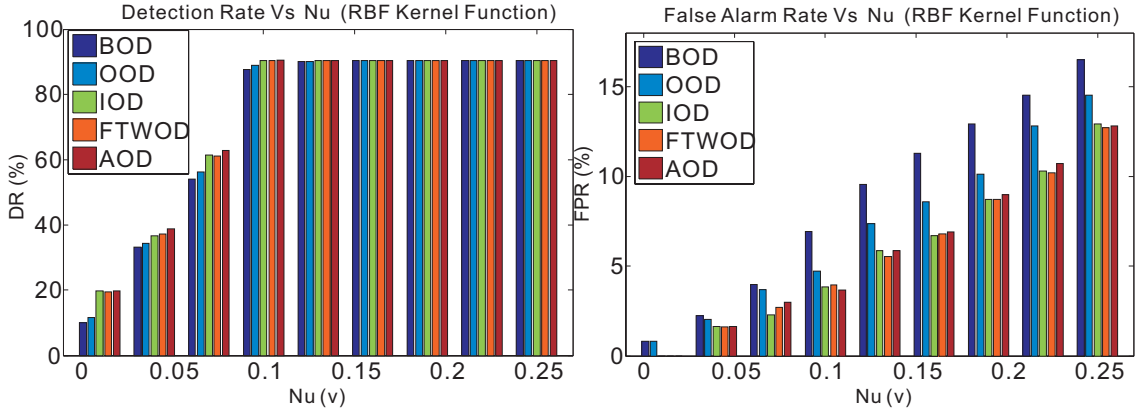


Figure 10 (a) Detection rate with RBF kernel for SensorScope System data; (b) False Alarm rate with RBF kernel for SensorScope System data (see online version for colours)

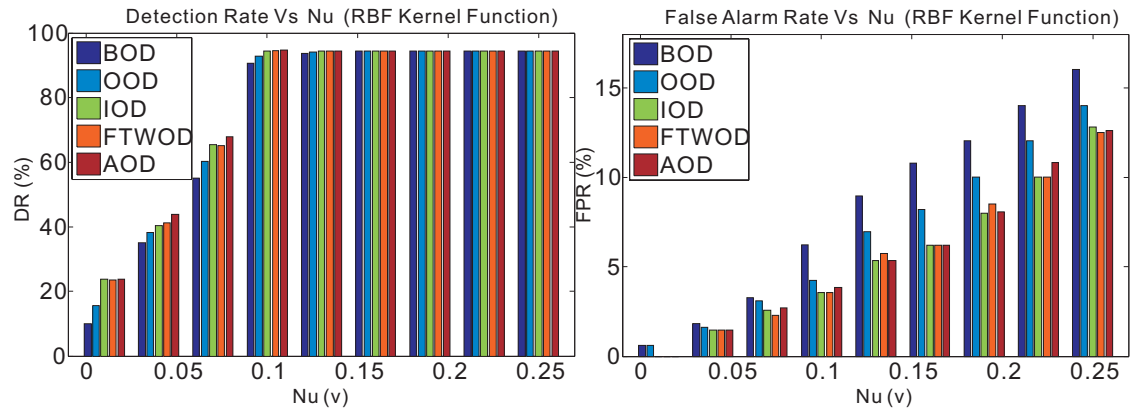
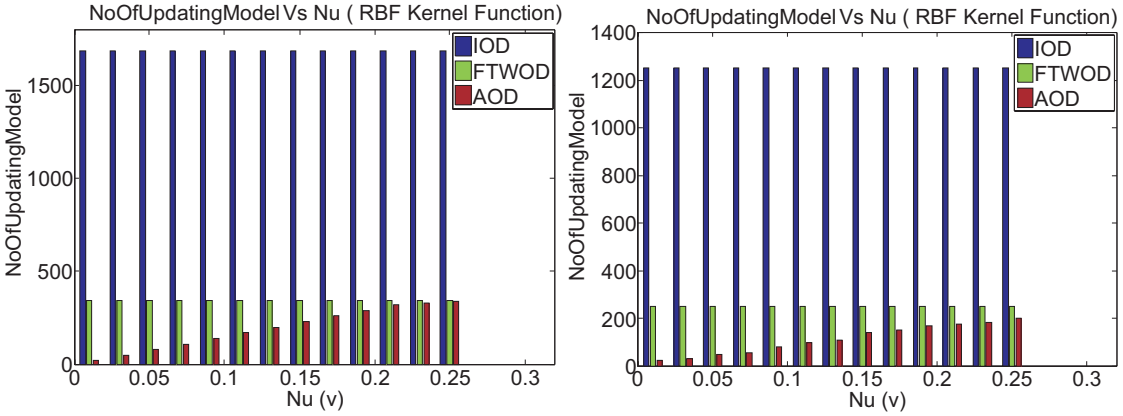


Figure 11 (a) Number of updating model with RBF kernel for IBRL data; (b) Number of updating model with RBF kernel for SensorScope System data (see online version for colours)



Furthermore, we compare the three techniques (IOD, FTWOD and AOD) concerning how often they update the normal boundary in both real data sets using the RBF kernel function. Results are depicted in Figure 11. We set n to 5 for the FTWOD. The simulation results show that the AOD updates the normal boundary in less time than the other two techniques. Thus, the AOD is more efficient in terms of communication overhead and computational complexity than the IOD and the FTWOD.

We further compare these techniques in terms of computational and memory complexity, as presented in Table 5, where m and N devote the number of data in the training and testing sets (N >= m), respectively, n devotes the size of time window in the FTWOD, n' devotes the times of relearning the minimal radius in the AOD$\left(n' < \frac{N}{n}\right)$, d represents the dimensionality of the measurements, and $O(L)$ represents the computational complexity of solving a linear optimisation problem in the training set, which consists of m data vectors.

From this table, we can see that our four techniques have a lower computational and memory complexity as well as a faster responsiveness for detecting outliers compared to the offline and batch technique of BOD. We can conclude that among these four techniques, IOD, FTWOD and AOD achieve better detection accuracy for distributed streaming data. The OOD technique, however, has the lowest computational and memory complexity and the fastest responsiveness. Furthermore, unlike IOD and FTWOD, that simply update the normal model either at each time interval or at a fixed-sized time interval, AOD takes the new strategy to update the normal model depending on the previous decision results and therefore achieves better detection accuracy, faster responsiveness and lower computational and memory complexity.

Table 5 Complexity analysis of five outlier detection techniques for WSNs

Techniques	*Computational complexity*		*Memory complexity*
	Training	*Testing*	
BOD	$O\left(\frac{N}{m} * L\right)$	$O(N)$	$O(d * N)$
OOD	$O(L)$	$O(N)$	$O(d * m)$
IOD	$O(N * L)$	$O(N)$	$O(d * m)$
FTWOD	$O\left(\frac{N}{n} * L\right)$	$O(N)$	$O(d * (m + n))$
AOD	$O(n' * L)$	$O(N)$	$O(d * (m + n'))$

6 Conclusions

Sensory data is inherently unreliable and inaccurate. To improve the sensor data quality, in this paper we have proposed four distributed and OOD techniques. These techniques are based on one-class centred quarter-sphere SVM and have low resource consumption, which make them suitable for resource constraint nature of WSNs. We compare performance of our techniques with a previously proposed distributed and batch technique using both synthetic and real data sets. Experimental results show that our approaches achieve better detection accuracy and lower false alarm in terms of parameter selection with different kernel functions, while keeping the computational complexity and memory costs low. We have also presented preliminary work on distinction mechanisms between events and errors. Our future research includes evaluating outlier detection performance while distinguishing between events and errors and real implementation of the protocols on wireless sensor nodes.

Acknowledgements

This work is supported by the EU's Seventh Framework Programme in the context of the SENSEI project.

References

Barnett, V. and Lewis, T. (1994) *Outliers in Statistical Data*, John Wiley Sons, New York.

Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, J. (2000) 'LOF: identifying density-based local outliers', *ACM SIGMOD Conference on Management of Data*, pp.93–104.

Chandola, V., Banerjee, A. and Kumar, V. (2007) *Outlier detection: a survey*, Technical Report, University of Minnesota, USA.

Davy, M., Desobry, F., Gretton, A. and Doncarli, C. (2006) 'An online support vector machine for abnormal events detection', *Journal of Signal Processing*, Vol. 8, No. 2, pp.52–57.

Gaber, M.M. (2007) 'Data stream processing in sensor networks', in Gama, J. and Gaber, M.M. (Eds): *Learning from Data Streams Processing Techniques in Sensor Network*, Springer, Berlin Heidelberg, pp.41–48.

Hawkins, D.M. (1980) *Identification of Outliers*, Chapman and Hall, London.

Hodge, V.J. and Austin, J. (2003) 'A survey of outlier detection methodologies', *Journal of Artificial Intelligence Review*, Vol. 22, pp.85–126.

IBRL (2004) *Intel Berkeley Research Laboratory*. Available online at: http//db.csail.mit.edu/labdata/labdata.html/ (accessed on 10 December 2008).

Laskov, P., Schafer, C. and Kotenko, I. (2004) 'Intrusion detection in unlabeled data with quarter sphere support vector machines', *Detection of Intrusions and Malware & Vulnerability Assessment*, Dortmund, Germany, pp.71–82.

Knorr, E. and Ng, R. (1998) 'Algorithms for mining distance-based outliers in large data sets', *Journal of Very Large Data Bases*, pp.392–403.

Krishnamachari, B. and Iyengar, S. (2004) 'Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks', *IEEE Transactions on Computers*, Vol. 53, No. 3, pp.241–250.

Nash, S.G. and Sofer, A. (1996) *Linear and Nonlinear Programming*, Vol. 37, Nos. 3–4, McGrawHill.

Rajasegarar, S., Leckie, C., Palaniswami, M. and Bezdek, J.C. (2007) 'Quarter sphere based distributed anomaly detection in wireless sensor networks', *IEEE International Conference on Communications*, Glasgow, England, pp.3864–3869.

Scholkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J. and Williamson, R.C. (2001) 'Estimating the support of a high-dimensional distribution', *Journal of Neural Computation*, Vol. 13, No. 7, pp.1443–1471.

SensorScope (2007) *SensorScope System*. Available online at: http://sensorscope.epfl.ch/index.php/MainPage (accessed on 10 December 2008).

Tan, P.N., Steinback, M. and Kumar, V. (2006) *Introduction to Data Mining*, Addison Wesley.

Tax, D.M.J. and Duin, R.P.W. (2004) 'Support vector data description', *Journal of Machine Learning*, Vol. 54, No. 1, pp.45–56.

Vuran, M.C., Akan, O.B. and Akyildiz, I.F. (2004) 'Spatiotemporal correlation: theory and applications for wireless sensor networks', *Journal of Computer and Telecommunications Networking*, Vol. 45, No. 3, pp.245–259.

Zhang, Y., Meratnia, N. and Havinga, P.J.M. (2008) *Outlier detection techniques for wireless sensor network: a survey*, Technical Report, University of Twente, The Netherlands.