IET Wireless Sensor Systems

Research Article

Outlier detection for wireless sensor networks using density-based clustering approach

ISSN 2043-6386
Received on 7th May 2016
Revised 5th December 2016
Accepted on 2nd February 2017
E-First on 15th March 2017
doi: 10.1049/iet-wss.2016.0044
www.ietdl.org

Aymen Abid[1] ✉, Abdennaceur Kachouri[2], Adel Mahfoudhi[3]
[1]*CES-Lab, ENIS, University of Sfax, Sfax, Tunisia*
[2]*LETI-Lab, ENIS, University of Sfax, Sfax, Tunisia*
[3]*CCIT, University of Taif, Taif, Saudi Arabia*
✉ *E-mail: aymen.abid.mail@gmail.com*

Abstract: Outlier detection (OD) constitutes an important issue for many research areas namely data mining, medicines, and sensor networks. It is helpful mainly in identifying intrusion, fraud, errors, defects, noise and so on. In fact, outlier measurements are essential improvements to quality of information, as they are not conforming to expected normal behaviour. Due to the importance of sensed measurements is collected via wireless sensor networks, a novel OD process dubbed density-based spatial clustering of applications with noise (DBSCAN)-OD has been developed based on the algorithm DBSCAN, as a background for OD. With respect to the classic DBSCAN approach, two processes have been jointly combined, the first of computing parameters, while the second concerns class identification in spatial temporal databases. Through both of these modules, one is able to consider real-time application cases as centralised in the base station for the purpose of separating outliers from normal sensors. For the sake of evaluating the authors proposed solution, a diversity of synthetic databases has been applied as generated from real measurements of Intel Berkeley lab. The reached simulation findings indicate well that their devised method can prove to help effectively in detecting outliers with an accuracy rate of 99%.

1 Introduction

Outlier detection (OD) is a serious challenge for several fields including biology, computing, wireless sensor networks (WSNs) and so on [1, 2]. With respect to WSN, it is applied for maintaining routing, security, supervision, data analysis and other important issues [3].

Composed of a diversity of small and low cost sensor nodes, WSNs help in supervising and sensing its environmental physical characteristics for the sake of achieving the appropriately fit decision making [4]. Yet, these frames might well turn out to be due to node or link constraint, e.g. low energy, harsh environment and so on. It is in this respect that OD appears to be useful, as it helps greatly in identifying anomalies through applying appropriate data to make the right decision achievable [5].

Indeed, outlier data in WSN represents measurements deviated from normal pattern [6], whereby outlier detector aims at discovering their abnormal behaviour [7].

Indeed, several detection techniques are available [8, 9] whereby the clustering methods prove to be useful and effective. In this regard, the elaborated study [10] have provided a primitive non-parametric pattern recognition method that integrates similarity and nearest neighbours to construct clusters based on a little priority knowledge of data structure.

For a better intrusion detection in WSN, the authors of [11] have introduced a density-based fuzzy imperialist competitive clustering algorithm, applying a density-based algorithm and fuzzy logic rules in a bid to optimise the data clustering process. In this context, the fuzzy logic controller serves to adjust the fuzzy rules for error tolerance at the imperialist actions. For evaluation sake, the real data benchmark of Intel Berkeley Research Lab [12] has been applied to establish a comparison with other empirical methods relevant to preserving accuracy (ACC) against malicious detection.

Similarly, Shamshirband *et al.* [13], on attempting to resolve distributed denial service attacks by cooperative game-based fuzzy Q-learning detector. They have appealed to a number of cooperative defence counter attack scenarios pertaining the sink node and the base station. The aim has been to establish rational decision layers via a game theory strategy. To test the flooding packet performance, the Low Energy Adaptive Clustering Hierarchy protocol under NS-2 simulator has been applied.

In turn, Ahmadi *et al.* [14] consider that routing helps greatly in reducing energy consumption and increasing network lifetime. Still, the choice of suitable track for a reliable undamaged data transfer to take place is also important. For this sake, they propose a technique useful for preserving k-coverage and data reliability within a logical fault tolerance. They proceed by firstly collecting primary information (position, energy, cluster heads, sink location etc.). Secondly, they undertake to set up coverage clusters for targets and selection of cluster heads. Ultimately, active transmitting nodes are detected for information transfer to the sink. For simulation purposes of lifetime and active nodes' number, sensor nodes are deployed randomly in the space with uniform distribution radius transmission.

In a bid to increase the sensors' lifetime in WSN, the authors in [15] put forward an imperialist competitive algorithm mechanism which serves to divide sensor nodes into cover sets for a rather efficient monitoring of routing targets to be maintained. In their simulations, a random dispersion of nodes and target is deployed within the same sensing range.

So, due to the importance of information quality for WSN and its applications (medicine, military etc.), which are generally implemented in real time process, the OD algorithm stands as a critical device appropriately fit for resolving such problems. Nevertheless, it would hurry out to be convenient to use a rather unsupervised technique perspective that waves sound to be so flexible and portable for WSN that its constraints and assumptions would differ from an application case to another. In what follows is a presentation of some works conducted as attempts to overcome such a challenge.

In this regard, and for an effective identification of anomalies to be safeguarded, Hassan *et al.* [16] have resorted to establishing cluster derived from data through applying current and historical measurements. Clustering has been used to identify anomalies by means of unsupervised detector dubbed ‘IKD’ during the clustering phase, an in-network clustering algorithm has been deployed to classify data according to cluster width. In the OD phase, the produced clusters are labelled as outliers or not, for once far from the average inter-cluster distance, they are out considered as such. Still with outlier-cluster identification, each sensor is rated with respect to the number of its corresponding outlier measurements. Their method has undergone an assessment stage via Intel base [12]. As the environment stability results in a noticeable lack of abnormalities, they have undertaken to randomly inject some synthetic data with abnormal measurements into this base. Such a process entails fixing the clusters width, which leads to restricting the range of application.

In [17], the authors reckon to apply spatial and temporal correlations as statistics-based OD techniques to ameliorate the quality of WSN produced data. In fact, temporal, spatial or spatial–temporal data turn out to be derived, depending on the correlations' type being applied, i.e. whether temporal or spatial. At a first stage, and on testing a temporal OD (TOD), they have managed to identifying outliers online, using the time-series model. The autoregressive model is used to predict the value. An outlier is identified if a point $x(s,\ t)$ is out of the confidence interval of its predicted value $\hat{x}(s, t)$. In a second stage, a spatial OD (SOD) is discussed to identify outlier nodes, as SOD helps analyse the neighbour nodes derived measurements to predict the next point $\hat{x}(s, t)$ and detect outliers being out of the confidence interval. Spatial–temporal outlier detectors could be elaborated. Temporal and spatial real-data-based OD (TSOD) turns on to be the combination of TOD and SOD, as each node would serve to locate its temporal aberrant values and check whether if the neighbourhood does affirm its diagnosis. Then, communication would be reduced, while their spatial predicted-data-based OD (POD) would set the neighbourhood measurements' stemming precision for outliers to be detected without data transmission. The framework's spatial and temporal integrated OD (STIOD) uses temporal and spatial correlations. In fact, each node should receive the neighbours' reached correlation results at a given time it considers in its correlation computation. The latter is subject of prediction, generated to compare actual and predicted observation and decide whether an outlier state has been achieved. These performances are evaluated through time-series and geostatistical models helping to compare their detection ACC using detection rate (DR) and false positive rate schemes.

As for Ahmadi Livani *et al.* [18], they propose appealing for a supervised OD approach via normalised data. This normal pattern is but an outcome of training phase pertaining to the first principal components analysis (PCA) applying it as global PCA (GPCA), they have managed to classify data into normal and outlier types throughout the OD phase. Building on this process, they undertake to modify their global pattern at update phase. Yet, this solution proves to be very sensitive to data size and scheme number (events etc.) in input data. As a simulation framework, they consider comparing their centralised and distributed propositions of GPCA outlier detector (GPCA-OD) by means of an Intel base [12], whereby failed nodes could be introduced. In fact, they help generate outlier data as measured by signal-to-noise ratio. Due to its complexity and to the PCA sophisticated use, the algorithm entails proceeding with an important computation energy.

Concerning Chitradevi *et al.* [19], they explain a density-based OD using a minimum support scheme in a bid to reduce the score calculating overload outliers in multivariate measurements sensors (DBOD-MSS). To calculate scores relevant to each measurement in the set of accumulated sensor data, they use a factor of local outlier based on a k-distance of neighbours. As a first step, they have implemented a calculation of the distance separating neighbours of each data set point and defining the adequate k-distance neighbourhood. In a second step, they have considered determining the reachability distance and local reachability density corresponding to each point. Ultimately, both of these stages are used to calculate the local outlier factor of each point and find outliers. In conformity with the evaluations provided by the authors of [16, 18], the Intel base [12] has been used to perform real-life test. Synthetic data have been inserted into the real base using the random function ‘randomGr’ to generate a different probability of error in the data suspicions and, subsequently, the failure of nodes. Yet, this detector yielded weak performance results due to its false and omission detection.

In [20], the authors make appeal to a faulty in-node detector technique (FINDT) that helps define the historical base of faulty nodes in order to be either corrected, replaced or removed. According to the FINDT, a node is dubbed a faulty once its data rank proves to be over a lower threshold. The process computation is performed using a predefined map division of nodes in concord with the network topology. Such a method is discovered to be WSN environment dependent as a number of parameters need to be calculated prior to OD implementation. Then, just a single real-life simulation is attributed to ranking parameters' performance and another with a manual selection of faulty nodes for detection performance.

In [21], a clear attempt has been made for an online and local OD approach (OLODA) to be applied for false alarm to be reduced and more accurate detection results to be achieved. They consider the possibility for parameters variation as in the case for real time use, through Markov statistics setting. Experiments are carried out via an application of the Neyman–Pearson (NP) test help produces false alarm sequences liable to evaluation by means of such a statistical detector.

The authors in [22] have put forward a spatial–temporal similarity (STODM) detection methodology conceived to classify abnormal data into error and event based on spatial similarity among neighbours and temporal similarity within a given time set. Similarity is calculated through Euclidean distance, and outliers are depicted once the fuzzy logic probability process proves not to meet a given set threshold. In fact, Matlab simulation is fit for application within a real dataset case.

As highlighted, some previously elaborated works appear to require a prior knowledge of the environment for data models to be set up and algorithmic parameters to be configured. Other studies require a great deal of energy to be made and resources' computation to be performed either for extra inter-node communications or for its many and complicated to be maintained or the several and sophisticated phases to be implemented. Insofar as the present work is concerned, an attempt is made to detect abnormal measurements delivered by nodes using an OD algorithm. In this respect, a density-based clustering method is applied for OD to be maintained in a bid to effectively identify any possible errors and events with no prior knowledge of clusters' number or labels being required. In addition, the process is implemented independently from topology, scalability changes and data shape.

The remaining of this work is organised as follows: Section 2 is devoted to explaining the proposed detection method and its theoretical background. While Section 3 involves the experiment's performances. Ultimately, Section 4 respects the major concluding remarks.

2 Advanced method

This section deals with studying the case of unsupervised clustering algorithm as based on the density approach dubbed *DBSCAN* (density-based spatial clustering of applications with noise). In a first place, an exploration of the density-based clustering theory and methods is exposed followed by development of our proposed our solution as conceived to detect outliers by means of such an algorithm.

2.1 Theoretic background

It is worth noting that density-based clustering methods are forwarded constructing a series of sets as derived from input data depicting a dense region [23]. Noteworthy also, in that a given region data are standing too firmly close to each other. The best-known methods fit for application with regard to a data-analysis of this type are mainly the density-based algorithm for discovering clusters *DBSCAN*, the density-based clustering *DENCLUE* along with the ordering points to identify the clustering structure *OPTICS*.

Concerning the *DBSCAN* [24], it serves to execute a random computation of reachability value with respect to each single point.

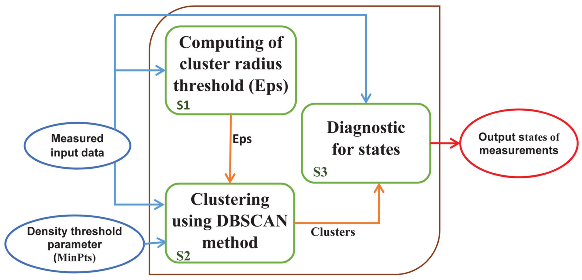
Computing of cluster radius threshold (Eps)
S1
Measured input data
Eps
Diagnostic for states
S3
Output states of measurements
Clustering using DBSCAN method
S2
Density threshold parameter (MinPts)
Clusters

Fig. 1 *General design of the proposed OD method*

```
Algorithm 1
Input:
    points: Measured input data
Output:
    Eps: computed of minimum reachability radius Eps for DBSCAN use
  1. Compute distance "dist" lying between points using equation 7
  2. Eps=mean(dist) using equation 6
```

Fig. 2 *Algorithm 1: The calculation of 'Eps'*

Under this framework, the various points can be combined into clusters by means of a density-based connectivity analysis. As for *OPTICS* [25], it follows the same clustering procedure, as undertaken via density and connectivity, only that it implements a pre-sorting process of the points on the basis of their distance reachability. In addition, hierarchical clusters are also achievable via this clustering mode [26]. With regard to *DENCLUE* [27], it helps in constructing data sets through implementation of a special function helping to compute the density distribution. Such a procedure entails pursuing a diversity of input configuration variables.

The *DBSCAN* methodology is particularly applicable for clustering spatial data relaying to several areas of application, mainly field as chemistry, social science and images [28]. The complexity of the method lies at the order of $O(n\log(n))$, which bears two parameters: *Eps* and *MinPts*. It follows a two-step undertaking, namely, similarity computing process applying the density principle as well as a cluster identification procedure through implementation of reachability principle.

As a first step, *DBSCAN* carries out a computation for each point $p_i \in P$ a local density, i.e. the number of points p_j which are shorter in distance than the *Eps* distance. So, this number can be calculated as follows:

$$N_{Eps}(p_i) = \left\{ p_j | \forall j \in [1, \ldots, N_{Eps}], \text{distance}(p_i, p_j) < Eps \right\} \quad (1)$$

It follows from the density computing process that a point can be classified into three types: core, border and noise. According to N_{Eps}, the point p_i can be categorised as a core point once the local density throw out to be higher than *MinPts*:

$$p_i \text{ iscore point if } \quad \text{card}(N_{Eps}(p_i)) \geq MinPts \quad (2)$$

In addition, the point p_i can stand as a border point if the number of its neighbours proves to be inferior to *MinPts* but lies in the neighbourhood of a core point, such as

$$\begin{gathered} p_i \text{ is border point if} \\ \text{card}(N_{Eps}(p_i)) < MinPts \\ \text{and} \\ \text{core points}(N_{Eps}) \neq \phi \end{gathered} \quad (3)$$

Another possible case is that where p_i appears to have only few neighbours and at least a single core point in its neighbourhood. In this case

$$\begin{gathered} p_i \text{ is noise point if} \\ \text{card}(N_{Eps}(p_i)) < MinPts \\ \text{and} \\ \text{core points}(N_{Eps}(p_i)) = \phi \end{gathered} \quad (4)$$

As a second step, DBSCAN derive to check reachability possibilities in order to create clusters. As figured in (5), two points are density-reachable if there exists a chain of points lying between them where each of them represents a core point and the next in the chain is neighbour of the current point. A point p would prove to be a neighbour of point q, if it proves to lie within the set $N_{\text{Eps}}(p)$.

So, p_n is density reachable from p_1 if there exists a chain

$$\begin{gathered} ch = \left\{ p_i | i \in [1, n], p_i \in N_{Eps} \right\} \\ \text{and} \\ \forall p \in ch, \ \ p \text{ is core point} \end{gathered} \quad (5)$$

Still, density-based algorithms throw out to display some weakness with respect to border points, as it could well be misidentified as much as it may be wrongly classified, e.g. points of adjacent clusters. Hence, a revised version has been proposed by Daszykowski *et al.* [28] through recovering the problem with *DBSCAN*. As for WSN, it might will result in a confusion between the events and the outliers, especially for the very near noise. In this work, this new version of *DBSCAN* is examined by means of real WSN events by evaluating the outliers' detection capacity through synthetic random outlier points.

2.2 OD process using DBSCAN method

The proposed detector's concept schema is shown in Fig. 1, involving three steps.

In a first stage *S*1, the advanced proposal denotes that the algorithm should compute an average of distance lying between the measured points as value for *Eps* (Algorithm 1). Actually, the distance shall be equal to the mean μ of the distances corresponding vector $\boldsymbol{V}$ (6), computed via a Euclidean method defining distance lying between the point and the rest of the vector (7) (Algorithm 1 (see Fig. 2)). (see (6))

$$\text{For a point} p, \ \text{distance}(p, \text{points}) = \sqrt{\sum\nolimits_{for\, all x_i \in \text{points}} (p, x_i)},$$
$$\text{with points representing vector for measurements of sensor nodes}. \quad (7)$$

In a second stage *S*2, the *DBSCAN* is implemented for the purpose of generating clusters of homogeneous values, while separating them from noisy points (Algorithm 2 (see Fig. 3)). As noted in [28], for each unclassified point x_i in the base, it should be checked whether the point is a core point or a noise one. In the core point case and as a first step, a *ClusterId* is assigned. As a second step, a list of reachable points' *seeds* is computed. For every point x_j appearing in this list, the algorithm would try to find its fit

$$\forall x_i \in V, \quad \mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \ \text{ with } V \text{ designating vector for distances lying}$$
$$\text{between measurements of sensor nodes}. \quad (6)$$

```
Algorithm 2
Input:
    points: Measured input data
    MinPts: Density threshold parameter
    Eps: radius threshold
Output:
    Eps: computed of minimum reachability radius Eps for DBSCAN use
$\forall x_i \in points$
  1. $seeds$=reachable($x_i$,points,Eps)
  2. if cardinality($seeds$)$< MinPts$,
       label($x_i$,NOISE)
  3. else
       create($ClusterId$)
       $\forall j \in seeds$,
           $N_{Eps}(x_j)$=reachable($x_j$,points,Eps)
           if cardinality($N_{Eps}(x_j)$)$\geq MinPts$
           assign($ClusterId$,$x_j$)
           $\forall$ unclassified $x_k \in N_{Eps}(x_j)$, insert($x_k$,$seeds$)
           $\forall$ unclassified and noise $x_k \in N_{Eps}(x_j)$, insert($x_k$,$border\ list$)
  4. FOR all $x_{border}$ of $border\ list$
       $N_{Eps}(x_{border})$=reachable($x_{border}$,points,Eps)
       Add $x_{border}$ to closed core point of this $N_{Eps}(x_{border})$
```

Fig. 3 *Algorithm 2: DBSCAN process*

```
Algorithm 3
Input:
    points: Measured input data
    Cluster$_{1..n}$: the "n" Cluster deduced from the points
Output:
    pointsSt: Vector of outlier states corresponding to points
  1. Clen=length(Cluster);
     //Clen contains the number of clusters
  2. if $(Clen > 0)$
     for i=1:Clen
       if all points of $Cluster_i$ originate from same Sensor
       for all $points_j$ in $Cluster_i$,
               $pointsSt_j$="OUTLIER"
       else
       for all $points_j$ in $Cluster_i$,
               $pointsSt_j$="NORMAL"
  3. j=1:length(points)
       if pointsId $(points_j)$=="NOISE"
               $pointsSt_j$="OUTLIER"
```

Fig. 4 *Algorithm 3: Diagnosis for the measured data states*

Table 1 Confusion matrix of OD

	Predicted outlier	Predicted normal
actual outlier	true positive – TP	false negative – FN
actual normal	false positive – FP	true negative – TN

neighbours $N_{\rm Eps}(x_j)$. If the number of neighbours proves to be higher than *MinPts*, x_j is then a core point and it is assigned to the current cluster of x_i. In a last stage, the entirety of unclassified points in $N_{\rm Eps}(x_j)$ is inserted into the current *seeds* list. To note also, is that the unclassified and noise points appearing in the neighbour list are labelled border points. At the end of the DBSCAN algorithm clustering process, the points $x_{\rm border}$ remaining in the borders' list are visited to be assigned to the closest core point in their neighbourhood list $N_{\rm Eps}(x_{\rm border})$.

In our context, as an input for this step *S*2, *MinPts* is selected as being equal to two for at least two other neighbours shall remain within the same group, thus, agreeing with the potential core point candidate. In fact, for an agreement consensus to be reached, it would be rather convenient to involve at least three members [29].

In the third step *S*3 of the process, we identify the points statuses, thus the sensors are recognised as the source of each measurement (Algorithm 3 (see Fig. 4)). Indeed, for points belonging to the same sensor, the entirety of the cluster points is presented as outliers. Consequently, the sensor that outputs them remains in a situation of failure. Noise points are also accounted for as outliers. Other remaining points and clusters are considered as normal.

3 Performance evaluation

3.1 Detection performance measures

Based on the confusion matrix, formulated to evaluate OD as shown in Table 1, some useful metrics are developed as indicated of DR, false alarm rate (FAR) and ACC.

In this way, the DR is reduced by

$$\mathrm{DR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100, \qquad (8)$$

The FAR is

$$\mathrm{FAR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \times 100, \qquad (9)$$

and ACC is

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \times 100. \qquad (10)$$

3.2 Simulation platform

The detector evaluation procedure is performed by means of a real database available at Intel Berkeley base [12]. Actually, several research works, as in [30, 31], have been conducted with a special reference made to this base. Data are stored and stem from 54 sensor-nodes in a 35 × 45 m field according to their locations. In a 30 s period called time slot (TS), a node can provide either none, one or several measurements. Regarding simulation, only the latest data provided are considered, as the process exclusively accounts for the latest state of sensor in a current period. With respect to our particular detection evaluation, 500TSare used providing 21,034 temperature measurements of a single event and containing only few erroneous points. So, in each TS an outlier value is introduced applying a synthetic database (SDB) involving 21,534 temperature values. On considering the time series flow regarding the 54 nodes, temperature degrees are appear to range between [14°C; 28°C] during all the TSs periods. The standard deviation $\sigma_{\rm RDB}$ relevant to this real base is 2.85 and the mean $\mu_{\rm RDB}$ is 20.26. Then, the confidence interval is $[\mu - 3\sigma; \mu + 3\sigma] = [20.22; 20.29]$ containing the proportion of 99.7% of measurements [32]. Accordingly, it has been noted that

$$\forall \text{ points } x_i \in V \text{ vector of } N \text{ points,} \quad \sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N} \left|x_i - \mu\right|^2} \qquad (11)$$

and that the mean μ relevant to the study points is computed with save way as in (6). Synthetic outlier measurements are established in terms of the mean of the real base $\mu_{\rm RDB}$ as deviated from coefSt × $\sigma_{\rm RDB}$ (Algorithm 4 (see Fig. 5)). For the sake of maintaining difference among points from each other, a random value (*R*) is introduced and extracted from RAN2 [33] with a uniform distribution in $[a;\ b] = [0;\ 1]$ and standard deviation $\sigma_R = (b - a)/2\sqrt{3} = 1/2\sqrt{3}$. The random number generator RAN2 is fitted with a long period reaching 2^{62} and can approve statistical

```
Algorithm 4
Input:
    k: ID of sensor to be outlier
    points: points of the base
    tsEnd: number of TS used to generate the synthetic base
    coefSt: coefficient of deviation of synthetic values
Output:
    SDB: Base having synthetic outlier sensor
    1. m=means(points);
    2. St = std(points); //St contains the standard deviation $\sigma_{RDB}$
    3. SDB=points(1:tsEnd,:); //SDB receive points into TS=tsEnd
    4. j=1;
    5. R=Ran2(i)
    6. for i=1:tsEnd
         $SDB(i,k) = m - j \times CoefSt \times St - j \times R;$
         if $j == 1 \quad j = -1;$
         else $\quad j = \quad 1;$
```

Fig. 5 *Algorithm 4: Synthetic base generator with a single outlier sensor*

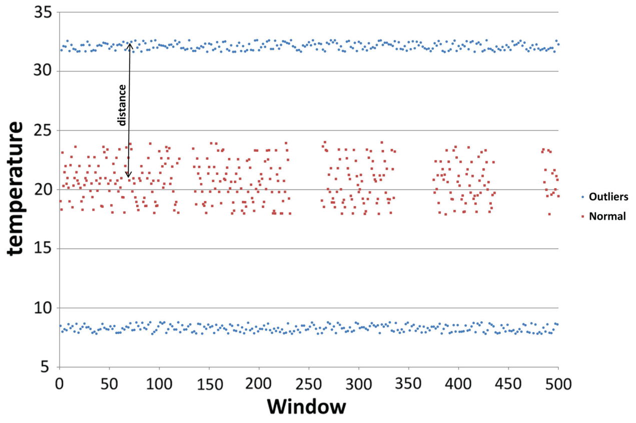

Fig. 6 *Distribution of synthetic outliers opposite to normal measurements*

Table 2 The synthetic bases characteristics as used in each simulation

Base	distance	σ_{Outliers}	σ_{SDB}	Deviation $\sigma_{\text{Outliers}}-\sigma_{\text{RDB}}$
SDB1	$0.5\sigma_{\text{RDB}}$	1.96	2.83	−0.9
SDB2	$1.0\sigma_{\text{RDB}}$	3.38	2.86	0.5
SDB3	$1.5\sigma_{\text{RDB}}$	4.8	2.9	1.95
SDB4	$2.0\sigma_{\text{RDB}}$	6.23	3	3.37
SDB5	$2.5\sigma_{\text{RDB}}$	7.65	3.04	4.8
SDB6	$3.0\sigma_{\text{RDB}}$	9.08	3.14	6.23
SDB7	$3.1\sigma_{\text{RDB}}$	9.36	3.16	6.51
SDB8	$3.2\sigma_{\text{RDB}}$	9.65	3.18	6.8
SDB9	$3.3\sigma_{\text{RDB}}$	9.94	3.2	7.08
SDB10	$3.4\sigma_{\text{RDB}}$	10.22	3.22	7.37
SDB11	$3.5\sigma_{\text{RDB}}$	10.51	3.24	7.66
SDB12	$3.6\sigma_{\text{RDB}}$	10.79	3.26	7.95
SDB13	$3.7\sigma_{\text{RDB}}$	11.08	3.28	8.23
SDB14	$3.8\sigma_{\text{RDB}}$	11.37	3.31	8.51
SDB15	$3.9\sigma_{\text{RDB}}$	11.65	3.33	8.8
SDB16	$4.0\sigma_{\text{RDB}}$	11.94	3.35	9.08

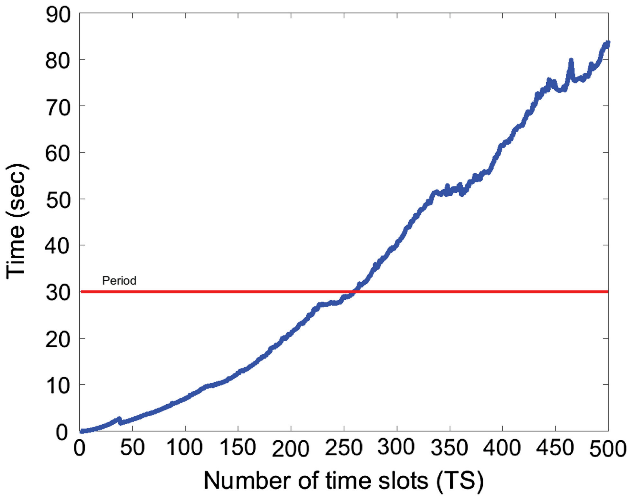

Fig. 7 *Window size execution time in respect of the number of TSs as used in each simulation test*

tests noticeably more appropriately than several other well-known generators (RAN0, RAN1 etc.) [34].

Our devised Algorithm 4 (Fig. 5), generate outlier values within both sides of a normal pattern, as shown in Fig. 6, depicting a sample measurement distribution relevant to a normal sensor. Besides, it illustrates a synthetic outlier sensor generated around distance = $4 \times \sigma_{\text{RDB}}$ far from the mean of the real-base RDB, μ_{RDB}. Hence, on decreasing the coefficient, the generated outliers turnout to get closer to the normal patterns.

The generated synthetic bases correspond to the *tsEnd* = 500TS from RDB. Outliers are inserted into the fifth sensor, $k = 5$, as actually it does not usually provide exact measurements. As a summary of useful information regarding our simulation applied synthetic bases is depicted in Table 2. σ_{Outliers} denotes the standard deviation of synthetic outliers generated for the given sensor. The distance distinguishing this value from the normal pattern, $\sigma_{\text{Outliers}}-\sigma_{\text{RDB}}$, figures in the last column of the table.

Concerning this study still and extend in every execution, the number of TSs is extended as a detector input by one TS involves only a single outlier measurement. We reckon extending the used data with a large number of real measurements as well as a linear incremental number of outliers.

3.3 Simulation results

For the purpose of evaluating the proposed method's performance, we start by discussing the execution time with respect to the actual WSN real-time application possibility. Then, a performance analysis is undertaken in respect of the random outliers' deviation. Subsequently, details of the performances as registered across a number of used data are outlined, followed by an ultimate comparison of our DBSCAN-OD achieved findings with the actual predominant methods.

In Fig. 7, an average of time spans recorded for the entirely of simulations with different deviations of random outlier values inserted in the data base. This time average is presented with relevance to a window size as applied with respect to each test. Worth noting in this regard, is that from 250TS, the detector proves to be unable to render any decision in real time whenever the average appears to exceed 30 s.

In Table 3, figures the study conducted to examine the detection behaviour through the performances' average as investigated through increasing the distance lying between normal and outlier measurements with respect to every simulation. The average is undertaken for each test with various deviation percent of random outlier values that help establish a different SDB regarding each case. It turns out that the detector does not seem to encounter any problems as to detecting normal patterns, even in the presence of synthetic normal measurements as it is the case with SDB1, SDB2 and SDB3. In fact, with the distance ranging between

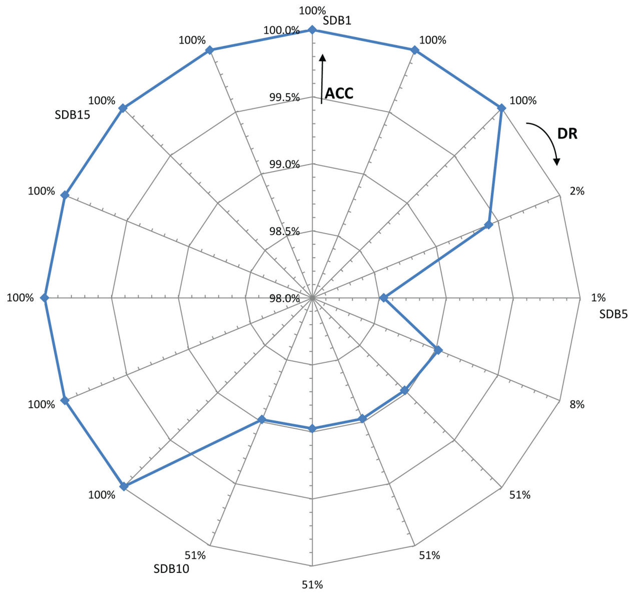

Fig. 8 *Rose plot relevant to the DR and ACC rates regarding each simulation base*

Table 3 Performance average results regarding each used synthetic outliers' deviation

BASE	DR, %	FAR, %	ACC, %
SDB1	100	0	100
SDB2	100	0	100
SDB3	100	0	100
SDB4	2	0.002	99
SDB5	1	0.002	98
SDB6	8	0.007	99
SDB7	51	0	98
SDB8	51	0	98
SDB9	51	0	98
SDB10	51	0	98
SDB11	100	0	99
SDB12	100	0	100
SDB13	100	0	100
SDB14	100	0	100
SDB15	100	0	100
SDB16	100	0	100
average	70	0.001	99

$[0.5\sigma_{\text{RDB}}; 1.5\sigma_{\text{RDB}}]$, synthetic values remain clearly without the normal pattern set. As for outliers lying to close to the normal values' interval, as is the case with SDB4, SDB5 and SDB6, the detector appears to fail in identifying errors (DR under 10%) with some confusion being registered for normals with a range of 0.005%. Nevertheless, ACC is discovered to maintain its high percentage (greater than 95%). At rate higher than $3\sigma_{\text{RDB}}$, the detector appear to succeed in identifying normals and outliers, with some handicaps noted to prevail with respect to SDB7, SDB8, SDB9 and SDB10 but with no problems being noted with regard to SDB11 up to SDB16.

Fig. 8 reflects the relationship prevailing between the DR and ACC rates, as depicted in Table 3. It figures the bases at which the joint performance of the couple (DR, ACC) turns out to be significantly for better than the other scored ones. Actually, according to this rose plot, starting from the fourth base up to the tenth the detector proves to meet some troubles finding outliers. Still, the ACC level remains somewhat high, even with *z* weakening DR percentage. Such a finding affirms well that the detector successfully manages all normals, even in the presence of some occasional confusion noted in identifying anomalies.

As in Fig. 9, it highlights the average detection performances, as recorded via the entirety of SDBs with different deviations scored with regard to each *TS* learning number. Ability to identify outliers, and more especially erroneous measurements, implies well the remarkable efficiency in detecting different anomaly states (noise, erroneous sensor). At its best effective rate, the DR is discovered to be noticeable by stable within the interval regarding between 60 and 65% once TS higher than 50TS are applied. For smaller windows, the DR turns out to be greater than 90% at the beginning, to unfortunately score a noticeable decrease to the rates of 80 and 70% whenever the tests applied TS are incremented. If at all, the DR scored results remain still fair even with a simultaneous application of several measurements, whether normal or abnormal. Meanwhile, accuracy computes the optimal degree fit for classifying measurements lying between normal and outliers. In Fig. 9*b* plot of the same figure, ACC is still greater than 99% although it proved to decrease by 0.1% once the window size is incremented from some tests' pack to another.

So, as highlighted concerning the previously discussed figures, the detector's work flow using extending window, prove to yield a detection results that seem critically important for the WSN applications. It is also worth noting that for all administered simulations, the DR average score turnout to be equal 70%. In addition, the ACC seems to have 99% and FAR gives 0.001% due to the SDB4, SDB5 and SDB6.

3.4 Discussion

In this sucsection, the focus of interest lies via a comparison purpose by rest, we will focus on the comparison of our proposed

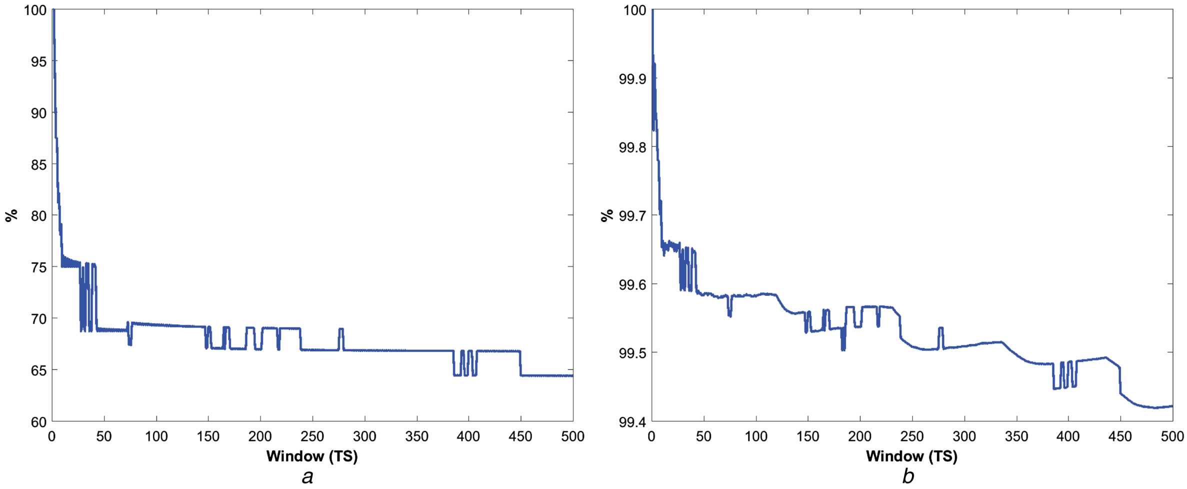

Fig. 9 *DR and ACC average rates for all bases across different simulation windows*
(a) DR, ***(b)*** ACC

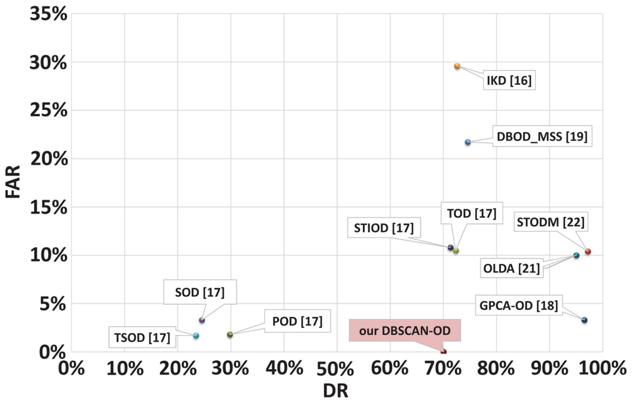

Fig. 10 *Comparative performances between our and reference methods*

solution with other methods. Fig. 10 establishes a comparison of the global average results relevant to the couple DR and FAR for all tests. It is a graphical comparison of our proposed solution as relying on a number of other related research works.

While the cited reference methods are focused on optimisation of DR or FAR. The present research is we aimed to optimise both the performance as well as the accuracy.

DBOD-MSS of [19] reaches an average of 75% DR with 22% FAR with regard to both solutions with simple minimum support scheme or with pairwise minimum support scheme. The IKD advanced by Hassan *et al.* [16] detects 73% of outliers with FAR equal to 30%. GPCA-OD [18] displays the best DR results with a 96% average, our solution process to perform even better with regard to FAR by 10^{-4}% against 3.3%. As a matter of fact, thanks to test performed with all possible outlier deviations from normal pattern and on comparing it to other works, our designed detector appears to yield a perfectly good FAR with fair DR of an average exceeding 70% in rate.

Statistics-based outlier detectors (TOD, SOD, TSOD, POD and STIOD) described in [17] prove to exhibit low percentage as the DR does not exceed 72% with important FAR between 2 and 11%. The FIND released by Guo *et al.* [20] does score good DR of 95% but records an important FAR with an average rate of 20%. In addition, STODM of [22] registers a bad FAR (10%), it has scored 97% of DR. The OLODA put forward by Ozkan *et al.* [21], which aims to reduce the false detection to improve accuracy, records still poor results of FAR although it displays a good DR.

Hence, our conceived DBSCAN-OD turns to participate successfully in reducing FAR rates with sufficient DR and optimal ACC. Perfect performances (100% DR, 0% FAR and 100% ACC) are recorded, as judging by the medium distance of outliers extracted from normal data (higher than $3.5\sigma_{\text{RDB}}$). Still, further amelioration and improvements seem impose, especially fir higher effectiveness to detect outliers, particularly the very near outlier points.

4 Conclusion

This paper has been conceived to depict a density-based outlier detection system using DBSCAN approach. Following a thorough computation of the minimum radius of accepted cluster '*Eps*', the revised version process of DBSCAN is safeguarded to further fit for data clustering. It is on this basis that decision can be made as to whether from this, we decide if each point is normal or abnormal.

The proposed algorithm's performance is tested on various synthetic bases generated from real life Intel Berkeley database, particularly, with such performance evaluation as namely DR, FAR and ACC. Actually, synthetic outlier values turnout to approach the normal pattern from a base to another. In each round test performed with specific base, the number of TSs, along with that the number of synthetic outlier has been incremented.

Relaying on the achieved experimental results, one might well notice that the proposed process performs remarkably better than the current devised methods mainly in terms of FAR and ACC. Indeed, the average DR does confirm well the robustness of the detector for near outliers with 70% of DR, 0.001% of FAR and 99% of ACC being scored.

5 References

[1] Qiu, Y., Cheng, X., Hou, W., *et al.*: 'On classification of biological data using outlier detection'. 12th Int. Symp. on Operations Research and its Applications in Engineering, Technology and Management (ISORA 2015), 2015, pp. 1–7

[2] Zhang, Y.-Y., Chao, H.-C., Chen, M., *et al.*: 'Outlier detection and countermeasure for hierarchical wireless sensor networks', *IET Inf. Sec.*, 2010, **4**, (4), pp. 361–373

[3] Aggarwal, C.C.: 'Outlier analysis', in '*Data mining*' (Springer International Publishing, 2015), pp. 237–263

[4] Khosravi, A., Kavian, Y.: 'Challenging issues of average consensus algorithms in wireless sensor networks', *IET Wirel. Sens. Syst.*, 2016, **6**, pp. 60–66

[5] Zhang, C., Ren, J., Gao, C., *et al.*: 'Sensor fault detection in wireless sensor networks'. Proc. of the IET Int. Communication Conf. on Wireless Mobile and Computing, 2009, pp. 66–69

[6] Zhang, Y., Meratnia, N., Havinga, P.: 'Outlier detection techniques for wireless sensor networks: a survey', *IEEE Commun. Surv. Tutor.*, 2010, **12**, (2), pp. 159–170

[7] Subramaniam, S., Palpanas, T., Papadopoulos, D., *et al.*: 'Online outlier detection in sensor data using non-parametric models'. Proc. of the 32nd Int. Conf. on Very Large Data Bases, VLDB Endowment, 2006, pp. 187–198

[8] ABID, A., KACHOURI, A., MAHFOUDHI, A.: 'Anomaly detection in wsn: critical study with new vision'. Int. Conf. on Automation, Control, Engineering and Computer Science – ACECS, IPCO, 2014

[9] Ahmed, M., Mahmood, A.N., Hu, J.: 'A survey of network anomaly detection techniques', *J. Netw. Comput. Appl.*, 2016, **60**, pp. 19–31

[10] Jarvis, R.A., Patrick, E.A.: 'Clustering using a similarity measure based on shared near neighbors', *IEEE Trans. Comput.*, 1973, **100**, (11), pp. 1025–1034

[11] Shamshirband, S., Amini, A., Anuar, N.B., *et al.*: 'D-ficca: a density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks', *Measurement*, 2014, **55**, pp. 212–226

[12] Madden, S.: 'Intel Berkeley research lab'. 2004

[13] Shamshirband, S., Patel, A., Anuar, N.B., *et al.*: 'Cooperative game theoretic approach using fuzzy q-learning for detecting and preventing intrusions in wireless sensor networks', *Eng. Appl. Artif. Intell.*, 2014, **32**, pp. 228–241

[14] Ahmadi, A., Shojafar, M., Hajeforosh, S.F., *et al.*: 'An efficient routing algorithm to preserve k-coverage in wireless sensor networks', *J. Supercomput.*, 2014, **68**, (2), pp. 599–623

[15] Mostafaei, H., Shojafar, M.: 'A new meta-heuristic algorithm for maximizing lifetime of wireless sensor networks', *Wirel. Pers. Commun.*, 2015, **82**, (2), pp. 723–742

[16] Hassan, A., Mokhtar, H., Hegazy, O.: 'A heuristic approach for sensor network outlier detection', *Int. J. Res. Rev. Wirel. Sens. Netw.*, 2011, **1**, (4), pp. 66–72

[17] Zhang, Y., Hamm, N.A., Meratnia, N., *et al.*: 'Statistics-based outlier detection for wireless sensor networks', *Int. J. Geograph. Inf. Sci.*, 2012, **26**, (8), pp. 1373–1392

[18] Ahmadi Livani, M., Alikhany, M., Yadollahzadeh Tabari, M., *et al.*: 'Outlier detection in wireless sensor networks using distributed principal component analysis', *J. AI Data Mining*, 2013, **1**, (1), pp. 1–11

[19] Chitradevi, N., Palanisamy, V., Baskaran, K., *et al.*: 'Efficient density based techniques for anomalous data detection in wireless sensor networks', *J. Appl. Sci. Eng.*, 2013, **16**, (2), pp. 211–223

[20] Guo, S., Zhang, H., Zhong, Z., *et al.*: 'Detecting faulty nodes with data errors for wireless sensor networks', *ACM Trans. Sens. Netw. (TOSN)*, 2014, **10**, (3), p. 40

[21] Ozkan, H., Ozkan, F., Kozat, S.S.: 'Online anomaly detection under Markov statistics with controllable type-i error', *IEEE Trans. Signal Process.*, 2016, **64**, (6), pp. 1435–1445

[22] Kamal, S., Ramadan, R.A., Fawzy, E.-R.: 'Smart outlier detection of wireless sensor network', *Facta Univ. Series: Electron. Energ.*, 2016, **29**, (3), pp. 383–393

[23] Duan, L.: 'Density-based clustering and anomaly detection'. Business Intelligence-Solution for Business Development, 2012, pp. 79–96

[24] Ester, M., Kriegel, H.-P., Sander, J., *et al.*: 'A density-based algorithm for discovering clusters in large spatial databases with noise'. KDD, 1996, vol. **96**, pp. 226–231

[25] Ankerst, M., Breunig, M.M., Kriegel, H.-P., *et al.*: 'Optics: ordering points to identify the clustering structure'. ACM Sigmod Record, 1999, vol. **28**, pp. 49–60

[26] Daszykowski, M., Walczak, B., Massart, D.L.: 'Looking for natural patterns in analytical data. 2. Tracing local density with optics', *J. Chem. Inf. Comput. Sci.*, 2002, **42**, (3), pp. 500–507

[27] Hinneburg, A., Keim, D.A.: 'An efficient approach to clustering in large multimedia databases with noise'. KDD, 1998, vol. **98**, pp. 58–65

[28] Daszykowski, M., Tran, T.N., Drab, K.: 'Revised dbscan algorithm to cluster data with dense adjacent clusters', *Chemometr. Intell. Lab. Syst.*, 2013, **120**, pp. 92–96

[29] Bressen, T.: 'Consensus decision making'. The change handbook: the definitive resource on todays best methods for engaging whole systems, 2007, pp. 212–217

[30] Luo, X.,, Chang, X.: 'A novel data fusion scheme using grey model and extreme learning machine in wireless sensor networks', *Int. J. Control Autom. Syst.*, 2015, **13**, (3), pp. 539–546

[31] Appice, A., Ciampi, A., Malerba, D.: 'Summarizing numeric spatial data streams by trend cluster discovery', *Data Min. Knowl. Discov.*, 2015, **29**, (1), pp. 84–136

[32] Efron, B., Tibshirani, R.: 'Bootstrap methods for standard errors, confidence intervals, other measures of statistical accuracy', *Stat. Sci.*, 1986, **1**, pp. 54–75

[33] Press, W.H., Teukolsky, S.A., Vetterling, W.T.*, et al.*: '*Numerical recipes in C*' (Citeseer, 1996), vol. **2**

[34] Katzgraber, H.G.: 'Random numbers in scientific computing: an introduction', in 'International Summer School Modern Computational Science' (Citeseer, 2010)