DODS: A Distributed Outlier Detection Scheme for Wireless Sensor Networks

Chafiq Titouna, Farid Naït-Abdesselam, Ashfaq Khokhar

Version of Record: https://www.sciencedirect.com/science/article/pii/S138912861930012X
Manuscript_801822f740fb4f5fc7a7a55433c7e8c6

DODS: A Distributed Outlier Detection Scheme for Wireless Sensor Networks

Chafiq Titouna[a,b,*], Farid Naït-Abdesselam[b,c], Ashfaq Khokhar[c]

[a] *Department of Computer Science, University of Batna 2, 53 route de Constantine, Fésdis, 05078 Batna, Algérie*
[b] *LIPADE Lab., Paris Descartes University, 45 rue des Saints Pères, Paris, F-75006*
[c] *Department of Electrical and Computer Engineering, Iowa State University, USA*

Abstract

In many wireless sensor network (WSN) applications, where a plethora of nodes are deployed to sense physical phenomena, erroneous measurements could be generated mainly due to the presence of harsh environments and/or to the depletion of a sensor's battery. The measurements that significantly deviate from a normal behavior of sensed data are considered as outliers. To address the problem of detecting these outliers in wireless sensor networks, we propose a new algorithm, called Distributed Outlier Detection Scheme (DODS), in which multiple sensed data types are considered and where outliers are detected locally by each node, using a set of classifiers, so that neither information about neighbors is needed to be known by other nodes nor a communication is required among them. These characteristics allow the proposed scheme to be scalable and efficient in terms of both energy consumption and communication cost. The functionalities of the proposed scheme have been validated through extensive simulations using real sensed data obtained from Intel-Berkeley Research Lab. The obtained results demonstrate the efficiency of the proposed scheme in comparison to the surveyed algorithms.

Keywords: Wireless sensor networks, Outlier detection, Bayes classifier

*Corresponding author
Email address: c.titouna@univ-batna2.dz (Chafiq Titouna)

1. Introduction

The advances in the fields of transistors and semiconductor devices have led to the deployment of wireless sensor networks (WSNs). A wireless sensor network (WSN) is a self-organized network that consists of a large number of low-cost and low-powered sensor devices, which can be deployed in a field, in the air, in vehicles, on bodies, underwater, and inside buildings. These small sensing devices can cooperatively monitor real world physical or environmental conditions, such as temperature, pollution, pressure, light, voltage, humidity and motion. They are also considered as particular networks which are widely used in commercial and industrial areas, for example, transportation tracking, environmental and habitat monitoring, healthcare, etc. Moreover, in a military applications, WSNs can be used for target tracking and battlefield surveillance. In many of these applications, the data sensed by nodes are often unreliable. The quality of the data is affected by multiple noises and errors, missing values, duplicated data, or inconsistent data [1], without forgetting the low performance of nodes in terms of energy, computational and memory capabilities. These issues generally lead into having the generated data unreliable and inaccurate. One of the most sources that influence the quality of sensed data are outliers. We can define outliers as those measurements that significantly deviate from the normal pattern of the sensed data [1]. It means that the sensed data should be in coherence with a pattern which represents the reality of the sensed data. Therefore, it is clear that outlier detection is a crucial task in WSNs as It improves the quality of data, the security of the system, and maximizes the lifetime of the network.

Historically, research in outlier detection started in data management field [2, 3]. A definition of an outlier is given by Hawkins [4] where he considered outlier as an observation that deviates a lot from other observations and can be generated from a different mechanism. In WSN, outlier detection technique is the process of identifying those data instances that deviate from the rest of the data patterns based on a certain measure [5]. So, every measurement whose features dissent significantly from the normal behaviors is considered as outliers. In this paper, we present a new outlier detection algorithm, called DODS (for Distributed Outlier Detection Scheme). The main idea is to clean sensed data (measurements) from outlier (incorrect data). The proposal is base on a classification method to classify sensed data in a distributed manner. The scheme operates in nodes which made the sensing operation and does not require any neighbor's communication. In short, our

main contributions can be summarized as follows:

- Design of multiclassifier-based outlier detection algorithm in nodes;
- Parameterization of classifiers to deal with different types of sensed data;
- Simulation of the proposal in order to show its effectiveness in terms of detection accuracy, false alarm, and energy consumption.

The remainder of this paper is organized as follows. Section 2 mainly reviews the literature related to outlier detection techniques in WSN. In Section 3, we first introduce some formulations and definitions used in our approach and then, we describe in detail our scheme. Section 4 presents the experimental results. We conclude the paper and suggest future work in Section 5.

2. Related Work

Outlier detection in WSNs has been studied and a number of schemes and surveys have been proposed in the literature [6, 7, 8, 9, 10]. However, designing a solution that does not require neighborhood information remains a challenging issue in WSNs research. Wu et al in [11] present two local techniques for identification of outlying sensors. The identification of event boundary is also proposed in this work. The authors use the spatial correlation exists among neighbors. To exploit this characteristic, nodes compute the difference between its own measurements and the median of those of the neighborhood. If the result is greater than a pre-defined threshold, the node is considered as outlying one. The accuracy is not high due to the fact that ignorance of the temporal correlation of sensors' measurements decreases the performance of the proposed protocol. In contrary, the authors in [12] propose a technique which exploits the temporal correlation concept. Each node computes a distance similarity to detect outliers and communicates the result to the neighborhood by a broadcasting message. This technique permits the identification of global outliers, but the use of the broadcasting technique increases communication overhead. Zhang et al. present in [13], a technique based on distance to identify a set of global outliers in a snapshot. This technique uses a structure of aggregation tree to minimize the broadcasting of messages and reduce communication overhead. The identification of n global outliers is done by sending a useful data from nodes to the sink. After that,

the sink treats these data and then broadcasts outlier to network's nodes for agreement. The result of the identification of outliers is not sure due to the fact that the topology of WSN is not stable. Zhuang and Chen in [14] present two in-network outlier cleaning techniques for data collection applications of sensor networks. The first technique uses wavelet analysis to detect outliers. The second uses dynamic time warping (DTW). These techniques exploit the advantage of spatiotemporal correlations existing in readings of sensor nodes. The disadvantage of these techniques is the use of many thresholds which are difficult to define. Other categories of techniques use the concept of clustering where they start by grouping similar data instances into clusters with similar behavior. Data instances are identified as an outlier if they do not belong to clusters or if the cluster is significantly smaller than other clusters. In [15], authors propose a technique that minimizes the communication overhead by clustering the sensor measurements and merging clusters before communicating with other nodes. The advantage of this technique is that it does not need any prior knowledge on data distribution, but it needs to fix the width of the cluster. However, in spectral decomposition-based approaches, several techniques are proposed in the literature, using principal component analysis (PCA) for outlier detection. Chatzigiannakis et al.[16] propose a technique based on PCA to resolve the problem of accuracy in data generated by faulty nodes. The technique develops a model for the spatiotemporal correlations existing between sensed data in a distributed way. This model is used to detect outlier in sensor node through neighboring sensor nodes readings. The disadvantage of this technique is computationally expensive; which is caused by the selection of a good model. Furthermore, other solutions are based on classification to detect outliers. These approaches are often used in data mining and machine learning community. These approaches allow learning a classification model using the set of data instances (training phase) and classify an unseen instance into one of the learned (normal/outlier) class (testing phase) [1]. Abid et al. [17] proposed a solution called OPTICS. The methodology developed is a density-based classification technique and method ordering points to detect the clustering structure. The proposal can configure automatically the parameters without previous known environmental conditions. However, the comparative results show a low outlier detection rate. Rajasegarar et al. [18] propose a technique using one-class quarter-sphere to identify outliers in each node in a distributed manner. All nodes analyze sensed data offline after collecting all readings, which causes an outlier detection delay. So, it cannot be applied in real-time applications. Lu et al. [19]

presented an outlier detection method based on Cross-correlation. The proposal involves three essential parts: using linear interpolation in order to reprocess the data, cross-correlation analysis for outlier analysis and a multilevel Otsu's method for outlier rank. The proposed method can detect and isolate outliers in high dimensional time series datasets, and the hierarchical output of detection results. The authors in [20] propose a technique based on spatiotemporal correlations to learn contextual information statistically. Markov models are used and every sensor node computes the probabilities of its readings being in one predefined interval. If the probability of the sensed data is not being in the target interval, it will be considered as an outlier. A similar approach was proposed by Bahrepour et al. [21], they used the naïve bayesian networks in collaboration with neural networks for the detection of outliers. In [22], authors propose two techniques using dynamic Bayesian networks (DBN) to detect outliers locally in each sensor node. The aim of using DBN is to prevent the dynamic network topology. Recently, the authors in [23] present a new approach called Combined Kernelized Outliers Detection Technique (CKODT) based WSNs in the domain of water pipeline. The authors combined numerous methods for dimensionality reduction techniques and fault detection such as the Kernel Fisher Discriminant Analysis (KFDA) and the One Class Support Vector Machine (OCSVM). The experimental results showed the efficiency of the proposal compared to other approaches in the literature.Contrary to the ideas developed in the above reviewed works in which the neighbor's information is required and only one type of sensed data is considered, our proposal mainly focused on the design and development of self-detection nodes that are able to detect autonomously outliers where several sensed data types are collected by sensors.

3. Distributed Outlier Detection Scheme

The main goal of the DODS algorithm is in-network outlier detection. The solution exploits the temporal correlations existing in the sensed data (current and history sensed data) of the same node and its remaining energy level. Outlier detection is performed using Bayes' classifier for each type of data. This technique permits a multivariate classification sensed data in a distributed fashion. Figure 2 shows the structure of our approach which is represented by a data type identifier and a set of classifiers. The data type identifier allows knowing the type of measured data to direct it to the good classifier (classifier 1, 2, 3, ..., n). In our simulation experiences, we

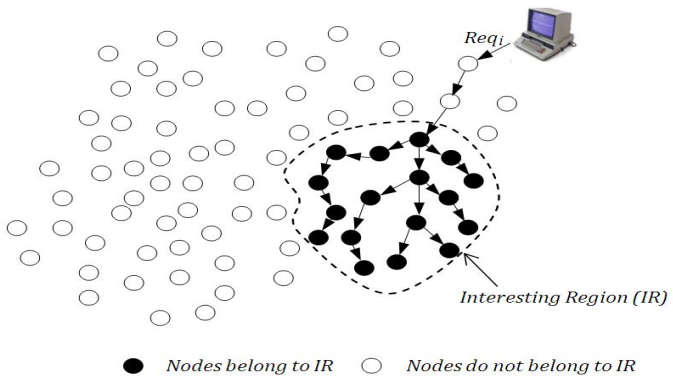


Figure 1: Nodes randomly deployed over an area.

use only four classifiers (temperature, light, voltage, and humidity classifier) according to the real datasets used in different scenarios. So, nodes belong to an interesting region (IR) participate in the outlier detection process. We mean that when a BS sends request req_i for example, only nodes of this region perform the classification task and not all nodes of the network. As shown in Figure 1, the black circles represent a set of nodes belongs to IR of the request req_i. The white circles are nodes belong to an uninteresting region by the request req_i. We describe the proposed algorithm and details its behavior in the next sub-sections.

3.1. System Assumptions

In the design of the proposed approach, some assumptions have been considered in order to be complying with a distributed detection. We assume that all static nodes are homogeneous, the computation and power capabilities of all of them are the same. Nodes' batteries cannot be recharged and each node is equipped with a power control device that has capabilities to vary their transmit/receive power. We assume that nodes are locations unaware. Let us say that $S = \{s_1, s_2, ..., s_n\}$ is the set of n stationary randomly deployed nodes with unique identifiers $ID \in [1, n] \cap N$, on a 2-dimensional square field. The hierarchical structure of WSN adopted in our approach, consist of a set of clusters $CL = \{cl_1, cl_2, ..., cl_m\}$. These clusters have not necessarily the same size. Furthermore, each node $s_i \in S$, $S = \{s_1, s_2, ..., s_n\}$ gathers information from the environment after receiving a request req_i from the base station. Finally, we summarize the used notations in Table 1.

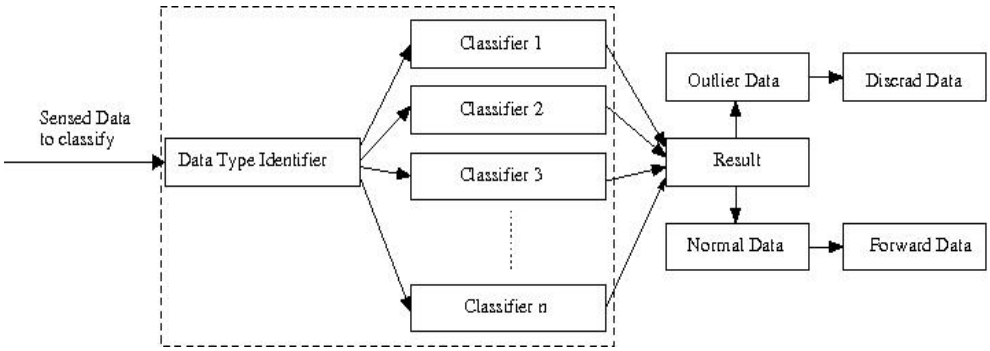


Figure 2: Classification structure of our approach.

Notation	Description
S	Set of static nodes
ID	Identificator of a node
CL	Set of clusters
BS	Base Station
CH	Cluster Head
req_i	Request i sent by BS
CPT_i	Conditional Probability Table of the node i
EL_i	Energy Level of the node i
HSD_i	History of Sensed Data of the node i
CSD_i	Current Sensed Data of the node i

Table 1: Notation.

3.2. Problem formulation

In order to classify sensed data, we employ the formalism of Bayesian networks. A Bayesian network is a directed acyclic graph (DAG) that represents a probability distribution. In such a graph, each random variable X_i is denoted by a node. A directed edge between two nodes indicates a probabilistic influence (dependency) of a child. Consequently, the structure of the network denotes the assumption that each node X_i in the network is conditionally independent of its non-descendants given its parents. To describe a probability distribution satisfying these assumptions, each node X_i in the network is associated with a conditional probability table (CPT_i), which specifies the distribution over X_i given any possible assignment of values to its parents[24]. A Bayesian classifier is simply a Bayesian network applied to

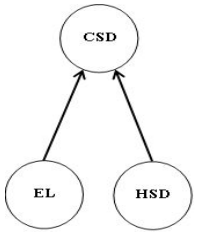


Figure 3: Our Bayesian Network.

a classification task[24]. It contains a node C representing the class variable and a node X_i for each of the features. Given a specific instance x (an assignment of values $x_1, x_2, \ldots, x_n$ to the feature variables), the Bayesian network allows us to compute the probability $P(C = c_k|X = x)$ for each possible class c_k . This is done via Bayes' theorem, giving us

$$p(C = c|X = x) = \frac{p(C = c)\ p(X = x|C = c)}{p(X = x)} \tag{1}$$

The critical quantity in Eq.1 is $P(X = x|C = c_k)$, which is often impractical to compute without imposing independence assumptions. The oldest and most restrictive form of such assumptions is embodied in the naïve Bayesian classifier [25] which assumes that each features X_i is conditionally independent of every other feature, given the class variable C. Formally, this yields

$$p(X = x|C = c) = \prod_i p(X_i = x_i|C = c) \tag{2}$$

In our approach, we consider the Bayesian Network presented in Figure 3. Our model consists of one observed variable (evidence), the Current Sensed Data (CSD) and two hidden data: the first one is the Energy Level (EL) of the node, the second one is the History of Sensed Data (HSD). The use of such data helps us to infer the classifier and give more accuracy in the detection of outliers. The HSD permits to exploit the temporal correlation exists between sensed data of the same node. On the other hand, the remaining energy represented by Energy Level is one of the influenced parameters on sensing operation [26], it is useful to verify if a node has enough energy to perform its function properly. Such a parameter can be computed by the node itself. According to the Eq.1, we obtain the following conditional

probabilities equations:

$$p(CSD|EL) = \frac{p(EL|CSD)\ p(CSD)}{p(EL)} \tag{3}$$

$$p(CSD|HSD) = \frac{p(HSD|CSD)\ p(CSD)}{p(HSD)} \tag{4}$$

Now, we compute the joint probability distribution $PJ(x_1, x_2, \ldots, x_n)$ which encapsulates all the variables (parameters). It is defined by using the chain rule, which is the result of the following product:

$$PJ(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i|par(x_i)) \tag{5}$$

Where x_1 represents the variable defined on the network and $par(x_i)$ represents the parents of the node. Matching the Eq.5 on the Bayesian network described by Figure 3, we obtain the following equation:

$$PJ(CSD|EL, HSD) = p(CSD|HSD)\ p(CSD|EL)\ p(CSD) \tag{6}$$

In order to learn the prior probability and to compute all CPTs, we use a supervised off-line method. Such a technique permits to reduce computation and maximizes outlier detection accuracy.

3.3. Inference algorithm

The process of detecting outliers begins by inferring the classifier. To achieve this purpose, we use the maximum a posteriori (MAP) concept [22, 27]. The aim of this technique is to determine all optimal classes $c = c_1, c_2, \ldots, c_m$ by maximization of MAP given the evidence. The MAP formula of our approach is described in the following equation.

$$c_{MAP} = \arg\max_{c_i \in C}\ p(CSD_i|EL_i, HSD_i) \tag{7}$$

$$c_{MAP} = \arg\max_{c_i \in C}\ p(EL_i|CSD_i)\ p(HSD_i|CSD_i)\ p(CSD_i) \tag{8}$$

We can apply Bayes' theorem to the formula above, we obtain:

$$c_{MAP} = \arg\max_{c_i \in C} \frac{p(EL_i, HSD_i|CSD_i)\ p(CSD_i)}{p(EL_i, HSD_i)} \tag{9}$$

$$c_{MAP} = \arg\max_{c_i \in C} \frac{p(EL_i|CSD_i)\ p(HSD_i|CSD_i)\ p(CSD_i)}{p(EL_i, HSD_i)} \tag{10}$$

We note that the denominator is a constant and its value does not affect the argmax, so we can drop it. We obtain the following formula:

$$c_{MAP} = \arg\max_{c_i \in C} p(EL_i|CSD_i)\ p(HSD_i|CSD_i)\ p(CSD_i) \tag{11}$$

We note that in our design, we consider different classes for different sensed data. To do that, we suppose $T = t_1, t_2, \ldots, t_n$, as a set of classes for the sensed data "Temperature". For "Humidity", we put $H = h_1, h_2, \ldots, h_m$ as classes of the classifier. The set of classes proposed to "Light" and "Voltage" is $L = l_1, l_2, \ldots, l_k$ and $V = v_1, v_2, \ldots, v_p$ respectively. So, c_i in Eq.7 represents one of the classes mentioned above. According to the sensed data, a node can use a specific classifier with a specific class. Figure 2 shows different classifiers implemented in nodes. For example, if the sensed data are measured by temperatures sensor unit, the classifier i specified to Temperature Data will use the classes $T = t_1, t_2, \ldots, t_n$ for inference's process and so on.

We summarize our approach in the following algorithm:

Algorithm 1 The DODS Algorithm

BEGIN

Step 1: Initialize parameters

N : node in an interesting region (IR)

//we consider only 4 classifiers (temperature, humidity, light and voltage)

$T = t_1, t_2, \ldots, t_n$: set of classes of temperature data

$H = h_1, h_2, \ldots, h_m$: set of classes of humidity data

$L = l_1, l_2, \ldots, l_k$: set of classes of light data

$V = v_1, v_2, \ldots, v_p$: set of classes of voltage data

$type\_of\_CSD = type_T, type_H, type_L, type_V$

Let EL_N be the energy level of the node N

Let CSD_N be the Current Sensed Data of the node N

Let HSD_N be the History (Last) Sensed Data of the node N

Step 2: Computing of maximum a posteriori (MAP)

Switch $type\_of\_CSD$ **do**

$type_T : c_{MAP} = \arg\max_{c \in T} p(EL_N|CSD_N)\ p(HSD_N|CSD_N)\ p(CSD_N)$

$type_H : c_{MAP} = \arg\max_{c \in H} p(EL_N|CSD_N)\ p(HSD_N|CSD_N)\ p(CSD_N)$

$type_L : c_{MAP} = \arg\max_{c \in L} p(EL_N|CSD_N)\ p(HSD_N|CSD_N)\ p(CSD_N)$

$type_V : c_{MAP} = \arg\max_{c \in V} p(EL_N|CSD_N)\ p(HSD_N|CSD_N)\ p(CSD_N)$

end Switch

Step 3: Comparison of result

Switch $type\_of\_CSD$ **do**

$type_T$: use T to find $class\_of\_CSD$;

if $class\_of\_CSD = class\_of\_c_{MAP}$ **then**

CSD is $Normal\_DATA$; $FORWARD\_CSD$

else CSD is $Outlier\_DATA$; $REMOVE\_CSD$ **endif**

$type_H$: use H to find $class\_of\_CSD$;

if $class\_of\_CSD = class\_of\_c_{MAP}$ **then**

CSD is $Normal\_DATA$; $FORWARD\_CSD$

else CSD is $Outlier\_DATA$; $REMOVE\_CSD$ **endif**

$type_L$: use L to find $class\_of\_CSD$;

if $class\_of\_CSD = class\_of\_c_{MAP}$ **then**

CSD is $Normal\_DATA$; $FORWARD\_CSD$

else CSD is $Outlier\_DATA$; $REMOVE\_CSD$ **endif**

$type_V$: use V to find $class\_of\_CSD$;

if $class\_of\_CSD = class\_of\_c_{MAP}$ **then**

CSD is $Normal\_DATA$; $FORWARD\_CSD$

else CSD is $Outlier\_DATA$; $REMOVE\_CSD$ **endif**

end Switch

END

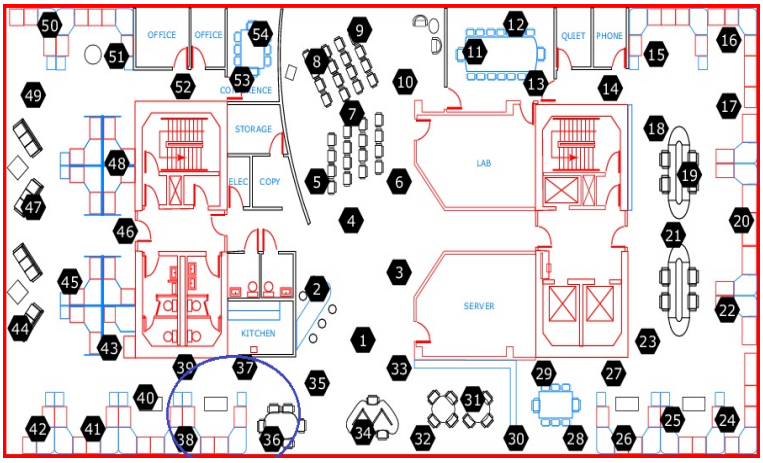


Figure 4: Sensors in the Intel Berkeley Research Lab[28].

4. Performance Evaluation

In order to evaluate our scheme, a set of data were obtained and a number of experiments were conducted. Section 4.1 describes the datasets, while Section 4.2 defines evaluation metrics; Section 4.3 shows the simulation parameters and in the Section 4.4 reports the final results.

4.1. Datasets

In order to be close to the reality, experiments have been performed by using the realistic sensed data collected from 54 Mica2Dot sensors deployed in Intel Berkeley Research Lab between February 28 and April 5, 2004 (see Figure 4) [28].

The sensed data included temperature, humidity, light, and voltage values collected once in $31s$. The quantity of data is about 2.3 million readings; it was collected using the TinyDB in-network query processing system, built on the TinyOS platform[28]. All values measured by sensors are presented in Table 2. The epoch is a monotonically increasing sequence number from each mote. Moteids range from 1 to 54; data from some motes may be missing or truncated. Temperature is in degrees Celsius. Humidity is ranging from 0 to 100%. Light is in Lux (a value of 1 Lux corresponds to moonlight, 400Lux to a bright office, and $100,000$ Lux to full sunlight). Voltage is expressed in volts, ranging from 2 to 3; the batteries, in this case, were lithium ion cells which maintain a fairly constant voltage over their lifetime.

Date	Time	Epoch	Moteid	Temp	Humidity	Light	Voltage
$(yy-mm$ $-dd)$	$(hh:mm$ $ss:xxx:)$	(int)	(int)	$(real)$	$(real)$	$(real)$	$(real)$

Table 2: Dataset schema.

In the experiments, we first selected some measurements from the nodes with $IDs = 36, 37$ and 38 (see Figure. 2), for the time period from 2004-03-11 to 2004-03-14 corresponding to 15763 log rows. We separate this dataset according to features (temperature, humidity, light, and voltage). We obtain 4 synthetic datasets named: Dataset-Tmp, Dataset-Hmd, Dataset-Lght, and Dataset-Volt. To evaluate our approach, we add 1000 outliers (Abnormal value) to each previous Datasets.

4.2. Evaluation metrics

To evaluate the performance of the proposed algorithm, we analyzed three principle metrics: Detection Accuracy Rate (DAR), False Alarm Rate (FAR) and Energy Consumption. To do that, we use a confusion matrix (CM) [29]. CM determines True and False Positives (TP, FP), thus True and False Negatives (TN, FN). TP can be defined as real outlier detection by a node. On the other side, FP is occurring when a node concludes that a sensed data are an outlier but is not. The TN denotes that when a node it signals that there is no outlier in a correct data. Finally, when a node does not detect an existing outlier, FN increases. This matrix allows us to evaluate carefully the accuracy of our approach. DAR and FAR can be computed using the following equations:

$$DAR = \frac{TP}{(TP + FN)} \tag{12}$$

$$FAR = \frac{FP}{(FP + TN)} \tag{13}$$

As regards energy consumption, this metric represents the total energy dissipated by all nodes to sense and transmit the measured data. The energy consumed by the radio of each node has been estimated basing on the model proposed by Heinzelman [30]. In this model, sending and receiving a k-bit packet with distance d, generate a radio consumption $E_{TX}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2$, and $E_{RX}(k) = E_{elec} * k$ respectively, Where:

Parameters	Value(s)
Square m^2	$100 * 100$
Number of nodes	81
Cluster size	10
Number of clusters	8
Node radio range	40 m
Transmission channel	Wireless channel
Propagation model log	Normal path loss model
Data packet size	32 *bytes*
Bandwidth	200 Kilobytes per second
Radio layer	$CC2420$ radio layer
Queue size	50 packets

Table 3: Simulation parameters.

- $E_{elec} = 50nJ/bit$: energy for running the transmitter/receiver circuitry.
- $\epsilon_{amp} = 100pJ/bit/m^2$: energy for running the transmitter amplifier.

4.3. Simulation parameters

Our experiments are conducted under TOSSIM tool [31]. TOSSIM is a TinyOS simulation tool which simulates WSN physical and link layer features accurately. This allows validating the solution under realistic WSN deployment conditions. In the experiments, we chose one of the most popular sensor platforms, $Mica2$. We use 81 sensor nodes to form 10 clusters. We Consider sensor node with $ID = 1$ as the sink and sensor nodes with $IDs = 36, 37, 38$ represent sensor nodes 36, 37 and 38 respectively of our Berkeley's dataset selected in section 4.1. Sensor node 2 is the CH of the previous set's sensor nodes. The simulation parameters are depicted in Table 3.

4.4. Results and discussion

In this section, we present our experimental results for the proposed algorithm. We compare the performance of our proposed DODS scheme with CollECT event detection proposed by Wang et al [32], and with the outlier detection algorithm (OD) proposed by Asmaa et al [33]. To do that, experiences are conducted according to three scenarios. We use different intervals (Small, medium and large) to compute c_{MAP}. Table. 4 and 5 summarize the initialization of these intervals. We also consider the initial energy of nodes with $IDs = 36, 37, 38$ equal to $18,720$ Joules, that corresponds to the energy of two AA batteries.

	Temperature(° C)
Small interval	$[-50,-45][-45,-40][-40,-35][-35,-30][-30,-25]$ $[-25,-20][-20,-15][-15,-10][-10,-5][-5,0][0,5]$ $[5,10][10,15][15,20][20,25][25,30][30,35][35,40][40,45]$ $[45,50]$
Medium interval	$[-50,-40][-40,-30][-30,-20][-20,-10][-10,0]$ $[0,10][10,20][20,30][30,40][40,50]$
Large interval	$[-50,-30][-30,-10][-10,10][10,30][30,50]$

(a) Intervals (case of Temperature).

	Voltage(Volt)
Small interval	$[2.000,2.025][2.025,2.050][2.050,2.075][2.075,2.100]$ $[2.100,2.125][2.125,2.150][2.150,2.175][2.175,2.200]$ $[2.200,2.225][2.225,2.250][2.250,2.275][2.275,2.300]$ $[2.300,2.325][2.325,2.350][2.350,2.375][2.375,2.400]$ $[2.400,2.425][2.425,2.450][2.450,2.475][2.475,2.500]$ $[2.500,2.125][2.525,2.550][2.550,2.575][2.575,2.600]$ $[2.600,2.625][2.625,2.650][2.650,2.675][2.675,2.700]$ $[2.700,2.725][2.725,2.750][2.750,2.775][2.775,2.800]$ $[2.800,2.825][2.825,2.850][2.850,2.875][2.875,2.900]$ $[2.900,2.925][2.925,2.950][2.950,2.975][2.975,3.000]$
Medium interval	$[2.00,2.05][2.05,2.10][2.10,2.15][2.15,2.20][2.20,2.25]$ $[2.25,2.30][2.30,2.35][2.35,2.40][2.40,2.45][2.45,2.50]$ $[2.50,2.55][2.55,2.60][2.60,2.65][2.65,2.70][2.70,2.75]$ $[2.75,2.80][2.80,2.85][2.85,2.90][2.90,2.95][2.95,3.00]$
Large interval	$[2.0,2.1][2.1,2.2][2.2,2.3][2.3,2.4][2.4,2.5][2.5,2.6]$ $[2.6,2.7][2.7,2.8][2.8,2.9][2.9,3.0]$

(b) Intervals (case of Voltage).

Table 4: Initialization of intervals (case of Temperature and Voltage)

	Light(Lux)
Small interval	$[0, 62.5][62.5, 125][125, 187.5][187.5, 250][250, 312.5]$ $[312.5, 375][375, 437.5][437.5, 500][500, 562.5]$ $[562.5, 625][625, 687.5][687.5, 750][750, 812.5]$ $[812.5, 875][875, 937.5][937.5, 1000][1000, 1062.5]$ $[1062.5, 1125][1125, 1187.5][1187.5, 1250][1250, 1312.5]$ $[1312.5, 1375][1375, 1437.5][1437.5, 1500][1500, 1562.5]$ $[1562.5, 1625][1625, 1687.5][1687.5, 1750][1750, 1812.5]$ $[1812.5, 1875][1875, 1937.5][1937.5, 2000]$
Medium interval	$[0, 125][125, 250][250, 375][375, 500][625, 750][750, 875]$ $[875, 1000][1000, 1125][1125, 1250][1250, 1375]$ $[1375, 1500][1625, 1750][1750, 1875][1875, 2000]$
Large interval	$[0, 250][250, 500][500, 750][750, 1000][1000, 1250]$ $[1250, 1500][1500, 1750][1750, 2000]$

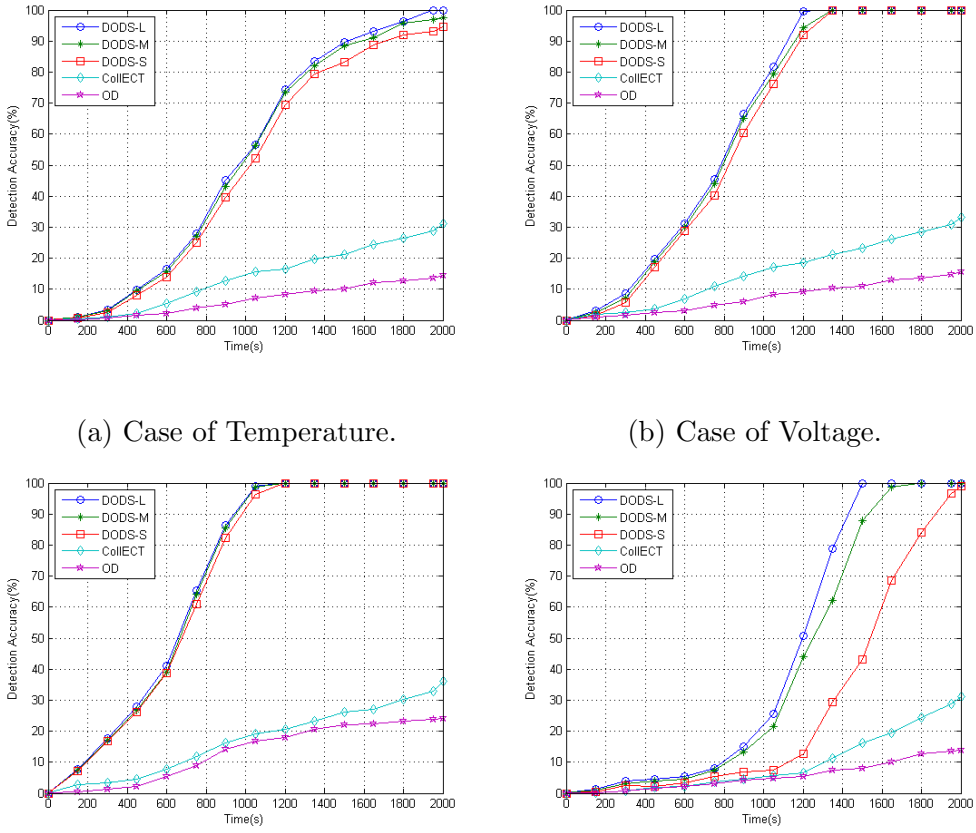
(a) Intervals (case of Light).

	Humidity(%)
Small interval	$[0, 5][5, 10][10, 15][15, 20][20, 25][25, 30][30, 35][35, 40]$ $[40, 45][45, 50][50, 55][55, 60][60, 65][65, 70][70, 75][75, 80]$ $[80, 85][85, 90][90, 95][95, 100]$
Medium interval	$[0, 15][15, 30][30, 45][45, 60][60, 75][75, 90][90, 100]$
Large interval	$[0, 25][25, 50][50, 75][75, 100]$

(b) Intervals (case of Humidity).

Table 5: Initialization of intervals (case of Light and Humidity).

For all scenarios, we proceed to 30 runs under the same test conditions. We execute temperature, light, voltage and humidity simulations separately. Figure. 5a shows the number of outliers detected in case of temperature, versus the simulation time. The rest of figures (Fig. 5b, Fig. 5c and Fig. 5d) concerns voltage, light and humidity. From the curves visible in Fig. 5a, it can be observed that the DODS-L with large intervals produces a good result. It can detect all outliers in a minimum of time. On the other hand, when the intervals become smaller, the detection of outlier needs more time (case of DODS-M and DODS-S). The Fig. 5a also shows clearly that our proposed approach DODS-L with large intervals outperforms outlier detection (OD) approach and the CollECT algorithm. Indeed, the use of wide intervals in DODS-L allows more possibility for a calculated value (c_{MAP})



(a) Case of Temperature.

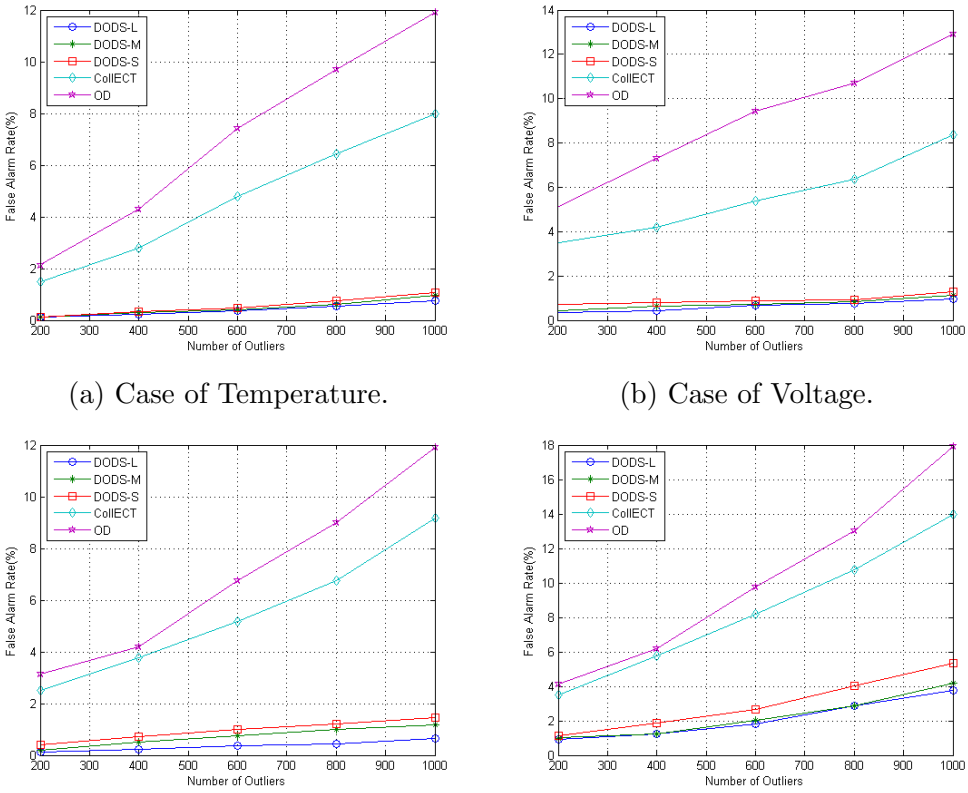
(b) Case of Voltage.

(c) Case of Light.

(d) Case of Humidity.

Figure 5: Detection accuracy for different types of data.

in Eq.11, to be in the range of the current sensed data. However; the OD approach is based on four steps to classify data; (a) first: clustering algorithm is applied to group data into clusters; (b) second: for each cluster, an algorithm of outlier detection is launched to classify normal and outlier cluster; (c) third step: outlier classification is executed to separate error and event data; (d) finally, computing the degree of trustfulness of the readings of each node. Each step requires time and energy to be finalized, which is not acceptable in WSN. In addition, if it occurs an error in the construction of clusters in step 1 of the approach, the process of classification will generate false results. For the case of CollECT algorithm, it is based on several pro-



(a) Case of Temperature.

(b) Case of Voltage.

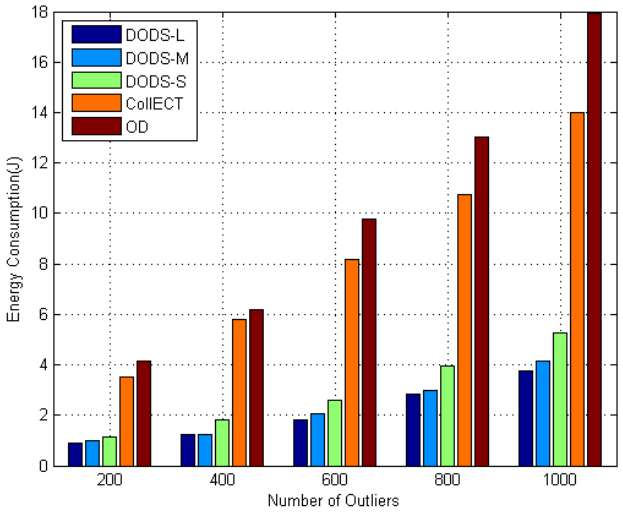
(c) Case of Light.

(d) Case of Humidity.

Figure 6: False alarm rate for different types of data.

cedures (vicinity triangulation, event determination, and border sensor node selection). It started by the construction of the estimated attribute region to determine the occurrence of the event (outlier), and to identify in some cases, the event boundary. However, the algorithm requires a collaboration of nodes to get high accuracy. This condition increases time and energy consumption. In our approach, DODS-L detects all outliers and the execution time is less than that outlier detection approach and CollECT algorithm. The good performance of DODS-L comes from the idea used to delegate the outlier detection process in a distributed manner. This solution attributes a twofold role to the node: at the same time, it serves as a measurement node and as a cleaning tool.

Besides the evaluation of the detection accuracy metric, Figure. 6 shows



(a) .

Figure 7: Energy consumed vs. Number of outliers.

the false alarm rate versus the simulation time. From results, there is a clear trend that the scenario with large interval (DODS-L) outperforms all approaches (outlier detection approach and CollECT algorithm), which reveals the effectiveness and efficiency of the proposed scheme. This gain is mainly favored by the adopted features of DODS and by the proposed model (see Figure. 2) for types of sensed data. However, from Figure. 6, we observe that DODS-S obtains higher false alarm rate than the other variants (DODS-M, DODS-L). The reason for this increase lies in the use of small intervals which increases the number of classes. That means, when we computed c_{MAP} of a current sensed data, even it is normal (not an outlier), the probability where it falls in the same interval is very low.

Finally, Figure. 7 depicts the energy consumed in joules by nodes. As shown, the histograms represent the consumption of energy when we variate the number of outliers (from 200 until 1000 outliers) in case of temperature. It is clear that our DODS-L outperforms OD approach and CollECT algorithm. In wireless sensor networks, three units consume energy: wireless communication, CPU and sensing unit. We note that the communication unit consumes more energy compared to other components. Since our algorithm detects outliers locally in nodes and does not require any neighbors

information exchanging, so it performs better than the other approaches and consumes less energy.

5. Conclusion

Most of the proposed approaches for outlier detection in wireless sensor networks require having some information and knowledge about the neighboring nodes. However, due to the high energy consumption due to wireless communications, these approaches are proven to not be optimal and efficient, and more research is needed to further enhance the performances of such algorithms. To this goal, we proposed in this paper a highly efficient algorithm, called Distributed Outlier Detection Scheme (DODS). The effectiveness of this scheme derived from its fully distributed way of operation as it does not involve any messages exchange in the neighborhood. To evaluate the performance of the proposed algorithm, a large number of experiments have been performed using real and synthetic datasets. The proposed algorithm delivers very interesting performances, thereby demonstrates its effectiveness. As a future work, we plan to introduce new models for a better and precise separation of the outlier detection from the event detection.

References

[1] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 12, pp. 159–170, Second 2010.

[2] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, (New York, NY, USA), pp. 427–438, ACM, 2000.

[3] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, SIGMOD '01, (New York, NY, USA), pp. 37–46, ACM, 2001.

[4] D. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, pp. 15:1–15:58, July 2009.

[6] M. Moshtaghi, C. Leckie, S. Karunasekera, and S. Rajasegarar, "An adaptive elliptical anomaly detection model for wireless sensor networks," *Computer Networks*, vol. 64, pp. 195 – 207, 2014.

[7] H. Liu, A. Nayak, and I. Stojmenović, *Fault-Tolerant Algorithms/Protocols in Wireless Sensor Networks*, pp. 261–291. London: Springer London, 2009.

[8] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," *Knowledge and Information Systems*, vol. 34, pp. 23–54, Jan 2013.

[9] C. Titouna, M. Aliouat, and M. Gueroui, "Outlier detection approach using bayes classifiers in wireless sensor networks," *Wireless Personal Communications*, vol. 85, pp. 1009–1023, Dec 2015.

[10] A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, "Outlier detection approaches for wireless sensor networks: A survey," *Computer Networks*, vol. 129, pp. 319 – 333, 2017.

[11] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, and P. Deng, "Localized outlying and boundary data detection in sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 1145–1157, Aug 2007.

[12] J. Branch, B. Szymanski, C. Giannella, R. Wolff, and H. Kargupta, "In-network outlier detection in wireless sensor networks," in *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, 2006.

[13] K. Zhang, S. Shi, H. Gao, and J. Li, "Unsupervised outlier detection in sensor networks using aggregation tree," in *Advanced Data Mining and Applications* (R. Alhajj, H. Gao, J. Li, X. Li, and O. R. Zaïane, eds.), (Berlin, Heidelberg), pp. 158–169, Springer Berlin Heidelberg, 2007.

[14] Y. Zhuang and L. Chen, "In-network outlier cleaning for data collection in sensor networks," in *In CleanDB, Workshop in VLDB 2006*, pp. 41–48, APPENDIX, 2006.

[15] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Distributed anomaly detection in wireless sensor networks," in *2006 10th*

IEEE Singapore International Conference on Communication Systems, pp. 1–5, Oct 2006.

[16] V. Chatzigiannakis, S. Papavassiliou, M. Grammatikou, and B. Maglaris, "Hierarchical anomaly detection in distributed large-scale sensor networks," in *11th IEEE Symposium on Computers and Communications (ISCC'06)*, pp. 761–767, June 2006.

[17] A. Abid, A. Masmoudi, A. Kachouri, and A. Mahfoudhi, "Outlier detection in wireless sensor networks based on optics method for events and errors identification," *Wireless Personal Communications*, vol. 97, pp. 1503–1515, Nov 2017.

[18] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *2007 IEEE International Conference on Communications*, pp. 3864–3869, June 2007.

[19] H. Lu, Y. Liu, Z. Fei, and C. Guan, "An outlier detection algorithm based on cross-correlation analysis for time series dataset," *IEEE Access*, vol. 6, pp. 53593–53610, 2018.

[20] E. Elnahrawy and B. Nath, "Context-aware sensors," in *Wireless Sensor Networks* (H. Karl, A. Wolisz, and A. Willig, eds.), (Berlin, Heidelberg), pp. 77–93, Springer Berlin Heidelberg, 2004.

[21] M. Bahrepour, N. Meratnia, and P. Havinga, "Use of ai techniques for residential fire detection in wireless sensor networks," in *AIAI 2009 Workshop Proceedings*, pp. 311–321, CEUR-WS.org, 7 2009.

[22] D. J. Hill and B. S. Minsker, "Real-time bayesian anomaly detection for environmental sensor data," 2007.

[23] A. Ayadi, O. Ghorbel, M. BenSalah, and M. Abid, "Kernelized technique for outliers detection to monitoring water pipeline based on wsns," *Computer Networks*, vol. 150, pp. 179 – 189, 2019.

[24] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[25] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, (San Francisco, CA, USA), pp. 338–345, Morgan Kaufmann Publishers Inc., 1995.

[26] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sen. Netw.*, vol. 5, pp. 25:1–25:29, June 2009.

[27] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, Inc., 1 ed., 1997.

[28] "Intel lab data home page, last consultation april 2018," http://db.csail.mit.edu/labdata/labdata.html, 2014.

[29] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proc. of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, (New York, NY, USA), pp. 157–166, ACM, 2005.

[30] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, Oct 2002.

[31] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), pp. 126–137, ACM, 2003.

[32] K.-P. Shih, S.-S. Wang, H.-C. Chen, and P.-H. Yang, "Collect: Collaborative event detection and tracking in wireless heterogeneous sensor networks," *Computer Communications*, vol. 31, no. 14, pp. 3124 – 3136, 2008.

[33] A. Fawzy, H. M. Mokhtar, and O. Hegazy, "Outliers detection and classification in wireless sensor networks," *Egyptian Informatics Journal*, vol. 14, no. 2, pp. 157 – 164, 2013.