# Distributed and Efficient One-Class Outliers Detection Classifier in Wireless Sensors Networks

Oussama Ghorbel, Mohamed Wassim Jmal, Mohamed Abid, Hichem Snoussi

# Distributed and Efficient One-Class Outliers Detection Classifier in Wireless Sensors Networks

Oussama Ghorbel[1], Mohamed Wassim Jmal[1], Mohamed Abid[1], Hichem Snoussi[2]

[1] National Engineers School of Sfax, CES Research Unit, Sfax University, Tunisia
[2]University of Technology of Troyes, LM2S Research Unit, Troyes, France

oussama.ghorbel@ceslab.org, wassim.jmal@gmail.com, mohamed.abid@enis.rnu.tn, hichem.snoussi@utt.fr

**Abstract.** In the data mining literature, many outlier detection models can be found. However, these models are not suitable for the energy constrained WSNs because they assumed the whole data is available in a central location for further analysis. In this paper, we propose Distributed and Efficient One-class Outliers Detection Classifier (DEOODC) based on Mahalanobis Kernel used for outlier detection in wireless sensor networks (WSNs). For this case, the task amounts to create a useful model based on KPCA to recognize data as normal or outliers. Recently, Kernel Principal component analysis (KPCA) has used for nonlinear case which can extract higher order statistics. Kernel PCA (KPCA) mapping the data onto another feature space and using nonlinear function. On account of the attractive capability, KPCA-based methods have been extensively investigated, and have showed excellent performance. Within this setting, we propose Kernel Principal Component Analysis based Mahalanobis kernel as a new outlier detection method using Mahalanobis distance to implicitly calculate the mapping of the data points in the feature space so that we can separate outlier points from normal pattern of data distribution. The use of KPCA based Mahalanobis kernel on real word data obtained from Intel Berkeley are reported showing that the proposed method performs better in finding outliers in wireless sensor networks when compared to the One-Class SVM detection approach. All computation are done in the original space, thus saving computing time using Mahalanobis Kernel.

**Keywords:** Wireless Sensor Networks, Outlier Detection, Kernel methods, Mahalanobis kernel, Kernel Principal Component Analysis (KPCA).

## 1 Introduction

With the increasing advances of digital electronics and wireless communications, in the past decade a new breed of tiny embedded systems known as wireless sensor nodes has emerged. These wireless sensor nodes are equipped with sensing, processing, wireless communication, and more recently actuation capability. They usually are densely deployed in a wide geographical area and continuously measure various parameters (e.g. ambient temperature, relative humidity, soil moisture, wind speed) of the physical world. A large collection of these sensor nodes forms a wireless sensor network (WSN) [1]. However, raw sensor observations collected from sensor

nodes often have low data quality and reliability due to the limited capability of sensor nodes in terms of energy, memory, computational power, bandwidth, dynamic nature of network, and harshness of the deployment environment. Use of low quality sensor data in any data analysis and decision making process limits the possibilities for reliable real-time situation-awareness. Wireless sensor networks are widely used and have gained attention in various fields including traffic control, health care, precision agriculture, etc [2, 3]. KPCA has been used in several applications, such as voice recognition, image segmentation, face detection, feature extraction, data denoising and etc. Most WSN's applications require precise and accurate data to provide reliable information to the end user. Although the importance of information quality provided from WSNs, collected sensor data may be of low quality and reliability due to the low cost nature and harsh deployments of WSNs [4]. To ensure the quality of sensor measurements, outlier detection methods allow cleaning and refinement of collected data and let providing the most useful information to end users, while maintaining low energy consumption and preserve high computational efforts due to the limited energy resources of sensor nodes. To detect outliers, a detection model is built upon historical data structure of WSN. This model should be able to detect outliers among new observations with good precision [5].

By means of an alternative way of computing the principal axes through the use of inner product evaluations, Principal Component Analysis has been extended to a kernel-based PCA. The use of non-linear dimensionality reduction to expand in many applications as recent research has shown that kernel principal component analysis (KPCA) can be expected to work well as a pre-processing device for pattern recognition. The use of KPCA is a new field on wireless sensor networks (WSN) which are composed of interconnected micro-sensors that are able to collect, store, process and transmit data over the wireless channel. KPCA has found a new field which is integrated in application of novelty detection.

Our work is a comparative study of One Class outlier detection method in wireless sensor networks. So, the main contribution of this paper is the uses of Mahalanobis kernel based KPCA for outlier detection method in wireless sensor networks. To identify outliers, we use Mahalanobis distance induced feature subspace spanned by principal components as obtained by Kernel PCA. If the distance of a new data point is above a prefixed threshold, the observation is considered as an outlier, which is also established experimentally. It assumes that the principal subspace represents the normal data. The model is tested on real data from Intel Berkeley. The obtained results are competitive and the proposed method can achieve high detection rate with the lowest false alarm rate.

The remainder of this article is organized as follows. Section 2, present the related work for KPCA. Section 3, describes outliers detection and its different category in wireless sensor networks. Section 4, describes adopted method. Section 5, showcases the obtained experimental results, and section 6 concludes and summarizes the main outcomes of the paper.

## 2 Related Works

Principal component analysis (PCA), first introduced by Hotelling [27], is a well-established dimension-reduction method. It replaces a set of correlated variables by a smaller set of uncorrelated linear combinations of those variables, such that these linear combinations explain most of the total variance. It is also a way of identifying inherent patterns, relations, regularities, or structure in the data. Because such patterns are difficult to detect in high-dimensional data, PCA can be a powerful tool. As a linear statistical technique, PCA cannot accurately describe all types of structures in a

given dataset, specially nonlinear structures. Kernel principal component analysis (KPCA) has recently been proposed as a nonlinear extension of PCA (Scholkopf, Smola, and Muller, 1998). See also Scholkopf and Smola (2002).
Kernel based principle components analysis is a non linear PCA created using the kernel trick. KPCA maps the original inputs into a high dimensional feature space using a kernel method [6].
Mathematically, we transform the current features into a high-dimensional space and the calculate eigenvectors in this space. We ignore the vectors with really low eigen-values and then do learning in this transformed space. KPCA is computationally intensive and takes a lot more time compared to PCA. The reason being that the number of training data points in KPCA is much higher than PCA. So number of principle components that need to be estimated is also much larger. The KPCA method has exhibited superior performance compared to linear PC analysis method in processing nonlinear systems [7], [8]. The detail introduction of the basic KPCA can be viewed in [7], and [9]. Kernel PCA (KPCA), as presented by Scholkopf et al., is a technique for nonlinear dimension reduction of data with an underlying nonlinear spatial structure. A key insight behind KPCA is to transform the input data into a higher-dimensional feature space. The feature space is constructed such that a nonlinear operation can be applied in the input space by applying a linear operation in the feature space.

Lee et al [19] and Cho et al [20] used kernel PCA with Gaussian kernel for fault detection and identification of process monitoring in the field of chemical engineering. Franc and Hlavac [24] used the greedy KPCA which essentially works by *filtering* or *sampling* the original training set for a lesser but representative subset of vectors which span approximately the same subspace as the subspace in the kernel induced feature space spanned by the training set. The training set is then projected onto the span of the lesser subset, where PCA is carried out. Other sampling-based methods exist [22, 25]. Current KPCA reconstruction methods equally weigh all the features; it is impossible to weigh the importance of some features over the others. Some other existing methods also have limitations. Some works only considers robustness of the principal subspace; they do not address robust fitting. Lu et al present an iterative approach to handle outliers in training data. At each iteration, the KPCA model is built, and the data points that have the highest reconstruction errors are regarded as outliers and discarded from the training set. However, this approach does not handle intra-sample outliers. Several other approaches also considering Berar et al propose to use KPCA with polynomial kernels to handle missing data. However, it is not clear how to extend this approach to other kernels. Furthermore, with polynomial kernels of high degree, the objective function is hard to optimize. Sanguinetti & Lawrence propose an elegant framework to handle missing data. The framework is based on the probabilistic interpretation inherited from Probabilistic PCA. However, Sanguinetti & Lawrence do not address the problem of outliers.

We present also the OCSVM method used on our work which belongs a family of classifiers based on the SVM of [31] that constructs a hypothesis which estimates the support of the normal distribution, the decision surface constructed separates the normal and anomaly data vectors in the data set and is able to classify unseen data with a decision function. The one-class SVM has two formulations; the hyperplane version of [32] operates by denoting the origin as the only member of the anomalous class, and then separating the majority of the data from the origin with a hyperplane. The alternative formulation is the hypersphere [33] where normal data is enclosed in a hypersphere with those data points outside the hypersphere being considered as anomalies.

This paper presents a novel cost function based on Mahalanobis kernel using Mahalanobis distance that unifies the treatment of outliers in KPCA. Experiments show that our algorithm outperforms existing approaches.

## 3 Outlier Detection in Wireless Sensor Networks

Sensor data is highly susceptible to various sources of errors such as changing environmental conditions which may produce noise or noise from other sources [10]. These noises can severely affect data transmitted to central base. These abnormal data are called outliers. It is used for finding errors, noise, missing values, inconsistent data, or duplicate data. This abnormal value may affect the quality of data and reduces the system performance. There are three sources of outliers occurred in WSNs: errors, events, and malicious attacks as described above. The use of Outlier detection technique is very important in several real life applications, such as, environmental monitoring, health and medical monitoring, industrial monitoring, surveillance monitors and target tracking [11].

### 3.1 Errors

An error refers to a noise-related measurement or data coming from a faulty sensor. Outliers caused by errors may occur frequently, while outliers caused by events tend to have extremely smaller probability of occurrence. Erroneous data is normally represented as an arbitrary change and is extremely different from the rest of the data

### 3.2 Events

An event is defined as a particular phenomenon that changes the real-world state, e.g., forest fire, chemical spill, air pollution, etc. This sort of outlier normally lasts for a relatively long period of time and changes historical pattern of sensor data. However, faulty sensors may also generate similar long segmental outliers as events and therefore it is hard to distinguish the two different outlier sources only by examining one sensing series of a node itself.

In wireless sensor networks, the sensors have low cost and low energy, so to improve the quality and performance, the better solution is to use outlier detection technique. Evaluation of an outlier detection technique for WSNs depends on whether it can satisfy the mining accuracy requirements while maintaining the resource consumptions of WSNs to a minimum. Outlier detection techniques are required to maintain a high detection rate while keeping the false alarm rate (number of normal data that are incorrectly considered as outliers) low [12, 13]. A receiver operating characteristic (ROC) curves usually is used to represent the trade-off between the detection rate and false alarm rate. For the problem, we can summarize many problems in detection of outliers in WSNs as follows:

- High communication cost
- Modeling normal objects and outliers effectively
- Application specific outlier detection
- Identifying outlier source
- Distributed data
- Communication failures frequently
- Dynamic network topology.

## 4 Proposed Model

A sensor network consist a collection of sensor that can measure characteristics of their local environment from real world physical phenomenon. It performs certain computation, and transmits the collected data samples to base station. Then it is partitioned onto groups or clusters. Each group consists of a cluster head and a number of members. Nodes which belong to the same cluster are geographically close and monitoring generally similar phenomenon (Fig 1). In this work, we will not take into consideration clustering details. We assume that the network is pre-partitioned and the clusters are predefined: every cluster is defined by his cluster head and members.

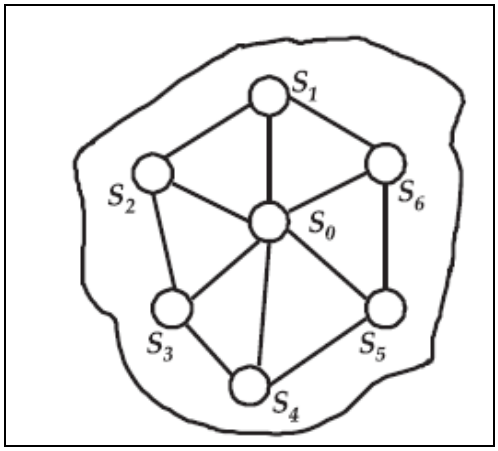


**Fig 1.** Example of a closed neighborhood $N(S_i)$ of the distributed sensor node $S_i$

A wireless sensor network consists of several sensors nodes which collect data samples from real world physical phenomenon. Let's consider a set of *m* sensor nodes measuring each one a multi-real valued attribute at each time instant where $X=(X_1,...,X_m)$ is an m-dimensional random variable [21]. To detect outlier, we present our methodology described in the figure below:

We first propose the Distributed and Efficient One-class Outliers Detection Classifier (DEOODC) based on Mahalanobis Kernel used for outlier detection in wireless sensor networks (WSNs).

Our DEOODC algorithm has some advantages such as lower training time, lower classification time, and lower memory requirements. So, the aims of our algorithm are presented in the following points:

- In DEOODC, the training is conducted using only one class which is the normal class since we do not have labeled training data that contains anomalies as the labeling is difficult and costly.
- The result of applying Automated Cluster Discovery Threshold (ACDT) procedure is only one threshold that separates the normal data from anomalies. This feature is very important for online detection in sensors because in the online testing phase it use only one value.
- In our proposed algorithm, the clustering threshold ($Cl_{th}$) in the ACDT procedure varies according to the WSN application. Our experimental results reveal that ($Cl_{th}$) is in the range [1 2] for IBRL dataset.

Our DEOODC model, like other classification models, has two main phases which are training phase and testing phase. We implement the model in each sensor node locally. The following subsections explain each phase in some details.

### 4.1 Training Phase (offline)

The normal data measurements in the training phase are collected at each sensor node to build the normal model. This latter will be used in testing phase for real time outlier detection. The procedure used to build the normal model is described after.
In this step, we present the training procedure. First, the collected normal data measurements $DM_{tr}$ are centered and normalized by the mean ( $\rho$ ) and standard deviation . Second, PCA is applied on the normalized measurements to obtain the Eigenvector Matrix (EV), and their corresponding eigenvalues $(ev_i)$ . Finally, the projection of each data measurement in the training data $(PDM_i)$ on the new PC space is calculated by Eq: (1):

$$PDM_i = DM_{tr}(i) * EV(i) \qquad (1)$$

Our proposed training phase of EOODC model (offline) is described as follow:
We start by normalize the normal training data. Then, apply the PCA on the training data and obtain the Eigenvector Matrix (EV), eigenvalues $(ev_i)$ and the training data scores in the PC space $(PDM_i)$ . Then, select the number of PCs suitable for the application. After that, calculate the dissimilarity measure $Diss_{tr}$ using the training data measurement scores $(PDM_i)$ and their corresponding eigenvalues $(ev_i)$ . Finally, apply the ACTD procedure to find the thresholds vector $(TV_i)$ that separates between the different classes of data and will be used in testing phase.
The dissimilarity measure $Diss_{tr}$ is calculated for each data measurement in the training set using Eq (2) and the number of PCs suitable for the application is chosen.

$$Diss_{Tr} = \sum \frac{PDM_i^2}{\lambda_i} \qquad (2)$$

To find the threshold vector $(TV_i)$ , we apply the ACTD procedure [26] which will be used for classifying each real time measurement as normal or outlier. For the proposed EOODC, it is important to know that the vector $(TV_i)$ contains a maximum of 2 values and one threshold value depending on the comparison threshold ( $Cl_{th}$ ).

### 4.2 Testing Phase (Online)

For every new data measurement, in online method, collected at each sensor node is tested using the normal model built in the training phase of that particular node. This normal model is composed of the normalization parameters of the training measurements (mean $(\rho)$ , standard deviation $(Std)$ ), Eigenvector Matrix (EV), eigenvalues $(ev_i)$ and the threshold vector $(TV_i)$ . Each measurement is classified as

normal or outlier based on the comparison of its projection on the PC space with the threshold computed in the training phase. The procedure used to classify each new measurement in real time is described after.
The new data measurement is first normalized and centered using the same normalization parameters computed from training measurements in the training phase. Then, the projection score of new measurement on the PC space is calculated using the normal model parameters as in Eq. (3):

$$PDM_i = DM_{test} * EV \qquad (3)$$

Our proposed testing phase of DEOODC model (online) is described as follow:
We start by normalize the real time data measurement using the same normalization parameters computed from the training data (Mean and Std). Then, calculate the $D_{test}$ measure of the real time threshold in the same way of training phase. So, calculate the dissimilarity measure $Diss_{tr}$ using the training data measurement scores $(PDM_i)$ and their corresponding eigenvalues $(ev_i)$. Finally, compare $D_{test}$ value with each value in the threshold value and assign the class index (i) that satisfies:

$$D_{test} > TH_i \text{ ---> } Outlier$$

$$0 \text{ ---> } Normal$$

After that, the dissimilarity measure $D_{test}$ for the new measurement is computed using Eq. (4).

$$D_{Test} = \sum \frac{PDM^2}{\lambda} \qquad (4)$$

Finally, the $D_{test}$ value is compared with the threshold value stored in the node and assigned the appropriate class (either normal or outlier) to the measurement if its $D_{test}$ is smaller or greater than the threshold respectively.

### 4.3 Mahalanobis Kernel

In literature, many types of kernels were employed in the nonlinear transformation of data points (polynomial kernel, sigmoid kernel, etc...) but as we know, Mahalanobis kernel was not used yet in the field of wireless sensor networks. The Mahalanobis kernel (MK) is defined as:

$$K(x_i, x_j) = \exp\left(\frac{-1}{2\sigma^2}(x_i - x_j)^T Q^{-1}(x_i - x_j)\right) \qquad (5)$$

Transformation results of such a kernel are similar to those of a density estimator as it gives a weighted value $w_i$ for every sample $x_i$ of input space. This weighting is not defined for each variable separately although some variables may be more relevant than others in the practice [14]. Let $\{x_1,...,x_N\}$ be a dataset composed of $N$ data points of dimension $m$ , we define the data center $c$ and the covariance matrix $Q$ :

$$c = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (6)$$

$$Q = \frac{1}{N}\sum_{i=1}^{N} (x_i - c)(x_i - c)^T \qquad (7)$$

The Mahalanobis distance between a point and the center is defined as:

$$d(x) = \sqrt{(x-c)^T Q^{-1} (x-c)} \qquad (8)$$

We define the Mahalanobis kernel function as follow, where $H$ is a positive semi definite matrix:

$$A(x, x') = \exp(-(x - x')^T H (x - x')) \qquad (9)$$

In this case, the Mahalanobis distance is calculated between every data point pair $x$ and $x'$. The Mahalanobis kernel is an extension of the RBF kernel when $H = \gamma I$ with $\gamma > 0$ is a parameter that controls the depth of the kernel and $I$ is the identity matrix. In practice, the Mahalanobis kernel (MK) is calculated only for one class:

$$A(x, x') = \exp(-\frac{\delta}{m}(x - x')^T Q^{-1}(x - x')) \qquad (10)$$

Where $\delta > 0$ is a scale factor to control the Mahalanobis distance.
The MK kernel differs from the Gaussian kernel in the fact that for every dimension of the input space data it defines a specific depth value or weight. This makes the calculated decision boundary has a non-spherical shape relative to the center of data points. Using kernel PCA in a learning task has to be well carried out. Choosing the better parameters is important in order to establish the best model with higher accuracy and lower false alarm rate. The outlier detection method of kernel PCA depends generally on kernel type and kernel parameters. In this work, Mahalanobis kernel given by (10) is chosen to resolve the nonlinearity of data distribution. This type of kernel depends on kernel width and number of principal components [23]. We present below the pseudo code algorithm of the training phase and the pseudo code algorithm of the detection phase.

### 4.4 Outlier detection metric

Outlier detection is aim to detect the outlier based on Detection rate and False Alarm Rate described on following equation:

$$Detection\ Rate = \frac{Number\ of\ correctly\ classified\ instances}{total\ number\ of\ instances} * 100\%$$

$$False\ Alarm\ Rate = \frac{Number\ of\ incorrectly\ classified\ instances}{total\ number\ of\ instances} * 100\%$$

Detection Rate: It is defined as the ratio between the numbers of correctly classified instances to the total number of instances.
False Alarm Rate: It is defined as the ratio between the numbers of incorrectly classified instances to the total number of instances.

# 5 Experimental results

## 5.1 Datasets

To validate the proposed models, some data samples were extracted from three WSN deployments which represent static and dynamic environments. The next subsections introduce the datasets and explain the data labeling procedure. The datasets that are used in this paper are extracted from the following WSN deployments:

- *Intel Berkeley Research Lab (IBRL):* IBRL dataset [16] was collected from the WSN deployed at Intel Berkeley Research Laboratory, University of Berkeley. The network consists of 54 Mica2Dot sensor nodes and was deployed in the period of 30 days from 15/04/2004 until 14/05/2004.

- *Grand St. Bernard (GStB):* GStB dataset [17] is one of sensorscope project deployment dataset was gathered using WSN deployment at the Grand St. Bernard pass that is located between Italy and Switzerland. The network is formed of 23 sensors that record metrological environmental data that include temperature and humidity.

- *Sensorscope Lausanne Urban Canopy Experiment (LUCE):* LUCE dataset [18] was collected by a sensorscope project in the École Polytechnique Fédérale de Lausanne (EFPL) campus between July 2006 and May 2007. The measurement system was based on a WSN of 110 sensor nodes deployed on the EPFL campus to measure key environment quantities which include; ambient temperature, surface temperature, and relative humidity.

## 5.2 DEOODC and OCSVM: Comparative Study

This section specifies the performance evaluation of our technique based DEOODC using Mahalanobis kernel and one class SVM. In our experiments, we have used a real data gathered from a deployment of WSN in the Intel Berkeley Research Laboratory, Grand St. Bernard and Sensorscope Lausanne Urban Canopy Experiment. We simulate our protocol both in Matlab and consider a closed neighborhood as shown in Figure 2, which is centered at a node with its 6 spatially neighboring nodes. Here, we use Intel CPU (centrino 2) with the MATLAB version R2009a. Mahalanobis kernel is used recently in the field of WSN, specially based outlier detection, was introduced in several works. Kernel PCA performance was showcased in comparison to other established kernel-based methods [17]. To compute the Kernel PCA transform of a set of test patterns, this approach chooses a training set and a suitable projection dimensionality *p*, and, finally, computes the Mahalanobis distance (MD) for each of these test patterns. Given the projection dimensionality *p*, outliers are identified as data points, whose MD exceeds an appropriately established threshold value $TH$ . Our method has been tested on real data as seen in Table 1.

**Table 1.** DEOODC and OCSVM on the real world Datasets.

| | DEOODC | OCSVM |
|---|---|---|
| **Intel Berkeley (IBRL)** | **0.9912** | 0.9783 |
| | 0.9635 | **0.9743** |
| | **0.9727** | 0.6152 |
| | 0.9551 | **0.9642** |
| | **0.9760** | 0.6396 |
| **Grand-St-Bernard (GStB)** | **0.9891** | 0.7528 |
| | **0.9752** | 0.8457 |
| | 0.9675 | **0.9732** |
| | **0.9686** | 0.8593 |
| | 0.9841 | **0.9876** |
| **Sensorscope (LUCE)** | **0.9837** | 0.9641 |
| | 0.9206 | **0.9360** |
| | **0.9172** | 0.7533 |
| | **0.8336** | 0.7997 |
| | 0.9611 | **0.9673** |

When comparing the results given on our experimentation by DEOODC and OCSVM, we see that using Mahalanobis distance is more beneficial to detect outliers. The value presented in the in bold represent the best value compared to the other values. For example, in IBRL dataset, the bold value (99.12) represents the percentage of accuracy using in MD which is the best one compared to OCSVM. Then, this comparison reveals that OCSVM may not be an effective measure of deviation from normalcy, when compared to using the DEOODC. Thus, it does not satisfactorily fit the normal data because many potential outliers would not be detected. So, our proposed method has an important advantage compared to OCSVM that detects perfectly the outliers as observed in our experiments and as mentioned by the table. Then, it is clear that DEOODC is more sensitive to the detection of FPR and DR (as shown in table 2) than OCSVM. However, it seems to capture much better the overall structure of the normal data.

**Table 2.** Detection rate and false alarm based DEOODC on IBRL real dataset.

| | **Nodes** | | | | | |
|---|---|---|---|---|---|---|
| | N25 | N28 | N29 | N31 | N32 | Average |
| **DR (%)** | 100 | 98 | 95 | 97 | 100 | 98 |
| **FPR (%)** | 14 | 0 | 8 | 1 | 0 | 4.6 |

Based on the following figures (Fig 2, Fig 3 and Fig 4), we presents a comparison between DEOODC and OCSVM. The ROC curve shows that DEOODC based Mahalanobis Kernel are much better than that of OCSVM in terms of outliers detection varied by sigma in our experiments. After the following figures we see that Mahalanobis kernel is more efficient either by simulation on Matlab or on real dataset in Wireless Sensors Networks.

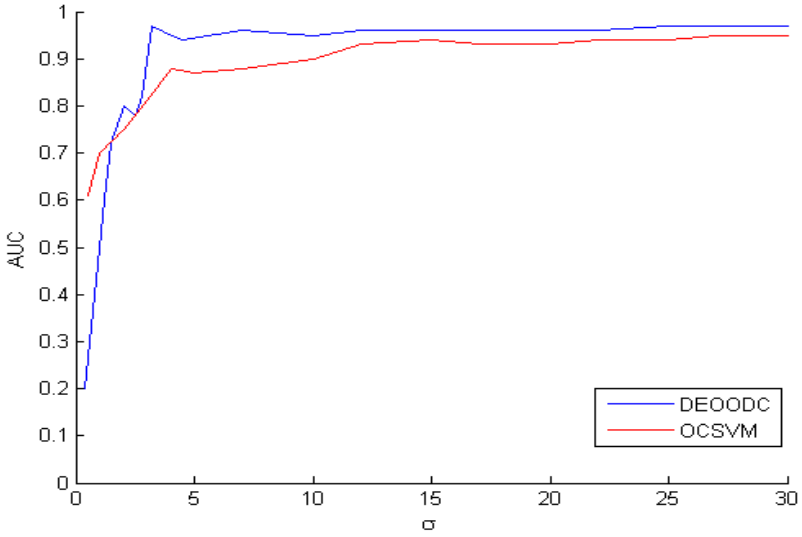


**Fig 2.** IBRL dataset: Maximum AUC value versus kernel parameter value.

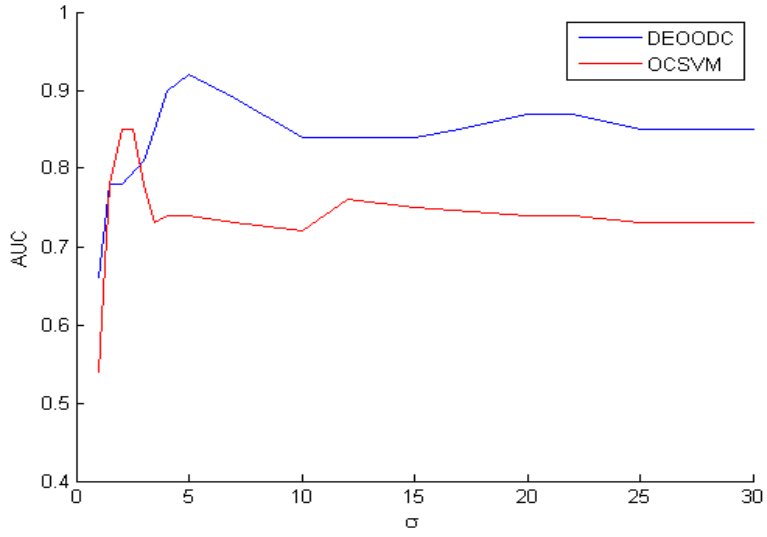


**Fig 3.** GStB dataset: Maximum AUC value versus kernel parameter value.

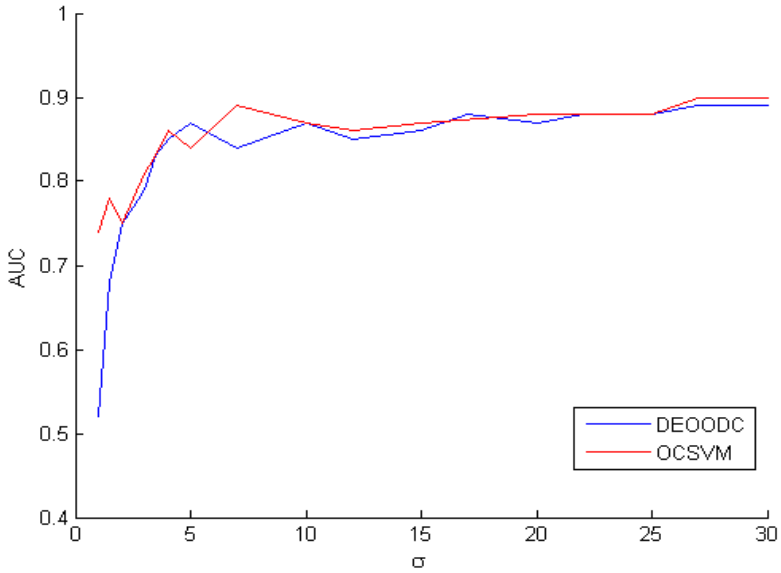


**Fig 4.** LUCE dataset: Maximum AUC value versus kernel parameter value

Our methodology can be applied on both small and large datasets. Our Approach is scalable and very efficient in the WSN application because when the dataset used are large, this give a better accuracy, increase the percentage of detection rate and decrease the false alarm rate. So, the use of other datasets doesn't affect the result because our solution is efficient and specially in WSNs domain. Our method is tested on control of fire in a wheat field application, so, it gives us a good detection rate with a minimum false alarm rate compared to Great-Duck-Island: [28], Volcano Monitoring: [29] and Sensorscope [30].

### 5.3 Computational complexity

We present the computational complexity of our proposed model on the testing phase in online manner. In this phase, the upper bound computational complexity involved in this process is O (M), where M is the number of observed variables. The training phase which involves the calculation of the PCs has a time complexity of O ($NM^2$) where N and M are the size and the dimension of the training set respectively. The complexity of online testing phase in our model structure is O (M) which is O ($N^3$) for the training phase of the OCSVM. The retraining of the OCSVM will cause a high power consumption which makes it unsuitable for anomaly detection in these types of environments compared to our DEOODC model.

## 5 Conclusion

In our work, we presents a comparative study between Distributed and Efficient One-class Outliers Detection Classifier (DEOODC) based on mahalanobis Kernel and One Class Support Vector Machine (OCSVM) for outlier detection in wireless sensor networks (WSNs). Our DEOODC demonstrated a higher classification performance on a real database used compared with OCSVM. So, our method demonstrated to be more robust against outlier detection within the training set. In order to showcase the

merits of our proposed approach, we performed a number of experiments that compared the capability of detecting outliers in data of the One-Class SVM and DEOODC detection methods. As a future work, we focus on improving the performances of the proposed model and extending it to be able to detect events that may occur instead of only considering outliers in an adaptive method. Also, we intend to utilize this model as a core for a cooperative framework for the whole network to achieve the energy efficiency.

## References


1. Naumowicz, T., Freeman, T., Heil, R., Calsyn, A., Hellmich, E., Brandle, A., Guilford, T., and Schiller, J. “Autonomous monitoring of vulnerable habitats using a wireless sensor network”. In : Proceedings of the Workshop on Real-World Wireless Sensor Networks, REALWSN’08. Glasgow, Scotland, 2008.
2. Marcelloni, F. and Vecchio, M. “An Efficient Lossless Compression Algorithm for Tiny Nodes of Monitoring Wireless Sensor Networks”. The Computer Journal 52(8), pp 969-987, 2009.
3. Akyildiz, A., Ian, F., Melodia, T., Kaushik, R. “A survey on wireless multimedia sensor networks”. Journal Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 51, Issue 4, Inc . New York, NY, USA, United State, 2007.
4. Ghorbel, O., Ayedi, W., Jmal, M.W., Abid, M. “Images compression and transmission in WSN: Performances Analysis”. 14th International Conference on Communication Technology. Chine, 2012.
5. Zhang, Y., Meratnia, N., Havinga, P. "Outlier detection Techniques for wireless sensor networks: A survey", pp .11-20, 2008.
6. Lee, J.-M., Yoo, C. Choi, S. W. Vanrolleghem, P. A. and Lee, I.-B. “Nonlinear process monitoring using kernel principal component analysis,” Chem. Eng. Sci., vol. 59, no. 1, pp. 223–234, January 2004.
7. Choi, S. W., Lee, C., Lee, J. M., Park, J. H and Lee, I. B. “Fault detection and identification of nonlinear processes based on kernel PCA,” Chemometrics Intell. Lab. Syst., vol. 75, no. 1, pp. 55–67, January 2005.
8. Scholkopf, B., Smola, A., and Muller, K.-R. “Nonlinear component analysis as a kernel eigenvalue problem,” Neural Comput., vol. 10, no. 5, pp. 1299–1319, July 1998.
9. Kapitanova, K., Son, S.H and Kang, K. D. “Event Detection in Wireless Sensor Networks. Second International Conference, ADHOCNETS 2010, Victoria, BC, Canada, August 2010.
10. Zhang, Y., Meratnia, N. P., Havinga, J.M. “Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine|. Ad Hoc Networks , December 2012.
11. Koupaie, HM. Ibrahim, S. and Hosseinkhani, J. “Outlier Detection in Stream Data by Machine Learning and Feature Selection Methods”, International Journal of Advanced Computer Science and Information Technology (IJACSIT), 2, 17-24, 2013.
12. Rajasegarar, S., Leckie, C., Palaniswami, M., Bezdek, J. C. “Quarter sphere based distributed anomaly detection in wireless sensor networks", Proceedings of IEEE International Conference on Communications, pp. 3864-3869, 2007.
13. Zhang, Y., Hammb, N.A.S, Meratniaa, N., Steinb, A., Van de Voorta, M. & Havingaa, P.J.M. “Statistics-based outlier detection for wireless sensor networks”, Volume 26, Issue 8, 2012.
14. Chakour, C. et al. 2012. “Adaptive kernel principal component analysis for nonlinear time-varying processes monitoring”. ICEECA 2012.
15. IBRL, Intel Berkely Reseach Lab Dataset, 2004. <http://db.csail.mit.edu/labdata/ 1120 labdata.html
16. GStB, Grand-St-Bernard dataset. http://lcav.epfl.ch/cms/lang/en/pid/86035. 2007.
17.LUCE, Lausanne Urban Canopy Experiment. <http://lcav.epfl.ch/cms/lang/en/ 1122 pid/86035>, ed, 2007.

18. Werner-Allen, G., Lorin CZ, K., Welsh, M., Marcillo, O., Johnson, J, Ruiz, M and Lees, J."Deploying a wireless sensors network on an active volcano", IEEE Internet computing ,pp.18-25,2006.
19. Szewezyk, R., Mainwaring, A., Polastre, J and Culler, D. "Analysis of alarge scale habitet monitoring application" ,In Proceedings of the second ACM conference en Embedded Networked sensors Systems(SenSys), Baltimore. 2004.
20. Verma, K., Kumar, V. and Samparthi, S. "Outlier Detection of Data in Wireless Sensor Networks Using Kernel Density Estimation". International Journal of Computer Applications. Published By Foundation of Computer Science, pp 28–32, 2010.
21. Nguyen, M.H., Torre, F. "Robust Kernel Principal Component Analysis", 2008.
22. Ding, M., Tian, Z and Xu, H. "Adaptive kernel principal component analysis. Signal Process, pp 1542-1553, 2010.
23. Zheng, W., Zou, C. and Zhao, L. "An Improved Algorithm for Kernel Principal Component Analysis. Neural Process, pp: 49-56, 2005.
24. Franc, V. and Hlavac, V. "Greedy algorithm for a training set reduction in the kernel methods," in Proc. Int. Conf. Computer Analysis of Images and Patterns, 2003, pp. 426–433.
25. Hoffmann, H. "Kernel PCA for novelty detection". Pattern Recognition, pp 863–874, 2007.
26. Xie, Z., Quirino, T., Shyu, M.-L., Chen, S.-C. and Chang, L."UNPCC: A Novel Unsupervised Classification Scheme for Network Intrusion Detection," presented at the Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, 2006.
27. Hotelling, H. Analysis of a Complex of Statistical Variables with Principal Components," Journal of Educational Psychology, 24, 498-520, 1933.
28. Szewezyk, R., Mainwaring, A., Polastre, J and Culler, D. "Analysis of alarge scale habitet monitoring application" ,In Proceedings of the second ACM conference en Embedded Networked sensors Systems(SenSys), Baltimore. 2004.
29. Werner-Allen, G., Lorin, K.CZ, Welsh, M., Marcillo, O., Johnson, J, Ruiz, M and Lees, J. "Deploying a wireless sensors network on an active volcano", IEEE Internet computing ,pp.18-25,2006.
30. Barrenetxea, G., Ingelrest, F., Schaefer, G., Vetterli, M., Couach, O and Parlange, M. "SensorScope: out-of-the-box Environmental monitoring", Proceeding of the 7th international conference on information processing in sensor networks, pp.332-343, April 22-24, 2008.
31. Vapnik, V. "The nature of statistical learning theory (Information Science and Statistics)," 1995.
32. Scholkopf, B., Platt, J, Shawe-Taylor, J, Smola, J and Williamson, R. C. "Estimating the support of a high-dimensional distribution." Neural computation, vol. 13, no. 7, pp. 1443–71, 2001.
33. Tax , D. M. and Duin, R. P. "Support Vector Data Description," in Machine Learning, vol. 27, no. 4, pp. 45–66, 2004.