Research Article

Lightweight Anomaly Detection for Wireless Sensor Networks

Pu Cheng\textsuperscript{1,2} and Minghua Zhu\textsuperscript{1}

\textsuperscript{1}Software Engineering Institute, East China Normal University, Shanghai 200062, China
\textsuperscript{2}Software Institute, Henan University, Kaifeng 475002, China

Correspondence should be addressed to Minghua Zhu; mhzhu@sei.ecnu.edu.cn

Received 9 January 2015; Accepted 3 June 2015

Abstract Anomaly detection in wireless sensor networks (WSNs) is critical to ensure the quality of sensor data, secure monitoring, and reliable detection of interesting and critical events. The main challenge of anomaly detection algorithm in WSNs is identifying anomalies with high accuracy while consuming minimal resource of the network. In this paper, two lightweight anomaly detection algorithms LADS and LADQA are proposed for WSNs. Both algorithms utilize the one-class quarter-sphere support vector machine (QSSVM) and convert the linear optimization problem of QSSVM to a sort problem for the reduced computational complexity. Experimental results show that the proposed algorithms can keep the lower computational complexity without reducing the accuracy for anomaly detection, compared to QSSVM.

1. Introduction

Wireless sensor networks (WSNs) have been widely used in various applications including civil and military domains [1]. However, the harsh deployment environment and the constrained capabilities of sensors (energy, CPU, memory, etc.) make WSNs more vulnerable to different types of misbehaviors or anomalies. In WSNs, an anomaly or outlier (these terms are used interchangeably in this paper) is defined as the measurement that significantly deviates from the normal pattern of the sensed data [2]. Sensor data are affected by these anomalies that always correspond to node software or hardware failure, reading errors, unusual events, and malicious attacks. Therefore, it is critical to efficiently and accurately identify anomalies in the sensor data to ensure data quality, secure monitoring, and reliable detection of interesting and critical events.

The context of sensor networks and nature of sensor data make design of an appropriate anomaly detection technique challenging [3]. The constrained environment of a WSN impacts on anomaly detection algorithms. Node constraints on computational power and memory mean that algorithms for anomaly detection should have low computational complexity and occupy little memory space. Moreover, prelabelled data are expensive or difficult to obtain in WSNs. Anomaly detection for WSNs should be able to operate on unlabelled data. In general, the key challenge of anomaly detection algorithm in WSNs is identifying anomalies with high accuracy while consuming minimal resource of the network.

Recently, support vector machines (SVM) in the form of the one-class quarter-sphere (QSSVM) have been used for anomaly detection in WSNs due to their reduced computational complexity and ability to operate on unlabelled data [4, 5]. QSSVM can convert the quadratic optimization problem of one-class SVM to a linear optimization problem. A family of algorithms based on QSSVM are proposed for anomaly detection in WSNs and have shown the potential for anomaly detection. However, the main disadvantage of those algorithms derived from QSSVM is the high computational cost for the solution of a linear programme.

In this paper, we convert the linear optimization problem of QSSVM to a sort problem and propose two lightweight anomaly detection algorithms for WSNs to identify anomalies with high accuracy while consuming minimal resource of the network.

Our paper makes the following contributions:

1. We present a mathematic method to prove that the linear optimization problem of QSSVM can be converted to a sort problem.
(2) Based on the presented method, we propose two lightweight anomaly detection algorithms for WSNs, namely, lightweight anomaly detection algorithm using sort (LADS) and lightweight anomaly detection algorithm using quick select (LADQS). It is shown that our algorithms are equivalent to QSSVM but have lower computational complexity.

(3) The experimental evaluation of the effectiveness and efficiency of the proposed algorithms on real world WSN dataset is presented.

The remainder of this paper is structured as follows. Related works on QSSVM-based anomaly detection models are presented in Section 2. In Section 3, first the principle of QSSVM is described, and then a method to convert the linear optimization problem of QSSVM to a sort problem is discussed. Furthermore, our proposed lightweight anomaly detection algorithms are also explained. Experimental results and performance evaluation are reported in Section 4. Section 5 concludes the paper and suggests some directions for future research.

2. Related Work

Anomaly detection typically makes use of data mining and machine learning techniques to detect abnormal activities in the systems [2]. Anomaly detection techniques for WSNs can be categorized into statistical-based, nearest neighbor-based, clustering-based, classification-based, and spectral decomposition-based approaches [6]. Classification models are important models of data mining and machine learning community in which a classification model is learned using the known training data and used after that to classify the unseen testing data into different types of classes. SVM-based techniques are one of the popular classification-based approaches and have been widely used to detect anomalies due to the advantages of no requirement of an explicit statistical model and prevention from the curse of dimensionality.

In WSNs, prelabelled data are expensive or difficult to obtain. Several one-class SVM-based anomaly detection techniques have been proposed to process the unlabeled data. Their main idea is to use a nonlinear function to map the data vectors collected from the original input space to a higher dimensional space called feature space. Then a decision boundary of normal data is found, which encompasses the majority of data vectors in the feature space. Those data vectors falling outside the normal boundary are classified as anomalous. To this end, Schölkopf et al. have presented a hyperplane-based one-class SVM by fitting a hyperplane from the origin. Those data vectors near the origin are considered as anomalous [7]. Tax and Duin have proposed a hypersphere-based one-class SVM by fitting a hypersphere with a minimum radius [8]. Those data vectors falling outside the hypersphere are considered as anomalous. However, these one-class SVM-based techniques still require solving a quadratic optimization, and that is extremely costly.

In order to reduce expensive computational complexity of the quadratic optimization, Campbell and Bennett have formulated a linear programming approach for the hyperplane-based SVM proposed in [9], which is based on attracting the hyperplane towards the average of the distribution of mapped data vectors. Laskov et al. have extended work in Tax and Duin by proposing a quarter-sphere one-class SVM, which converts the quadratic optimization problem to a linear optimization problem by fitting a hypersphere centered at the origin and consequently reduces computational complexity of learning the normal boundary of data vectors [5, 8]. Based on QSSVM, several distributed outlier detection techniques for WSNs are proposed by Rajasegarar et al. and Zhang et al. [10, 11]. After that, Rajasegarar et al. have further extended work in [12] by proposing a hyperellipsoidal one-class SVM using a linear optimization. However, the solution of a linear optimization, rather than a quadratic, still requires expensive computational complexity due to the fact that the solution of the linear programme is \( O(n^2) \) where \( n \) is the number of data vectors in the training set.

In this paper, we propose two lightweight anomaly detection techniques which convert the linear optimization problem of QSSVM to a sort problem and consequently reduce computational complexity of learning the normal boundary of data vectors.

3. Lightweight Anomaly Detection Algorithm for WSNs

In this section, we first introduce the principles of modeling the one-class quarter-sphere support vector machine (QSSVM) proposed in [5]. After that, we use a mathematical method to prove that the linear optimization problem of QSSVM can be converted to a sort problem and further propose two lightweight anomaly detection algorithms for WSNs.

3.1. Principles of the One-Class Quarter-Sphere SVM

In this section we discuss the principles of QSSVM proposed by Laskov et al. in [5]. They have converted the quadratic optimization problem of the one-class SVM to a linear optimization problem by fixing the center of the quarter-sphere at the origin. The geometry of hypersphere SVM is shown in Figure 1.

Assume that \( n \) data vectors \( x_i \in \mathbb{R}^d, \ i = 1, \ldots, n \) of \( d \) variables in the input space are mapped into the feature space using a certain nonlinear mapping function \( \phi(g) \). The constrained optimization problem of QSSVM can be formalized as follows:

\[
\begin{align*}
\min_{R \in \mathbb{R}, \xi \in \mathbb{R}^n} & \quad R^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i, \\
\text{subject to:} & \quad \|\phi(x_i)\|^2 \leq R^2 + \xi_i, \\
& \quad \xi_i \geq 0, \quad i = 1, \ldots, n,
\end{align*}
\]

(1)

where \( \{\xi_i : i = 1, 2, \ldots, n\} \) are the slack variables that allow some of the data vectors to fall outside the quarter-sphere.
The regularization parameter \( \nu \) is a representation of the number of data vectors that are expected to be anomalies, where \( \nu \in (0, 1) \).

Obtaining the dual form of the optimization problem allows its formulation in terms of dot products of the data vectors in the training set. Using the kernel trick, the dot products \( |\phi(x_i)|^2 \) are replaced by the kernels \( k(x_i, x_j) \). The dual formulation of (1) will become

\[
\min_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^{n} \alpha_i k(x_i, x_i)
\]

subject to:

\[
\begin{align*}
\sum_{i=1}^{n} \alpha_i &= 1, \\
0 &\leq \alpha_i \leq \frac{1}{\nu} \quad \forall i, \quad i = 1, \ldots, n.
\end{align*}
\]

(2)

It can be seen from (2) that optimization problem is stated in terms of a dot product of an image vector with itself; this causes an issue with distance-based kernels, such as the RBF kernel, as the diagonal term \( k(x_i, x_i) \) becomes equal for all the vectors. This can be solved by centering the kernel matrix in feature space where the mean of the image vectors is subtracted from each image vector as follows:

\[
\phi(x_i) = \phi(x_i) - \frac{1}{n} \sum_{i=1}^{n} \phi(x_i).
\]

(3)

There is no explicit vector in feature space that represents the mean; however the dot product \( K_c = \langle \phi(x_i), \phi(x_j) \rangle \) of the centered image vectors can be obtained in terms of the kernel matrix \( K = k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \) using \( K_c = K - \frac{1}{n} K_n - \frac{1}{n} K_{11} \), where \( K_n \) is an \( n \times n \) matrix with all values equal to \( 1/n \) [13]. When the kernel matrix is centered in feature space the norms of the kernels are no longer equal and the terms \( k(x_i, x_i) \) of (2) are replaced by the diagonal elements \( k_c(x_i, x_i) \) of the centered kernel matrix \( K_c \). \( \sqrt{k_c(x_i, x_i)} \) can be considered as the distances between data vectors \( x_i \) and the origin of its centered quarter-sphere in the feature space. Consequently, the dual problem (2) can be solved.

After solving (2) for \( \alpha_i \), the data vectors can be classified as follows. Data vectors with \( \alpha_i = 0 \), which lie inside the hypersphere, are considered as normal and their distances from the origin are smaller than the radius of the quarter-sphere. Data vectors with \( \alpha_i = 1/n \) are considered as anomalies, which lie outside the hypersphere. Data vectors with \( 0 < \alpha_i < 1/n \), which lie on the surface of the hypersphere, are called the border support vectors. Moreover, the minimal radius \( R \) of the hypersphere can be obtained using \( R^2 = k_c(x_i, x_j) \) for any border support vector \( x_i \).

3.2. Lightweight Anomaly Detection Algorithm Using Sort. In the previous section, we introduced the principle of QSSVM which convert the quadratic optimization problem to a linear optimization problem. We are aware that the learning process of QSSVM is to find the minimal radius \( R \) of one-class quarter-sphere for the training set, which can be obtained by the distances between border support vectors \( x_i \) and its origin in the feature space using \( R^2 = k_c(x_i, x_j) \). This process has high computational and memory complexity due to the fact that it requires solving the linear programme and finding data vectors \( x_i \) with \( 0 < \alpha_i < 1/n \), that is, the border support vectors. To reduce the cost of modeling QSSVM, we propose a lightweight anomaly detection algorithm using sort to identify anomalies. Our algorithm can obtain the minimal radius \( R \) of QSSVM, that is, the distance \( \sqrt{k_c(x_i, x_j)} \) between data vector \( x_i \) with \( 0 < \alpha_i < 1/n \) and the origin, by a descending sorted sequence instead of the solution of a linear optimization problem. That is to say, our algorithm will be equivalent to QSSVM but converts the quadratic optimization problem to a sort problem. For doing so, we first fix the data vectors at the origin in the feature space through generation of kernel matrix and the transformation of central kernel matrix like QSSVM [5] and obtain the formulation of (2). Note that the terms \( k(x_i, x_j) \) of (2) have been actually replaced by the diagonal elements \( k_c(x_i, x_i) \) of the centered kernel matrix \( K_c \) as we discussed in the previous subsection. Next we require finding the minimal radius \( R \) from (2). Obviously, the process for calculation of \( R \) only needs to find a distance \( \sqrt{k_c(x_i, x_i)} \) corresponding to \( \alpha_i \) satisfying the formulation of (2) and \( 0 < \alpha_i < 1/n \), instead of solving the linear programme for (2).

In order to find the minimal radius \( R \), the sequence \( \{k_c(x_i, x_j)\} \) is sorted in descending order and becomes \( \{k_c(x_i, x_j)\} = \{k_c(x_1, x_1), k_c(x_2, x_2), \ldots, k_c(x_n, x_n)\} \), where \( k_c(x_1, x_1) \geq k_c(x_2, x_2) \geq \cdots \geq k_c(x_n, x_n) \). For convenience, we replace \( k_c(x_i, x_i) \) with \( d_i \). Consequently, the dual formulation of (2) will be simplified to

\[
\max_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{n} \alpha_i d_i
\]

subject to:

\[
\begin{align*}
\sum_{i=1}^{n} \alpha_i &= 1, \\
0 &\leq \alpha_i \leq \frac{1}{\nu} \quad \forall i, \quad i = 1, \ldots, n.
\end{align*}
\]

(4)
where $d_{1^v} \geq d_{2^v} \geq \cdots \geq d_{n^v}$, $v \in (0, 1)$ is the regularization parameter that represents the fraction of outliers, and $n$ is the number of the data vectors in the training set.

Actually, there exists an implied constraint that $\alpha_j = \alpha_i$ if and only if $d_{i^v} = d_{j^v}$. This implies that the data vectors which have the same distances to origin in the feature space either are normal or not.

Now we prove that the minimal radius $R$ can be computed directly from the descending sequence $\{d_{1^v}, d_{2^v}, \ldots, d_{n^v}\}$ through the parameters $v$ and $n$.

**Theorem 1.** If the values of $\alpha_1, \alpha_2, \ldots, \alpha_n$ are the solution of (4), one has $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$.

**Proof.** Since $\alpha_1, \alpha_2, \ldots, \alpha_n$ is the solution of (4), the value of the objective function attains maximum under the constrained condition and can be denoted as $f_1 = \alpha_1 d_{1^v} + \alpha_2 d_{2^v} + \cdots + \alpha_n d_{n^v}$.

Firstly, we prove that $\alpha_1$ is the maximum of $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ if $d_{1^v}$ is the maximum of $\{d_{1^v}, d_{2^v}, \ldots, d_{n^v}\}$. Assume the contrary that the value of $\alpha_1$ is not maximal if the value of $d_{1^v}$ is maximal. This assumption indicates that there exists $\alpha_j$ such that the value of $\alpha_j$ is the maximal of $\{\alpha_i\}$. We exchange the value of $\alpha_j$ with $\alpha_1$, so the value of the objective function becomes

$$f_2 = \alpha_1 d_{1^v} + \alpha_2 d_{2^v} + \cdots + \alpha_j d_{j^v} + \alpha_{j+1} d_{(j+1)^v} + \alpha_{n} d_{n^v}.$$  

(5)

Subtract $f_1$ from $f_2$ and we have

$$f_2 - f_1 = \left(\alpha_1 d_{1^v} + \alpha_2 d_{2^v} + \cdots + \alpha_j d_{j^v} \right) - \left(\alpha_1 d_{1^v} + \alpha_2 d_{2^v} + \cdots + \alpha_n d_{n^v} \right).$$  

(6)

From the contradictions, we have $d_{1^v} \geq d_{j^v}$ and $\alpha_j \geq \alpha_1$. So we get $f_2 - f_1 \leq 0$. This contradicts the fact that $f_1$ is not the maximum of the objective function. Thus we have derived that $\alpha_1$ is the maximum of $\{\alpha_i\}$ for $i = 1, 2, \ldots, n$.

In a similar way, we can prove that $\alpha_2$ is the second maximum of $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ if $d_{2^v}$ is the second maximum of $\{d_{1^v}, d_{2^v}, \ldots, d_{n^v}\}$. And so, we can prove that $\alpha_j$ is the $j$th maximum of $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ if $d_{j^v}$ is the $j$th maximum of $\{d_{1^v}, d_{2^v}, \ldots, d_{n^v}\}$.

As proved above, it is evident that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$ if the values of $\alpha_1, \alpha_2, \ldots, \alpha_n$ are the solution of (4). This completes the proof.

According to Theorem 1, we can infer Theorems 2 and 3 as follows.

**Theorem 2.** If the values of $\alpha_1, \alpha_2, \ldots, \alpha_n$ are the solution of (4), one has $0 < \alpha_{j+1} < 1/vn$ when $j = [vn]$ and $j \neq vn$.

**Proof.** Since one of the constraint conditions is $0 \leq \alpha_i \leq 1/vn$, Theorem 3 will be proved if we can show that $\alpha_{j+1} \neq 0$ and $\alpha_{j+1} \neq 1/vn$ when $j = [vn]$ and $j \neq vn$. If the assertion were not true, then we would have $\alpha_{j+1} = 0$ or $\alpha_{j+1} = 1/vn$ when $j = [vn]$ and $j \neq vn$.

Recalling Theorem 1, we have $1/vn \geq \alpha_{j+1} \geq \cdots \geq \alpha_n \geq 0$. By the assumption of $\alpha_{j+1} = 0$, it is easy to see that $\alpha_{j+1} > \alpha_{j+2} = \cdots = \alpha_n = 0$. And we have $\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{j} \alpha_i \leq j/vn$. When $j = [vn]$ and $j \neq vn$, the above inequality will become $\sum_{i=1}^{n} \alpha_i < j/vn < 1$. This contradicts the fact of $\sum_{i=1}^{n} \alpha_i = 1$. By the assumption of $\alpha_{j+1} = 1/vn$, it is clear that $\alpha_{j+2} = \cdots = \alpha_n = \alpha_{j+1} = 1/vn$. And we get $\sum_{i=1}^{n} \alpha_i = (j+1)/vn$. When $j = [vn]$ and $j \neq vn$, the inequality will become $\sum_{i=1}^{n} \alpha_i = (j+1)/vn > 1$. This is contrary to the constraint condition of $\sum_{i=1}^{n} \alpha_i = 1$. This completes the proof.

**Theorem 3.** Given that the values of $\alpha_1, \alpha_2, \ldots, \alpha_n$ are the solution of (4), if $j = vn$ and $j = \lfloor vn \rfloor$, one has

$$d_{1^v} = d_{2^v} = \cdots = d_{j^v}$$  

(7)

$$\alpha_1 = \alpha_2 = \cdots = \alpha_n;$$  

otherwise,

$$\alpha_1 = \cdots = \alpha_{j-1} = 1/vn.$$  

(8)

$$\alpha_{j+1} = \cdots = \alpha_n = 0.$$  

**Proof.** In the case when $d_{1^v} = d_{2^v} = \cdots = d_{j^v}$, obviously, we have $\alpha_1 = \alpha_2 = \cdots = \alpha_n$ by the implied constraint that there exists $\alpha_j = \alpha_j$ if and only if $d_{j^v} = d_{j^v}$. In the other cases, let the value $\beta_j$ of $\alpha_j$ be the solution of (4). If the assertion were not true, we would have the value $\beta_j$ of $\alpha_j$ such that $\beta_j \neq 1/vn$ or $\beta_j \neq 0$, where $k \in \{1, \ldots, j\}$ and $l \in \{j+1, \ldots, n\}$. Consequently, the value of objective function becomes $f_1 = \sum_{i=1}^{n} \beta_i d_{i^v}$. When $\alpha_1 = \alpha_2 = \cdots = \alpha_j = 1/vn$ and $\alpha_{j+1} = \cdots = \alpha_n = 0$, where $j = vn$ and $j = \lfloor vn \rfloor$, let the value of objective function be $f_2 = \sum_{i=1}^{n} \alpha_i d_{i^v} = (1/vn) \sum_{i=1}^{j} d_{i^v}$. Since the value $\beta_j$ of $\alpha_j$ is the solution of (4), $f_1$ is the maximal of the objective function and $f_1 \geq f_2$. Subtract $f_1$ from $f_2$ and we have

$$f_2 - f_1 = (1/vn - \beta_{j+1})d_{j+1^v} + \cdots + (1/vn - \beta_j)d_{j^v} - \beta_j d_{j+1^v} - \cdots - \beta_j d_{j^v}.$$  

According to the implied constraint, the equality sign holds if and only if $d_{j+1^v} = \cdots = d_{j^v}$ or $\beta_{j+1} = \cdots = \beta_j = 0$. And we also have $f_2 - f_1 \geq (1/vn - \beta_{j+1})d_{j+1^v} + \cdots + (1/vn - \beta_j)d_{j^v} - \beta_j d_{j+1^v} - \cdots - \beta_j d_{j^v}$. According to the implied constraint, the equality sign holds if and only if $d_{j+1^v} = \cdots = d_{j^v}$ or $\beta_{j+1} = \cdots = \beta_j = 1/vn$ and $\beta_j d_{j+1^v} = \cdots = \beta_j d_{j^v} = 0$. However, this is not the case when $d_{j+1^v} = \cdots = d_{j^v}$ and the definition and condition of $\beta_j$ show that $\beta_{j+1} = 1/vn$ or $\beta_j \neq 0$, where $k \in \{1, \ldots, j\}$ and $l \in \{j+1, \ldots, n\}$. So the equality sign does not hold and we have $f_2 - f_1 > (1/vn - \beta_{j+1})d_{j+1^v} + \cdots + (1/vn - \beta_j)d_{j^v} - \beta_j d_{j+1^v} - \cdots - \beta_j d_{j^v}$. And we get $f_2 - f_1 > (j/vn - \beta_{j+1} - \cdots - \beta_j)d_{j^v}$. From $j = vn$ and $j = \lfloor vn \rfloor$ and the constraint of $\sum_{i=1}^{n} \beta_i = 1$, we obtain $f_2 - f_1 > 0$. This is contrary to $f_1 > f_2$. That is, it completes the proof.
Step 1. Calculate \( k_c(x_i, x_j) \).
Step 2. Sort \( k_c(x_i, x_j) \) in descending order and have \( \{k_c(x_j, x_i)\} \).
Step 3. Calculate \( R = \sqrt{d_{(j+1)'}^c} \) using (9).
Step 4. If \( k_c(x_i, x_j) > R^2 \)
   \( x_i \) is outlier
   Else
   \( x_i \) is normal
End If

\[
R^2 = d_{(j+1)'}^c, \quad (9)
\]

where \( d_{(j+1)'}^c \) is the \((j+1)\)th largest squared distances \( k_c(x_i, x_j) \) between \( x_i \) and the origin of its centered quarter-sphere in the feature space.

Proof. As we know, the constrained optimization problem of QSSVM can be formulated as (2). Note that the terms \( k(x_i, x_j) \) of (2) are replaced by \( k_c(x_i, x_j) \). After calculating the values of \( \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \) in (2), the minimal radius \( R \) of the quarter-sphere in QSSVM can be computed using \( R^2 = k_c(x_i, x_j) \) for any data vector \( x_i \) with \( 0 < \alpha_i < 1/\nu \). Given \( d_i = k_c(x_i, x_j) \), when \( \{k_c(x_i, x_j)\} \) in (2) is sorted in descending order, the optimization problem of (2) becomes the formulation of (4).

According to Theorems 2 and 3, the minimal radius \( R \) can be obtained in two cases. In the first case when \( j = [\nu n] \) and \( j \neq \nu n \), we have \( 0 < \alpha_{(j+1)'} < 1/\nu \) from Theorem 3 and \( R^2 = d_{(j+1)'}^c \). In the second case when \( j = [\nu n] \) and \( j = \nu n \), we have \( \alpha_{1'} = \alpha_{2'} = \cdots = \alpha_{\nu'} = 0 \) or \( \alpha_{1'} = \alpha_{2'} = \cdots = \alpha_{\nu'} = 1/\nu \). After calculating the values of \( \{\alpha_{(j+1)'} \} \) in (2), the minimal radius \( R \) of the quarter-sphere in QSSVM can be computed as \( R^2 = d_{(j+1)'}^c \). The fact of \( \alpha_{1'} = \alpha_{2'} = \cdots = \alpha_{\nu'} = 0 \) means that we get \( 0 < \alpha_{k} < 1/\nu \). By the decision rule of QSSVM, the vectors with \( \alpha_j = 0 \) or the vectors with \( \alpha_j = 1/\nu \) are considered as normal or outliers, respectively. And their squared distances to origin are, respectively, not larger or larger than \( d_{(j+1)'}^c \), which means the largest squared distance between the normal vectors and the origin. Therefore the minimal radius \( R \) can be treated as \( \sqrt{d_{(j+1)'}^c} \). Moreover, we have \( R^2 = d_{(j+1)'}^c \) in both cases. This completes the proof. \( \square \)

From Theorem 4, the minimal radius \( R \) of the quarter-sphere in QSSVM can be obtained by a descending sorted sequence using (9). So the linear optimization problem of QSSVM is converted to a sort problem of \( \{k_c(x_i, x_j)\} \). Now, we propose a lightweight anomaly detection algorithm using sorting (LADS), which is equivalent to the QSSVM. The detail of LADS is described as follows.

After the generation of kernel matrix and the transformation of central kernel matrix, the data vectors \( x_i \) are centered at the origin in the feature space the same as QSSVM. These squared distances to the origin are sorted in descending order and form a descending sequence \( \{k_c(x_j, x_i)\} \), where \( k_c(x_j, x_i) \geq \cdots \geq k_c(x_m, x_i) \). The \((j+1)\)th largest squared distance \( k_c(x_j, x_i) \) is chosen from the descending sequence as the minimal radius \( R \) by (9). The data vectors can be classified depending on \( R \). The data vectors whose distances to origin are not larger than \( R \) are considered as normal. Otherwise, the data vectors whose distances to origin are larger than \( R \) are considered as outliers.

Now, our algorithm can be described in Algorithm 1.

**Algorithm 1.**

3.3. Lightweight Anomaly Detection Algorithm Using Quick Select. As discussed in the previous section, the minimal radius \( R \) of the quarter-sphere in QSSVM can be obtained from the descending sequence \( \{k_c(x_j, x_i)\} \). The computational complexity for obtaining the descending sequence is \( O(n \log_2 n) \). In fact, the process for the descending sequence is unnecessary, because \( R \) can be determined by (9) if the \([\nu n] + 1\)th largest element of the original sequence \( \{k_c(x_j, x_i)\} \) can be found. So we propose an anomaly detection method LADQS based on Quickselct algorithm [14] to find the \([\nu n] + 1\)th largest distance to the origin, that is, minimal radius \( R \). In computer science, Quickselct is a selection algorithm to find the \( k \)th largest element in an unordered list. Quickselct uses the same overall approach as quicksort [14], choosing one element as a pivot and partitioning the data in two based on the pivot, accordingly as less than or greater than the pivot. However, instead of recursing into both sides, as in quicksort, Quickselct only recurses into one side—the side with the element it is searching for. This reduces the average computational complexity from \( O(n \log_2 n) \) to \( O(n) \). The pseudocode of LADQS algorithm is shown in Algorithm 2.

4. Experimental Results and Evaluation

This section specifies the performance evaluation of our two techniques compared to QSSVM. In our experiments, we have used real data gathered at the Grand-Saint-Bernard [15], which is similar to the one used in [16]. For simulation, we use Matlab to implement our algorithms and QSSVM on a single node in a WSN. For fairness, we use the average of the tests operated on 7 different nodes, respectively, as the experimental results.
function LADQS()
    let \( \{x_i \in \mathbb{R}^d, i = 1, \ldots, n\} \) be the data vectors in the training set;
    calculate the distance \( k(x_i, x) \) to origin for each data vector \( x_i \);
    obtain \( A = \{k(x_i, x), i = 1, \ldots, n\} \);
    return \( R = \text{QuickSelect}(A, 1, n, \lfloor V n \rfloor + 1) \)
    // Returns the \( n \)th largest element of list.

function QuickSelect(list, \( l, r, n \))
    if \( l = r \) // If the list contains only one element
        return list[\( l \)] // Return that element
    pivotIndex = \( l + \lfloor \text{rand()} \times (r - l + 1) \rfloor \)
    // randomly select a pivotIndex between \( l \) and \( r \)
    pivotIndex = \text{QuickPartition}(list, \( l, r, \text{pivotIndex} \))
    if \( n = \text{pivotIndex} \)
        return list[\( n \)]
    elseif \( n < \text{pivotIndex} \)
        return QuickSelect(list, \( l, \text{pivotIndex} - 1, n \))
    else
        return QuickSelect(list, \( \text{pivotIndex} + 1, r, n \))

Algorithm 2: The pseudocode of LADQS algorithm.

4.1. Experimental Datasets. The real data are collected from a closed neighborhood from a WSN deployed in Grand-Saint-Bernard as shown in Figure 2. The closed neighborhood consists of node 2 and its 6 spatially neighboring nodes, namely, nodes 3, 4, 8, 12, 20, and 14. In our simulations, we test the real data collected during the period of 6 am–6 pm on September 20, 2007, with two attributes: ambient temperature and relative humidity for each sensor measurement. The data is preprocessed and normalized to the range \([0, 1]\). The number of anomalous data is about 10% of normal data. Measurements are labeled depending on the degree of dissimilarity between one another.

4.2. Experimental Results and Evaluation. We choose radial basis function (RBF) kernel function to generate kernel matrices. RBF kernel function \( k_{\text{RBF}} = \exp(-\|x_1 - x_2\|^2 / \sigma^2) \), where \( \sigma \) is the width parameter of the kernel function. And the kernel width parameter \( \sigma \) is set to 0.25 in our experiments.

We have examined the effect of the regularisation parameter \( \nu \) for our two anomaly detection algorithms and QSSVM. \( \nu \) represents the fraction of anomalies in training set and we have varied it in the range from 0.01 to 0.25 in intervals of 0.03. And we also have examined the training time for the three algorithms.

Figures 3 and 4 show that the detection rate and the false alarm rate obtained for our algorithm LADS use RBF kernel function for real data, respectively. As we discussed in the previous section, LADS, LADQS, and QSSVM have the same principle of data classification but have difference in the way of finding the minimal radius \( R \). This means the LADS...
behaves in a similar manner to LADQS. Therefore, results of LADQS and QSSVM have been omitted.

Figure 5 shows the training time elapsing for the three algorithms. We can see that the training time of our algorithms LADS and LADQS is significantly less than that of QSSVM. It indicates that our two algorithms have less time and lower computational complexity, compared to QSSVM.

Simulation results show that our two algorithms LADS and LADQS have the lower computational complexity without reducing the accuracy for anomaly detection, compared to QSSVM.

Computational complexity of our techniques is presented in Table 1, where $n$ denotes the number of data in the training sets, $d$ represents the dimensionality of the measurement, $\nu$ represents the fraction of anomalies in the training set, and $O(L)$ represents the computational complexity of solving a linear optimization problem.

| Techniques | Computational complexity |
|------------|--------------------------|
| QSSVM      | $O(L)$                   |
| LADS       | $O(n \log n)$            |
| LADQS      | $O(n)$                   |

5. Conclusion

In this paper we propose two lightweight anomaly detection algorithms for WSNs, LADS and LADQS. Both algorithms are based on QSSVM but convert the linear optimization problem of QSSVM to a sort problem. Simulation results show that our algorithms reduce the computational complexity while achieving the same accuracy for anomaly detection. Our future research includes selecting the optimal parameters for $\nu$ and implementing our algorithms on multiple sensor nodes in real-life.

Conflict of Interests

The authors declare no conflict of interests.

Acknowledgment

The work is supported in part by National Science and Technology Major Project under Grant no. 2011ZX03005-002.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4, pp. 393–422, 2002.
[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," ACM Computing Surveys, vol. 41, no. 3, article 15, 2009.
[3] Y. Zhang, N. Meratnia, and P. Havinga, “An online outlier detection technique for wireless sensor networks using unsupervised quarter-sphere support vector machine,” in Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP’08), pp. 151–156, December 2008.
[4] C. O’Reilly, A. Gluhak, M. Imran, and S. Rajasegarar, "Online anomaly rate parameter tracking for anomaly detection in wireless sensor networks," in Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’12), pp. 191–199, Seoul, Korea, June 2012.
[5] P. Laskov, C. Schafer, and I. Kotenko, “Intrusion detection in unlabeled data with quarter sphere support vector machines,” in Detection of Intrusions and Malware & Vulnerability Assessment, Dortmund, 2004.
[6] Y. Zhang, N. Meratnia, and P. J. M. Havinga, “Outlier detection techniques for wireless sensor network: a survey,” Tech. Rep., University of Twente, 2008.

[7] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Journal of Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[8] D. M. J. Tax and R. P. W. Duin, “Support vector data description,” *Journal of Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[9] C. Campbell and K. P. Bennett, “A linear programming approach to novelty detection,” in *Advances in Neural Information Processing Systems*, vol. 14, pp. 395–401, 2001.

[10] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, “Quarter sphere based distributed anomaly detection in wireless sensor networks,” in *Proceedings of the IEEE International Conference on Communications (ICC ’07)*, pp. 3864–3869, June 2007.

[11] Y. Zhang, N. Meratnia, and P. J. M. Havinga, “Ensuring high sensor data quality through use of online outlier detection techniques,” *International Journal of Sensor Networks*, vol. 7, no. 3, pp. 141–151, 2010.

[12] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, “Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 518–533, 2010.

[13] B. Schölkopf, A. J. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[14] C. A. R. Hoare, “Algorithm 65: find,” *Communications of the ACM*, vol. 4, no. 7, pp. 321–322, 1961.

[15] http://sensorscope.epfl.ch/index.php/MainPage.

[16] Y. Zhang, N. Meratnia, and P. Havinga, “Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks,” in *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (WAINA ’09)*, pp. 990–995, May 2009.
