



This is a peer-reviewed, post-print (final draft post-refereeing) version of the following published document, © 2020 John Wiley & Sons, Ltd. This is the peer reviewed version of the following article: Mahmood Safaei, Maha Driss, Wadii Boulila, Elankovan A. Sundararajan, Mitra Safaei, Global outliers detection in wireless sensor networks: A novel approach integrating time-series analysis, entropy, and random forest-based classification., Software: Practice and Experience, 10.1002/spe.3020, 52, 1, (277-295), (2021), which has been published in final form at <https://doi.org/10.1002/spe.2785>. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving. and is licensed under Creative Commons: Attribution-Noncommercial 4.0 license:

**Safaei, Mahmood ORCID logoORCID: <https://orcid.org/0000-0002-3924-6927>, Ismail, Abul Samad, Chizari, Hassan ORCID logoORCID: <https://orcid.org/0000-0002-6253-1822>, Driss, Maha, Boulila, Wadii, Asadi, Shahla and Safaei, Mitra (2020) Standalone noise and anomaly detection in wireless sensor networks: A novel time-series and adaptive Bayesian-network-based approach. Software - Practice and Experience, 50 (4). pp. 428-446. doi:10.1002/spe.2785**

Official URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/spe.2785>

DOI: <http://dx.doi.org/10.1002/spe.2785>

EPrint URI: <https://eprints.glos.ac.uk/id/eprint/10687>

### **Disclaimer**

The University of Gloucestershire has obtained warranties from all depositors as to their title in the material deposited and as to their right to deposit such material.

The University of Gloucestershire makes no representation or warranties of commercial utility, title, or fitness for a particular purpose or any other warranty, express or implied in respect of any material deposited.

The University of Gloucestershire makes no representation that the use of the materials will not infringe any patent, copyright, trademark or other property or proprietary rights.

The University of Gloucestershire accepts no liability for any infringement of intellectual property rights in any material deposited but will remove such material from public view pending investigation in the event of an allegation of any such infringement.

PLEASE SCROLL DOWN FOR TEXT.

# Stand-alone noise and anomaly detection in wireless sensor networks: a novel time-series and adaptive Bayesian-network based approach

Mahmood Safaei · Abul Samad Ismail ·  
Hassan Chizari · Wadii Boulila · Shahla  
Asadi · Mitra Safaei · Amir Hussain

**Abstract** In recent years, Wireless Sensor Networks (WSN) have received worldwide attention due to their practicality in monitoring and communicating physical phenomena. A WSN consists of small sensors with limited computational and communication capabilities. Reading data in WSN is not always reliable due to open environmental factors such as noise, weak received signal strength and intrusion attacks. Sending highly noisy data has negative effects not just on data accuracy and network reliability, but also on the decision making process in the base station. The process of detecting highly noisy data is called anomaly or outlier detection. Anomaly detection or deviation detection, is one of the fundamental tasks of time series analysis that relates to predictive modeling, cluster analysis and association analysis. Anomaly detection has been widely researched in various disciplines such as statistics, data mining, machine learning, information theory, and spectral decomposition, but less so in WSN. Moreover, the challenging aspect of noise detection in WSN is related to limited computational and communication capabilities of sensors. There are only a few noise detection techniques in WSNs, but algorithms that detect noise and anomaly in real data with high level of accuracy, locally without any network communication, is very rare. Hence, the purpose of this research is to design a local time-series based data noise and anomaly detection approach for WSN. The proposed Local Outlier Detection Algorithm (LODA) is a decentralized noise detection algorithm that runs on each sensor node individually with three important features: reduction mechanism that eliminates the noneffective features, determination of the memory size of data histogram to accomplish the effective available memory, and classification for predicting noisy data. An adaptive Bayesian Network is used as the classification algorithm for prediction and identification of outliers in each sensor node locally. Results of our approach are compared to four well-known algorithms using benchmark real-life datasets, which demonstrate that LODA can achieve higher (up to 89%) accuracy in prediction of outliers in real sensory data.

**Keywords** Outlier detection · Anomaly detection · Wireless sensor network · Time series analysis

histogram to accomplish the effective available memory, and classification for predicting noisy data. An adaptive Bayesian Network is used as the classification algorithm for prediction and identification of outliers in each sensor node locally. Results of our approach are compared to four well-known algorithms using benchmark real-life datasets, which demonstrate that LODA can achieve higher (up to 89%) accuracy in prediction of outliers in real sensory data.

**Keywords** Outlier detection · Anomaly detection · Wireless sensor network · Time series analysis

## 1 Introduction

In the last few years, wireless sensor networks (WSNs) have attracted worldwide interest, in particular due to the fast development of technology that has led to the production of smart sensors. These sensors are typically small and are limited in terms of their computation unit and processing resources, and therefore are less expensive than traditional sensors. Smart sensor nodes can sense, measure, and collect data from the environment ; in addition, they can transfer sensor data to the base station where decision processes are conducted. They are low-power devices equipped with a number of sensors, power supply, processor, radio transmitter, and memory, as well as an actuator [Yick et al.(2008)Yick, Mukherjee, and Ghosal]. Smart sensor nodes can use many different types of sensors, including biological, mechanical, chemical, optical, thermal, and magnetic sensors, for measuring properties associated with the environment. Because the memory of sensor nodes is limited and they are usually implemented in areas that are not easily accessible, a radio transmitter is implemented for transferring the data to the base station through a wireless communication.

Existing WSNs can be used on land, underwater, and underground. A sensor network encounters various difficulties and limitations according to its environment. There are five different types of WSNs: multi-media, underground, underwater, terrestrial, and mobile. A WSN typically contains hundreds to thousands of low-cost wireless sensor nodes implemented in a location provided by the end user and can usually be positioned in a location of interest either deterministically or randomly [Akyildiz et al.(2002)Akyildiz, Su, Sankarasubramaniam, and Cayirci]. The selection of their implementation scheme depends very heavily on the sensor type and the application, as well as on the environment in which the sensors will operate [Younis and Akkaya(2008)]. In WSNs, reliable communication in a dense environment is one of the most important factors. It is essential that the data transmitted from terrestrial sensor nodes to the base station interact effectively. To overcome the difficulties generated by the restrictions caused by their non-rechargeable battery power, terrestrial sensor nodes are facilitated by an additional power supply, such as solar panels. In all cases, data reliability is important for WSNs. Many algorithms have been proposed for guaranteeing the reception of reliable data at a sink, and

the detection of noise, anomalies, and outliers is one of the techniques applied for this purpose.

The three important techniques applied for data collection are as follows. The first is applied in static networks. A network is defined as static when the sensor nodes are static or fixed. In a fixed sensor network, each sensor node forwards the data to the base station through one or more hops [Ma and Yang(2007)]. Therefore, the power of sensors located closer to the sink is depleted earlier than that of sensors at a greater distance. In the second data collection technique, the structure of the data collection is hierarchical. It is typically divided into the two layers: the first is the lower layer and the second is the higher layer. All the sensor nodes in the lower layer are homogeneous. The nodes in the higher layer tend to be more powerful than the nodes in the lower layer. The higher layer nodes are called cluster heads. The hierarchical topology is normally referred to as being composed of clusters. The third technique involves a mobile data collector that can be used to gather the data periodically and is employed to collect the data dynamically. The sensor nodes positioned nearer to the data collector are able to upload their data directly and those located at a greater distance from the collector can transmit their data through a relay [Rubia and ArulSelvan(2014)].

The development of data collection methods for WSNs remains in its very early stages and the particular unique features of this type of data collection necessitate unique solutions and novel approaches that differ from those for other applications. For instance, in a typical approach applied in sensor applications, such as those for target tracking, information or sensor data are processed locally and stored at the sensor node. The stored data can then be queried, later, by other sensor nodes [Wang and Liu(2011)]. When the data are collected by the sensor, all the sensing data must be accurately and correctly gathered and sent to the base station. The processing of these data requires global knowledge and is considerably more complicated than that in similar target tracking applications. In addition, this feature avoids utilizing data fusion or aggregation techniques to improve the performance of the network. Because of this, the main traffic in systems where data are collected by the sensor comprises the reported data transmitted from each sensor to the sink. There is no doubt that WSN applications need correct and accurate data to provide reliable information to the consumer, despite the current low quality and reliability of WSNs caused by their low cost and the harsh environment in which they are deployed.

In this paper, we propose a new outlier detection approach based on time series modeling on each sensor node locally without collaboration with neighbors. We first define a memory size for time series data at the local sensor. Then, we extract features representing time series concept. After identifying outlier in sensory data, we will use an adaptive Bayesian network (BN). The proposed method enables each sensor node to check the correctness of its

sensory data independently from the rest of other sensors. Therefore, no communication cost is needed. Additionally it is possible to calculate the spatial-temporal correlation in order to identify efficiently the anomaly type.

This paper is organized as follows. In the next section, we provide a summary of related research works regarding outlier detection in WSNs. Section 3 presents a brief introduction about outlier types. The proposed method is described in Section 4. Experimental results based on real-world datasets are depicted in Section 5. Section 6 explains the outcomes and the obtained results. Eventually, Section 7 provides concluding remarks and some future works.

## 2 Outlier Detection in Wireless Sensor Network

The term outlier or anomaly was first used in the field of statistics [Hodge and Austin(2004)]. There are two standard definitions of outliers, given below:

1. An outlier represents an observation that diverges to a large extent from other observations so as to give rise to doubts that it was produced by a separate method [Hawkins(1980)].
2. An outlier represents an observation (or a set of observations) that seems to be inconsistent with the rest of the data in that set [Ord(1996)].

There are various other definitions of outliers based on the particular technique used by the outlier recognition methods [Zhang et al.(2007)Zhang, Meratnia, and Havinga]. The most common definitions suggest how outliers can be determined in a specific kind of data group. Outliers in WSNs may be described as "the measurements that show significant deviation from the typical pattern of sensed data" [Kandhari et al.(2009)Kandhari, Chandola, Banerjee, Kumar, and Kandhari]. The basis of this definition is that sensor nodes in WSN are assigned to observe the practical world and hence, there may be a model that represents the typical behavior of detected data. Possible sources of outliers in the data gathered by WSNs include actual events, malicious attacks, errors and noise. If feasible, erroneous and noisy data should be rectified or removed because noise is an incidental error that does not have any true weight, but it still has a significant impact on the data analysis [Pang-Ning et al.(2006)Pang-Ning, Steinbach, and Kumar]. It is also important to determine outliers created by other origins since they may include information that is vital for the researchers.

To determine specific origins of outliers that take place in WSNs, various research topics have been examined. These topics pertain to event detection [Martincic and Schwiebert(2006), Ding et al.(2005)Ding, Chen, Xing, and Cheng, Chen et al.(2005)Chen, Lam, and Fan], fault detection [Chen et al.(2006)Chen, Kher, and Somani, Luo et al.(2006)Luo, Dong, and Huang], and intrusion detection [da Silva et al.(2005)da Silva, Martins, Rocha, Loureiro, Ruiz, and Wong, Bhuse and Gupta(2006)].

In the last few years, the issue of outlier detection has been of significant interest to researchers. There have been several studies of outlier detection in different fields, for example data mining, statistics, information theory, machine learning, statistics, and spectral decomposition [Kandhari et al.(2009)Kandhari, Chandola, Banerjee, Kumar, and Kandhari]. It has also been extensively used in several applications, for example in weather forecast, network intrusion, performance analysis, and fraud identification. In these applications, outliers' detection ensures secure functioning of network, reliability of the data and event reporting.

In WSN, the quality of measured data is especially regulated by outlier detection as it leads to enhanced validity of the data analysis that has been carried out in the presence of defective sensors and noise. This would decrease the communication costs of incorrect data and ensure that the final outcomes are not affected. Outlier identification also provides an effective means of obtaining data values that do not adhere to the typical model of sensor data that is part of the network being examined. Consequently, the values obtained are considered as incidents which leads to a change in the phenomenon being examined. Furthermore, potential network attacks carried out by competitors and enemies are identified through outlier detection, which also identifies malicious sensors that persistently create outlier values, while also enhancing the security of the network.

Detecting outlier in WSN is extremely challenging because of the characteristics of sensor network: resource limitation, costly communication, etc.

In the literature, numerous methods have been proposed; the majority of them are based on statistical approaches [Breunig et al.(2000)Breunig, Kriegel, Ng, and Sander, Saneja and Rani(2017)]. Their principal task is to approximate the sensor data distribution, that can easily be used to flag outliers by calculating probabilities, or metrics like, variance, correlations, and mean [Shahid et al.(2015)Shahid, Naqvi, and Qaisar].

Zhuang and Chen [Zhuang and Chen(2006)] proposed two outlier-detection techniques. They extract the spatio-temporal correlations of the detected measures attained by several sensor nodes. The first technique applies wavelet analysis and the second technique uses a dynamic time warping method. Both techniques need to fix a specified threshold to sense the anomalies.

For identifying an anomalies detection data, Wu et al. [Wu et al.(2007)Wu, Cheng, Ding, Xing, Liu, and Deng] applied k-nearest neighbors; where k is median among sensor node and its neighbors. The sensed data is considered as an anomaly if the determined threshold is lower than the deviation from the computed median. The detection procedure is relying on incorrect of half detected data from the neighboring.

An enhanced version of [Wu et al.(2007)Wu, Cheng, Ding, Xing, Liu, and Deng] work is proposed by [Guenterberg et al.(2007)Guenterberg, Ghasemzadeh, Loseu, and Jafari]. In this research, a temporal correlation is added to the algorithm. The algorithm uses the median of k-nearest neighbors for each sensed data and compare it with locally saved data in the corresponding sensor. This technique improves the accuracy of the detection algorithm; however it re-

quires an extra computational cost.

Sheng et al. [Sheng et al.(2007)Sheng, Li, Mao, and Jin] proposed a histogram-based technique for global outliers' detection in WSNs. Instead of sending all sensory data to base station, each sensor node will keep a summary of relevant sensed data over a sliding window. With using the summaries, the base station extracts the data distribution and filters out the typical data. Outliers are detected if the measures exceed a static threshold value. The primary disadvantage for this technique is the accidental availability (failure, shut-down, etc.) at the base station which will stop the entire analysis system. Moreover, this technique is only used in one-dimensional data where spatial distance among the sensor nodes is important.

Abid et al. [Abid et al.(2017)Abid, Masmoudi, Kachouri, and Mahfoudhi] presented a density-based method, named OPTICS, for ordering points to identify the clustering structured. This non-parametric method calculates each point of metric for reach-ability distance and labeled it as an outlying or normal measure. The dataset is divided into clusters by considering the minimum numbers of points in the neighborhood of nominated points regarding threshold-distance. The mentioned dataset is processed point by point, which needs a high-computation cost.

Barakkath Nisha et al. [Barakkath Nisha et al.(2017)Barakkath Nisha, Uma Maheswari, Venkatesh, and Yasir Abdullah] proposed a fuzzy-based approach for anomaly-detection. The dataset used in this work is divided into sets where the likenesses inside these sets are greater between peers. This study applies subtractive clustering method for outlier detection. It uses a Takagi-Sugeno fuzzy model for membership functions and selection of parameter. However, the proposed method depends on a divided clustering WSN to spot outliers and therefore cannot be used for other network architectures. Additionally, it just tackles outliers in tow-dimensions datasets and therefore it is not able to detect anomalies in the higher dimensions.

In healthcare, an application of outlier detection is proposed by Saneja and Rani [Saneja and Rani(2017)]. This approach is based on correlation and dynamic sequential minimal-optimization regression (SMO). At the first stage, a correlation feature is computed for sorting the attributes (just strongly correlated features are considered). In the second stage, outliers' sensors are distinguished applying dynamic SMO regression. Saneja and Rani have applied their technique to fasten the processing of a Hadoop MapReduce framework. In this paper, the proposed process requires a high calculation charge that is not appropriate for WSNs.

Rajasegarar et al. [Rajasegarar et al.(2006)Rajasegarar, Leckie, Palaniswami, and Bezdek] used a cluster-based method to combine sensory data into clusters. This method starts by clustering the sensor measurements and then merging clusters before sending a description of clusters to other nodes. It does not need any advanced knowledge of the data distribution but it generates a high overhead in terms of communication.

Identifying outliers may also be performed by calculating the density associated with the sensory data measures in a target area. This density calculation

can be executed in a distributed method. In [Ghalem et al.(2019)Ghalem, Kechar, Bounceur, and Euler], a technique, labeled as Local Outlier Factor (LOF) is proposed. It first marks around at minimum k-measures called "outlier metric", outlier metric dependent on the obtained level of density it adds to sensory data to identify if this measure is an outlier or possibly not. To make sure a good amount of accuracy, it might be required to perform the LOF technique with numerous values of k, which increases the the load of the computational.

### 3 Type of Outliers

Unlike a centralized approach in which the data is processed in a single place, outliers in WSNs may be detected and evaluated at distinct nodes all across the network. Regional models developed from data flows of individual nodes are entirely different from the global models because of the type of multilevel outlier detection in WSNs [Subramaniam et al.(2006)Subramaniam, Palpanas, Papadopoulos, Kalogeraki, and Gunopulos]. A certain outlier may be either local or global, based on the scope of data that has been utilized in outlier detection.

#### 3.1 Global Outliers

Global outliers are detected in a more global perspective. They awarded special interest in order is to understand the typical data features in WSNs. It is possible to recognize global outliers at distinct levels of the network, based on the network typology [Meratnia and Havinga(2010)]. When the architecture is centralized, the whole data is dispatched to the sink node in order to detect global outliers. The disadvantages of this method is that it increases the response time and generates high communication costs. Data is obtained in aggregate/clustering-based topology using the aggregator/cluster head from the nodes that fall in its range of control. The outliers are determined by the aggregator/cluster head on the basis of the data obtained. Through this method, there is an optimal-energy use and a better response time. However, this method faces the same complication as the centralized method when there are several nodes being supervised by the aggregator/cluster head. In addition, global outliers may be detected by single nodes when they have attained a copy of global estimator model from the sink node [Subramaniam et al.(2006)Subramaniam, Palpanas, Papadopoulos, Kalogeraki, and Gunopulos].

#### 3.2 Local Outliers

Local outliers are detected in single sensor nodes. Methods used for identifying local outliers try to improve the scalability and to decrease communication ex-

penses. Different applications use local-outlier detection for identifying events, such as surveillance monitoring and vehicle tracking. There are two main variations used to detect local outliers in WSNs: in the first one, the irregular values are determined by each node based on its preceding values; while the second method depends on its own preceding values and the values of every neighboring sensor nodes to collaboratively determine the irregular values. In the latter approach, the outlier detection becomes more accurate and robust, and the advantages acquired from the spatio-temporal correlations among the overall sensor data are improved in contrast to the first method.

#### 4 Proposed Local Outlier Detection Algorithm (LODA)

In this section, we explain the detail of our independent local outlier detection algorithm. The detection process runs on each sensor node individually based on time series' properties and forecasting techniques. In this paper, feature selection techniques have been chosen with respect to time series concept.

##### 4.1 LODA Features Selection

To achieve an accurate and reliable noise detection algorithm, it is important to identify the features that they have high impact. To do this, we select features that will participate to the greedy algorithm and the simulation analysis. Below the considered features for the LODA.

###### 4.1.1 Subtraction of Selected Data

Subtraction in simple language denotes the operation or the process of finding the difference between two numbers or quantities. In time series, assume that  $T$  denotes time and  $t \in T = \{t_1, t_2, t_3, \dots, t_n\}$ .  $X$  is data observed periodically in  $T$  timestamp. Hence, observed data in each time can be represented as  $X = \{x_{t_1}, x_{t_2}, x_{t_3}, \dots, x_{t_n}\}$ . The process of features' subtraction is performed by subtracting current read data from last read data. Equation 1 illustrated the definition of the subtraction operation:

$$F = \{x_{t_i} - x_{t_{i-1}}\}_{i=2, \dots, n} \quad (1)$$

Where  $F$  is the matrix of the subtraction data,  $x_{t_i}$  is the current read data by a sensor node and  $x_{t_{i-1}}$  is the last read data.

###### 4.1.2 Adaptive Kolmogorov

By considering two random samples independently  $(X_1, X_2, \dots, X_n)$  and  $(Y_1, Y_2, \dots, Y_m)$ , where  $n$  and  $m$  are respectively sizes of the two sets. Assume that  $F(x)$  and  $G(x)$  are respectively the unidentified distribution functions of the two previously sets. Hypotheses for testing are described as follow:

A: Two sided : (2)

$$\begin{aligned} H_0 &: F(x) = G(x) \text{ for each } x \\ H_1 &: F(x) \neq G(x) \text{ or at least one value of } x \end{aligned}$$

B: One sided:

$$\begin{aligned} H_0 &: F(x) \leq G(x) \text{ for each } x \\ H_1 &: F(x) > G(x) \text{ for at least one value of } x \end{aligned}$$

C: One sided:

$$\begin{aligned} H_0 &: F(x) \geq G(x) \text{ for each } x \\ H_1 &: F(x) < G(x) \text{ for at least one value of } x \end{aligned}$$

In scenario A, the hypothesis shows that there is no distinction among the distribution functions of the two populations. Both sets are able to be viewed as one set.

In scenario B, the hypothesis shows that distribution functions of the first population are smaller than those in the second population.

In scenario C, the hypothesis shows that  $X > Y$ .

#### 4.1.3 Hamming Weight

The Hamming weight (denoted by  $w(A)$ ) of a binary vector (denoted by  $A = \{a_0, \dots, a_{n-1}\}$ ) is the number of bits within the vector, which varies from 0 to  $N$ . The Hamming distance  $d(A, B)$  between two vectors  $A$  and  $B$  is the range of matching elements. Particular applications, which are essential to computer science and information need  $w(A)$  and  $d(A, B)$  to be calculated and examined for both an individual vector, or even a group of vectors. Hamming weight is applied in several fields such as combinatorial search, image and data processing, cryptography, encoding and error correction, digital signal processing, and DNA computing.

For instance,  $w(A)$  usually requires to be compared with a fixed threshold  $k$ , or with  $w(B)$ , where  $B = \{b_0, \dots, b_{Q-1}\}$  is a different binary vector, and  $Q$  and  $N$  may or may not be equal. Equation 3 describes the Hamming Weight formula.

$$\begin{aligned} V &= \{v_1, v_2, \dots, v_n\} \\ H(V) &= \sum_{1 \leq i \leq n} v_i \\ U &= \{u_1, u_2, \dots, u_m\} \\ H(V) &\geq k \quad \text{or} \quad H(V) \geq H(U) \end{aligned} \tag{3}$$

$H(V)$  is the total of 1s in the vector  $V$  that is compared to a fixed threshold  $k$  or with  $H(U)$ .  $U$  is another binary vector of arbitrary length  $m$ .

#### 4.1.4 Kurtosis Probability

In statistics and likelihood theory, kurtosis measures the "tailedness" for the likelihood distribution of the real valued random variable. In the literature, numerous interpretations exist for kurtosis. The normal way of measuring kurtosis depends on the scaled variant of the 4th time for the population or data, and it is related to tails of the distribution. Thus, the often observed characterization as "peakedness" is incorrectly recognized. With this evaluation, greater kurtosis could be the outcome of irregular extreme variances or anomalies. The kurtosis result for each univariate normal distribution is 3. Kurtosis distributions having a value smaller than 3 are assumed to be platykurtic, even though this does not indicate that the distribution is "flat-topped" as often reported. Therefore, this indicates the circulation creates lower extreme anomalies compared to the normal distribution. Uniform distribution is an example of platykurtic distribution, and can not create anomalies. Kurtosis distributions with are higher than 3 are considered leptokurtic. This later is represented by Laplace distribution. It includes tails that are asymptotically close to zero more gradually compared to a Gaussian. Equation 4 depicts the Kurtosis formula.

$$Kurt[X] = E \left[ \left( \frac{X - \mu}{\sigma} \right)^4 \right] = \frac{\mu_4}{\sigma^4} = \frac{E[(X - \mu)^4]}{(E[(X - \mu)^2])^2} \quad (4)$$

#### 4.1.5 Standard Deviation

Standard Deviation (SD) is a measure used to quantify the amount of variation or dispersion of the group of data values. A low value of SD shows that data points are close to the mean (also known as the expected value) for the set, meanwhile a higher value of SD shows that the data points are distributed in a wider range of values. The SD of the dataset, random variable, likelihood distribution or statistical population is the square root of its variance. It is algebraically easier to compute, although in practice less robust, compared to the mean absolute deviation. An important property of the SD is that, in contrast to the variance, it is explained in identical units as the data. Equation 5 shows the SD formula.

$$S_N = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (5)$$

#### 4.1.6 Variance

Variance is employed for probability distribution in statistics. While variance calculates the variability from the average. A value of variance equal to zero means that all values inside a set of numbers are similar; all variances which are non-zero would be positive numbers. When the variance is large, it shows

that numbers within the set are far from the mean, while a small variance shows the opposite. Equation 6 depicts the variance formula.

$$\text{Var}(X) = \left( \sum_{i=1}^n p_i x_i^2 \right) - \mu^2 \quad (6)$$

#### 4.1.7 Harmonic Mean

Harmonic mean is described as the value obtained when the quantity of values in the dataset is split by the sum of its reciprocals. Furthermore, it is considered as one of the measures of central tendency. Let  $x_1, x_2, \dots, x_n$  be the set of observations, then the harmonic mean is given by Equation 7.

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad x_i > 0 \text{ for all } i \quad (7)$$

Harmonic mean is applied to the group of observations having shape of fractions or values that are extreme.

#### 4.1.8 Geometric Mean

Geometric mean, often known as compounded annual growth rate or time weighted rate of return, is the mean rate of set of values calculated using the products associated with terms as shown by Equation 8.

$$\left( \prod_{i=1}^n a_i \right)^{\frac{1}{n}} = \sqrt[n]{a_1 a_2 \dots a_n} \quad (8)$$

#### 4.1.9 Mode

Mode identifies the central value of distribution, and it is associated with the variable occurring most frequently. It represents the most typical value present in series. Equation 9 describes the Mode formula.

$$\text{mode} = l + h \left( \frac{f_m - f_1}{2f_m - f_1 - f_2} \right) \quad (9)$$

Where  $l$  is lower boundary of the model class,  $h$  is the size of a model class,  $f_m$  represents the frequency corresponding to a modal class, and  $f_1$  and  $f_2$  are the frequency preceding to a modal class.

**Table 1** Adopted Notation

Name	Description	Name	Description
$rowM$	$54 \times n$ original aggregated data	Noisetdata	The percentage of total noise in data
$\mathcal{N}(0, \sigma_{\theta}^2)$	Gaussian noise	$\sigma_{\theta}^2$	Gaussian parameter
$D$	Polynomial interpolated matrix	$noisData$	Random injected noise matrix
$featSelection$	Calculated features	$P(B, C)$	Bayesian network
$f_1$	Subtraction	$f_2$	<i>Kolmogrov</i>
$f_3$	Haming weight	$f_4$	Kurtosis
$f_5$	Standard Deviation	$f_6$	Variance
$f_7$	Harmonic Mean	$f_8$	Geometric Mean
$f_9$	Mode		

## 5 LODA Implementation and Experimental Results

This section explains the simulation and the experimental results for our LODA. Matlab and R languages are used to implement codes in this paper. Figure 1 depicts steps of the simulation of our paper. The proposed is based on three main steps: 1) Dataset preparation and cleansing, 2) Simulation configuration, and 3) Simulation and result analysis. In the first step, time series data polynomial interpolation and aggregation has been conducted to achieve a reliable and accurate dataset to use in the next phases. This step is one of most important and time consuming part. Because all the results and analysis on next steps are dependent on accurate dataset. The second step in operational framework is configuration and feature selection for simulation part. The goal of this step is to prepare features and all configurations need in simulation section. Features has been selected based on literature review. The third step will describe the noise generation algorithm and percentage of noisy data also has been define in this section. Last section of the frame work in one hand is simulation and implementing all configuration in Matlab software as simulation software and on the other hand analysis the result and find optimal solution and select effective features.

The benchmark dataset used for this research is the Intel Laboratory dataset. This dataset contains information about data collected from 54 sensor nodes deployed in the Intel Berkeley Research Laboratory between February 28<sup>th</sup> and April 5<sup>th</sup>, 2004. The dataset includes a large amount of sensor reading data from the sensor nodes (around 2.3 million records). Figure 2 shows the position of the sensors in the test environment. The dataset contains the coordination of each sensor for further usage if required. Table 1 summarizes the notation adopted by our LODA.

The dataset contains many sensor data, such as temperature, humidity, and light, as well as information about sensor-node battery voltage. Table 2 presents information provided by sensors and defines their types. All the data were collected based on a 31 s time stamp interval.

Cleansing dataset is one of the critical part when time series concept is considered. In the dataset preparation step, most researchers delete inaccurate data or data with NaN values to prevent fault and error in their proposed

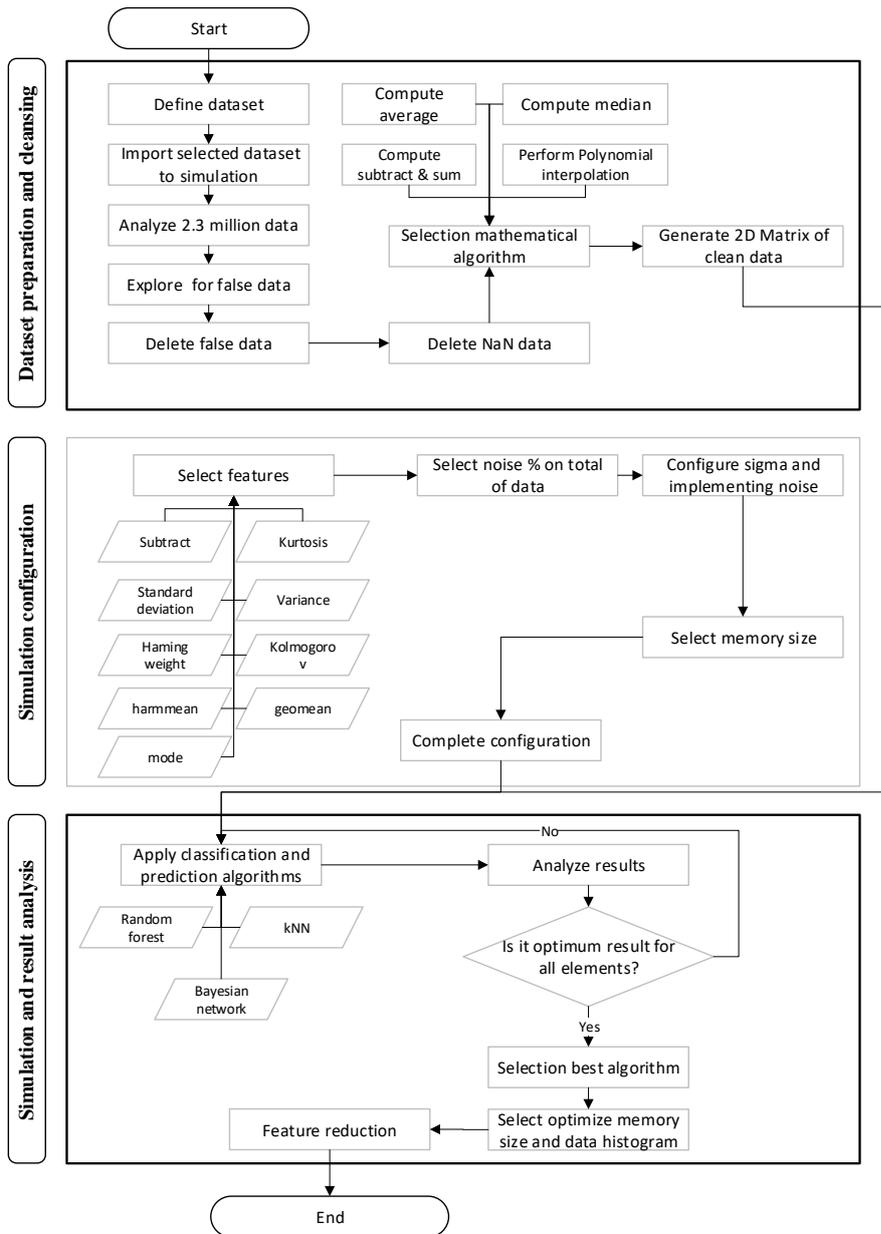


Fig. 1 Proposed Process for LODA

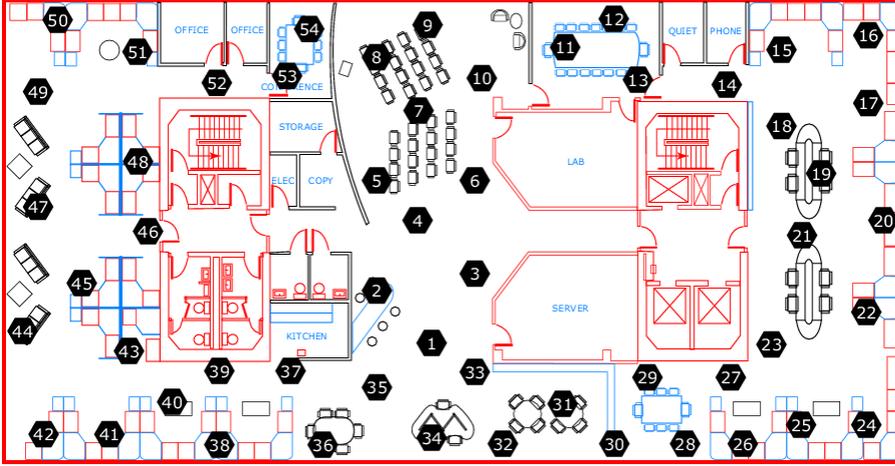


Fig. 2 Schematic of sensor positioning and numbering

Table 2 Dataset variables and their types

Variable	Type
Date	yyyy-mm-dd
Time	hh:mm:ss.xxx
Epoch	Int
moteid	Int
Temperature	Real
Humidity	Real
Light	Real
Voltage	Real

algorithms. However, deleting NaN data from the dataset, in a context of time series, can lead to wrong situations because each data in time series has an own time value. Hence, in this paper an adaptive Polynomial Interpolation has been used to replace the empty, NaN or faulty data with a predictable value.

### 5.1 Adaptive Polynomial Interpolation

In this section, we will detail the adaptive polynomial interpolation.

Let us suppose that we have  $n + 1$  points,  $x_0, x_1, \dots, x_n$ , and  $n + 1$  function values,  $f_0, f_1, \dots, f_n$ .

A polynomial denoted by  $p_n(x)$  of a degree  $n$  is defined by the Equation ??.

$$p_n(x_k) = f_k \quad \text{for } k = 0, \dots, k = n. \quad (10)$$

In our context, we suppose that the interpolating points are distinct ( $x_j \neq x_k$  if  $j \neq k$ ). The construction of the interpolating polynomial is mostly used in scientific computing using the Newton construction. This construction builds

a sequence of polynomials, each interpolates one extra point. The first polynomial is  $p_0(x)$ , which has a degree zero that is a constant. This constant is chosen so that  $p_0$  satisfies the first interpolating condition given by the Equation 10 for  $k = 0$ .

$$p_0 = f_0 \quad (11)$$

Next, finding a polynomial,  $p_1(x)$ , of degree 1 that interpolates both  $f_0$  and  $f_1$ .  $p_1(x)$  is defined as follow:

$$p_1(x) = p_0(x) + (x - x_0) \cdot c_1 \quad (12)$$

In the previous interpolation condition at  $x_0$  is not disturbed, that is  $p_1(x_0) = p_0(x_0) = f_0$ . Then the remaining used for interpolation condition,  $p_1(x_1) = f_1$ , to solve for  $c_1$ :

$$f_1 = f_0 + (x_1 - x_0) \cdot c_1 \quad (13)$$

$$c_1 = \frac{f_1 - f_0}{x_1 - x_0} \quad (14)$$

$$p_1(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) \quad (15)$$

The Equation 14 shows that the coefficient  $c_1$  is a divided difference. The construction of  $p_2$  will make the general pattern clear. First, we choose a form for  $p_2(x)$  that insures that the interpolation conditions at  $x_0$  and  $x_1$  are preserved:

$$p_2(x) = p_1(x) + (x - x_1)(x - x_0)c_2 \quad (16)$$

Then, we choose  $c_2$  to respect the final interpolation condition  $p_2(x_2) = f_2$ :

$$c_2 = \frac{\frac{f_2 - f_0}{x_2 - x_0} - \frac{f_1 - f_0}{x_1 - x_0}}{x_2 - x_1} \quad (17)$$

The general construction proceeds by induction. If a polynomial of degree  $j - 1$  that satisfies the interpolation conditions given by the Equation 10 for  $k < j$ , then we look for  $p_k(x)$  of the form:

$$p_j(x) = p_{j-1}(x) + (x - x_{j-1}) \cdots (x - x_0)c_j \quad (18)$$

Then, we use the interpolation condition  $p_j(x_j) = f_j$  to solve the  $c_j$ . If  $p_{j-1}(x)$  has a degree  $j - 1$ , then the Equation 18 gives  $p_j$  degree  $j$  because the second term has  $j$  factors involving  $x$ . If we continue in this way, eventually we construct  $p_n(x)$  that satisfies all the conditions given by the Equation 10.

Another way to construct the interpolating polynomial is through the Lagrange interpolation formula. For this purpose specific polynomials has been used:

$$m_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^n (x - x_j) \quad (19)$$

$$l_k(x) = m_k(x)/m_k(x_k) \quad (20)$$

The denominator in the Equation 20 is not zero because the points  $x_j$  are never equal to  $x_k$  if  $j \neq k$ . The polynomials  $m_k(x)$  and  $l_k(x)$  have  $n$  degrees because the products in the Equation 19 have  $n$  terms. Moreover,  $m_k(x_j) = l_k(x_j) = 0$  if  $j \neq k$ . This is the purpose of the formula illustrated by the Equation 19. The normalization of the Equation 20 makes  $l_k(x_k) = 1$ . We express the interpolating polynomial  $p_n(x)$  in terms of the  $l_k(x)$  simply as:

$$p_n(x) = \sum_{k=0}^n f_k l_k(x) \quad (21)$$

If  $x$  is one of the interpolating points, say  $x = x_m$ , then only the term on the right side of the Equation 21 with  $k = m$  is different from zero. Thus, the Equation 21 satisfies the interpolation conditions given by the Equation 10. The formula illustrated by the Equations 10, 14, and 15 together are called the Lagrange interpolation formula.

The interpolation problem as a system of linear equations uses general theorems from linear algebra. The polynomial  $p_n(x)$  is written in terms of its coefficients as follow:

$$p_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad (22)$$

The interpolation conditions given by the Equation 10 are represented as follow :

$$\begin{aligned} a_0 + a_1x_0 + \cdots + a_nx_0^n &= f_0 \\ a_0 + a_1x_1 + \cdots + a_nx_1^n &= f_1 \\ &\vdots \\ &\vdots \\ &\vdots \\ a_0 + a_1x_n + \cdots + a_nx_n^n &= f_n \end{aligned}$$

This system is written according to the following matrix :

$$Va = f \quad , \quad (23)$$

where

$$V = \begin{pmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 1 & x_n & \dots & x_n^n \end{pmatrix} \quad (24)$$

$$a = \begin{pmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_n \end{pmatrix} \quad , \quad f = \begin{pmatrix} f_0 \\ f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_n \end{pmatrix} \quad (25)$$

The matrix  $V$  is the *vander Monde* matrix. It is a reformulation of the polynomial interpolation problem. Interpolating polynomials exist and are unique if the solution of the system of linear equations (23) exists and is unique.

Since  $V$  is a square matrix, there are only two cases to get a solution for (23). Either there is a unique solution for every  $f$ , or there is some  $f$  with no solution at all. The second case cannot happen for our interpolation problem, because there is an evidence of presence of interpolating polynomial for our data (from Newton or Lagrange construction).

## 5.2 LODA Implementation

This section describes steps needed to implement our proposed LODA. All equations for the LODA presented in this paper are developed using Matlab. Bayesian Network has been selected as the outlier detection algorithm for prediction and identification of outliers in each sensor node locally.

The dataset has been sorted based on time stamp for each sensor node and saved into an  $m \times n$  matrix called rowM matrix. The rowM matrix contains 54 rows representing sensor numbers and 2.3 million columns representing sensory data as shows in Equation 26.

$$rowM = \begin{pmatrix} r_{s_1}^{t_1} & \dots & r_{s_1}^{t_n} \\ r_{s_2}^{t_1} & \dots & r_{s_2}^{t_n} \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ r_{s_i}^{t_1} & \dots & r_{s_i}^{t_n} \end{pmatrix} \quad (26)$$

Where  $r_{s_i}$  is data collected by members  $s_i$ ,  $i = \{1 \dots s\}$ ,  $s = 54$ ,  $r^{t_n}$  is data collected by members  $s_i$  in  $\Delta t_n$ . Each  $t_n$  has been defined as epoch and all sensory data from the members with same epoch number are in same columns. The Noisdata matrix depicted by the Equation 27 represents the percentage of total noise in data.

$$Noisetdata = \{n_1, n_2, \dots, n_m\} \quad (27)$$

Where  $n_m$  is the percentage of noise compared to the total of data for each sensor node. The values of  $n$  is a row matrix  $n = \{10, 15, 20.25, 30, 35, 40, 45, 50, 55, 60, 65, 70\}$ .

Noiselevel matrix contains the noise level applied to each sensory data in our simulation. In the literature, to achieve better accuracy of algorithm the Noiselevel matrix is defined by the Equation 28.

$$\mathcal{N}(0, \sigma_\theta^2) \quad (28)$$

$$\sigma_\theta^2 = \{0.001, 0.01, 0.1, 0.5, 1, 2.5, 5, 10, 15, 20, 50, 100\}$$

Where  $\mathcal{N}(0, \sigma_\theta^2)$  is the Gaussian noise and  $\sigma_\theta^2$  is the Gaussian parameter. The parameter for noise level will be changed in each iteration of the simulation to

get better output of our algorithm.

Additionally, since the memory resource in sensor node is too small, it is essential to identify the best memory size to achieve best accuracy in outlier prediction. The size of memory denoted by MemSize is defined as  $MemSize = \{10, 20, 30, 40, 50, 60\}$  where each number in MemSize matrix demonstrates the size of the data histogram.

The output of the Polynomial Interpolation function is a data matrix denoted by  $D$ . This latter will be used in our simulation and it is represented by the Equation 29.

$$D = \begin{pmatrix} d_{s_1}^{t_1} & d_{s_1}^{t_2} & \dots & d_{s_1}^{t_n} \\ d_{s_2}^{t_1} & d_{s_2}^{t_2} & \dots & d_{s_2}^{t_n} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ d_{s_i}^{t_1} & d_{s_i}^{t_2} & \dots & d_{s_i}^{t_n} \end{pmatrix} \quad (29)$$

Noise-generation algorithm takes as input the matrix  $D$ ,  $n_i$  from Noisetdata and  $\sigma_{\theta_j}^2$  from Noiselevel matrix. Noise-generation algorithm chooses randomly data from the matrix  $D$  and adds noise to this data based on Noiselevel matrix. The output of this function is NoisData matrix as shows by the Equation 30.

$$noisData = \begin{pmatrix} d_{s_1}^{t_1} & \cdot & \cdot & \cdot & d_{s_1}^{t_n} \\ d_{s_2}^{t_1} & \cdot & d_{s_{2no}}^{t_m} & \cdot & d_{s_2}^{t_n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ d_{s_{ino}}^{t_m} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & d_{s_{ino}}^{t_m} & \cdot \\ d_{s_i}^{t_1} & \cdot & \cdot & \cdot & d_{s_i}^{t_n} \end{pmatrix} \quad (30)$$

Where  $i \in \{1, 2, \dots, 13\}$  and  $j \in \{1, 2, \dots, 12\}$ .

NoisData matrix will be used to compute features required by LODA. The following features are considered in this paper:

- (i) Subtract  $\rightarrow f_1$
- (ii) Kolmogrov  $\rightarrow f_2$
- (iii) Haming weight  $\rightarrow f_3$
- (iv) Kurtosis  $\rightarrow f_4$
- (v) Standard Deviation  $\rightarrow f_5$
- (vi) Variance  $\rightarrow f_6$
- (vii) Harmeian  $\rightarrow f_7$
- (viii) Geomean  $\rightarrow f_8$
- (ix) Mode  $\rightarrow f_9$

All these features have been calculated on the noise matrix. The value 10 is chosen as a starting point to calculate features after reading data from sensor node. Computed features are represented by the matrix feaMatrix illustrated by the Equation 31. This matrix will be used by the prediction algorithm function.

$$featSelection = \begin{pmatrix} f_{1s_1} & f_{2s_1} & f_{3s_1} & f_{4s_1} & f_{5s_1} & f_{6s_1} & f_{7s_1} & f_{8s_1} & f_{9s_1} \\ f_{1s_2} & f_{2s_2} & f_{3s_2} & f_{4s_2} & f_{5s_2} & f_{6s_2} & f_{7s_2} & f_{8s_2} & f_{9s_2} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ f_{1s_n} & f_{2s_n} & f_{3s_n} & f_{4s_n} & f_{5s_n} & f_{6s_n} & f_{7s_n} & f_{8s_n} & f_{9s_n} \end{pmatrix} \quad (31)$$

$$P(A|B, C) = P(A, B, C)/P(B, C) \quad (32)$$

$$P(B, C) = \sum_d P(B|D)P(C|D)P(D = d) \quad (33)$$

## 6 Discussion and Results

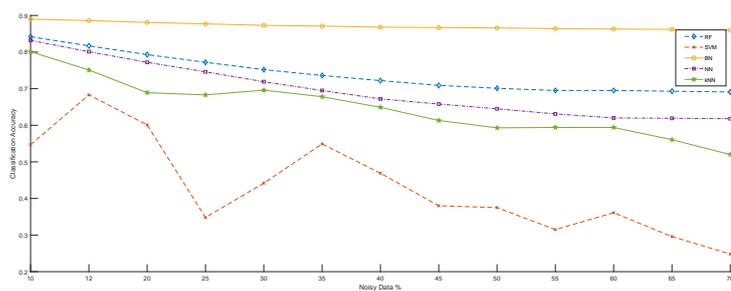
LODA is developed in order to classify outliers and anomalies of each individual sensor-data while considering the time concept. Therefore, LODA needs to consider three important factors which include memory size of the historical data, percentage of noisy data, and time-series features' selection.

To identify the best memory size, a statistical randomness analysis, time series analysis, and simulation have been used. Simulation result shows that 10 is the best size for historical memory among numbers 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, and 60.

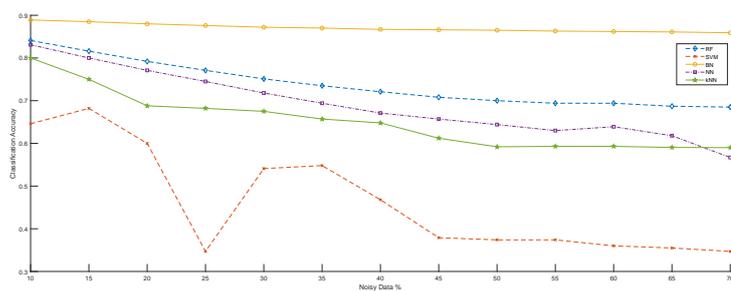
For accuracy evaluation, results obtained by the proposed algorithm are compared with four algorithms that have been widely used in the literature. The four outlier detection algorithms are support vector machine (SVM), random forest (RF), k-nearest neighbors (kNN), and neural network (NN). Figures 3, 4 and 5 depict comparisons between LODA and the four algorithms SVM, RF, kNN and NN while considering a memory size of 10 and for noisydata = {10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70}. We note that LODA outperforms the four other algorithms; it has best identification accuracy for different memory size (10, 20 and 30). In addition, LODA is not affected by the noise added to the data and it maintains best outlier detection accuracy for all percentage of noisy data. The obtained results show that the proposed LODA achieves good performances even with low memory size.

In order to better evaluate the robustness of LODA, we propose to compare the outlier detection accuracy of our algorithm while increasing the level of noise. Figure 6 shows that the LODA is not sensitive to the increase of noise level.

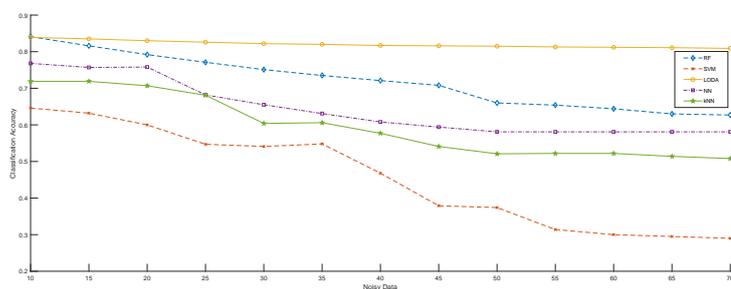
Figure 7 presents a 3D illustration of the LODA outlier detection accuracy while changing the memory size and the noise level. We note that at the beginning of the graph, LODA achieves the highest level of accuracy with the hottest color code and graph slope.



**Fig. 3** Comparison of outlier detection accuracy between LODA, SVM,RF, kNN, and NN with memory size of 10 and for noisydata = {10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70}

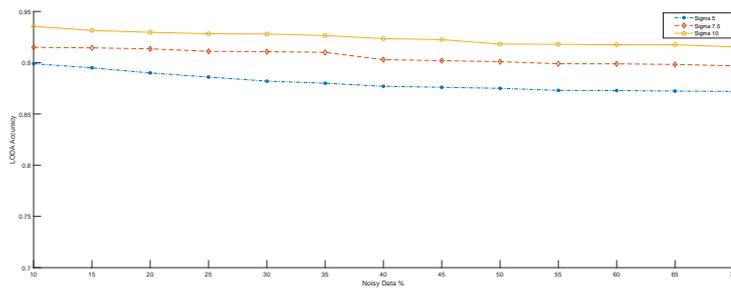


**Fig. 4** Comparison of outlier detection accuracy between LODA, SVM,RF, kNN, and NN with memory size of 20 for noisydata = {10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70}

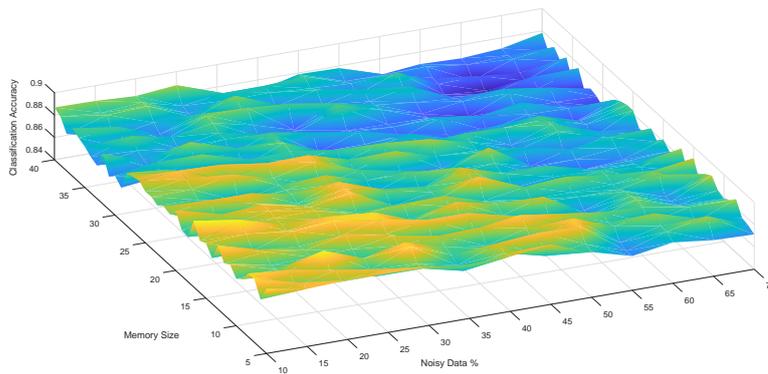


**Fig. 5** Comparison of outlier detection accuracy between LODA, SVM,RF, kNN, and NN with memory size of 30 for noisydata = {10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70}

The reduction algorithm identified  $f_1$ ,  $f_5$ ,  $f_6$ ,  $f_7$ , and  $f_2$  as most effective features in our case. Figure 8 depicts the internal correlation coefficient between the selected features. Rows represent features and columns represent correlation between features [Boulila et al.(2017)Boulila, Ayadi, and Farah]. The low correlation coefficient here is the key point of the feature selection in LODA. Indeed, when correlation between features is very low; it can be



**Fig. 6** Comparison of LODA outlier detection accuracy with memory size of 10 and  $\sigma = \{5, 7.5, 10\}$  for noisydata =  $\{10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70\}$



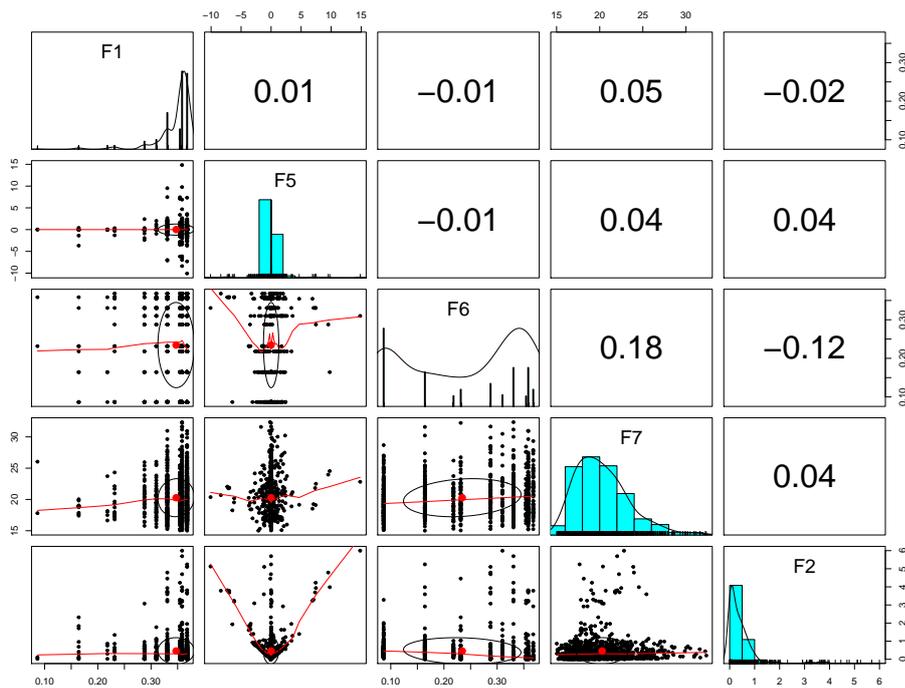
**Fig. 7** 3D illustration and relation between LODA accuracy, memory size between 5 to 40 and percentage of total noisy data 10% to 70%

considered from a mathematical aspect that there is no correlation between features.

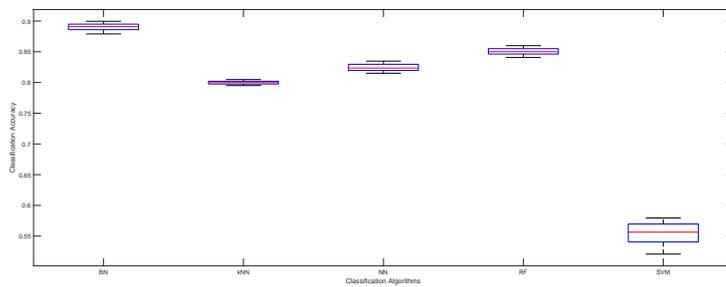
outlier detection accuracy obtained in this paper shows good performance of our proposed LODA. It achieves an accuracy of 88.9% based on a memory size equal to 10 and with noisy data between 10% to 70% of the total data in dataset.

Performance of LODA are also evaluated using two measures namely confidence interval (CI) and P-value. CI shows that in 95% of repetition times, LODA achieves an accuracy between 88.3% and 89.8%, and only in 5% of time the accuracy will be beyond these number. The second measure, P-value, determines the significance of obtained results. A small p-value (typically 0.05) indicates good results. In our case, P-Value for LODA is 0.00369, which is below the 0.05 and hence indicates good performance of LODA.

Figure 9 depicts a comparison of the mean and the standard deviation of the outlier detection accuracy between LODA, kNN, NN, RF, and SVM. In this



**Fig. 8** Correlation matrix between selected features  $f_1$ ,  $f_5$ ,  $f_6$ ,  $f_7$ , and  $f_2$



**Fig. 9** Median and standard deviation of outlier detection accuracy between LODA, kNN, NN, RF, and SVM based on simulation repetition with memory size equal to 10 and noisy data equal to 10%

example, we consider a memory size equal to 10 and a percentage of noisy data equal to 10%. We note that LODA has the highest accuracy for predicting local outlier.

## 7 Conclusion

Sensor data errors are very typical in wireless sensor networks data collection. Identifying such errors is occasionally a difficult process. This paper offers a probabilistic local outlier detection technique based on time-series based approach. An adaptive Bayesian Network is used as a outlier detection algorithm for the prediction and the identification of outliers in each sensor node locally. We evaluated our technique with benchmark real-life datasets, showing promising outcome in terms of accuracy and performance in comparison with some existing well known algorithms. The proposed technique also demonstrates resistance to outlier polluted training datasets. The proposed approach is energy efficient. In addition, the main advantage of the proposed approach is that no network communication is required during the whole process of outlier detection, in contrary of most existing detection models that need small communication with one or more neighbors.

Future works in this direction can aim to predict the real value of outlier data, based on memory size of the data and also time series forecasting. State of the art in machine learning, such as deep and reinforcement algorithms approaches can also be exploited [Mahmud et al.(2018)Mahmud, Kaiser, Hussain, and Vassanelli] [Zhang et al.(2018)Zhang, Huang, Zhang, and Hussain]. Another important perspective related to the current work will focus in integrating uncertainty modeling to LODA to improve the outlier detection in WSNs [Farah et al.(2008)Farah, Boulila, Ettabaa, Solaiman, and Ahmed, Ferchichi et al.(2017)Ferchichi, Boulila, and Farah].

## References

- [Abid et al.(2017)Abid, Masmoudi, Kachouri, and Mahfoudhi] Abid A, Masmoudi A, Kachouri A, Mahfoudhi A (2017) Outlier Detection in Wireless Sensor Networks Based on OPTICS Method for Events and Errors Identification. *Wireless Personal Communications* 97(1):1503–1515, DOI 10.1007/s11277-017-4583-7
- [Akyildiz et al.(2002)Akyildiz, Su, Sankarasubramaniam, and Cayirci] Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. *Computer Networks* 38(4):393–422, DOI 10.1016/S1389-1286(01)00302-4
- [Barakkath Nisha et al.(2017)Barakkath Nisha, Uma Maheswari, Venkatesh, and Yasir Abdullah] Barakkath Nisha U, Uma Maheswari N, Venkatesh R, Yasir Abdullah R (2017) Fuzzy-Based Flat Anomaly Diagnosis and Relief Measures in Distributed Wireless Sensor Network. *International Journal of Fuzzy Systems* DOI 10.1007/s40815-016-0253-2
- [Bhuse and Gupta(2006)] Bhuse V, Gupta A (2006) Anomaly Intrusion Detection in Wireless Sensor Networks. *Journal of High Speed Networks* Volume 15(Number 1/2006):33–51, DOI 10.1007/978-3-540-77871-4\_4
- [Boulila et al.(2017)Boulila, Ayadi, and Farah] Boulila W, Ayadi Z, Farah IR (2017) Sensitivity analysis approach to model epistemic and aleatory imperfection: Application to land cover change prediction model. *Journal of Computational Science* 23:58 – 70
- [Breunig et al.(2000)Breunig, Kriegel, Ng, and Sander] Breunig M, Kriegel HP, Ng R, Sander J (2000) LOF: Identifying density-based local outliers. *SIGMOD Record (ACM Special Interest Group on Management of Data)* DOI 10.1145/335191.335388, 342009.335388
- [Chen et al.(2006)Chen, Kher, and Somani] Chen J, Kher S, Somani A (2006) Distributed Fault Detection of Wireless Sensor Networks. *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks* pp 65–72, DOI 10.1145/1160972.1160985

- [Chen et al.(2005)Chen, Lam, and Fan] Chen Q, Lam KY, Fan P (2005) Comments on "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks". *IEEE Transactions on Computers* 54(9):1182–1183, DOI 10.1109/TC.2005.140
- [Ding et al.(2005)Ding, Chen, Xing, and Cheng] Ding M, Chen D, Xing K, Cheng X (2005) Localized Fault-Tolerant Event Boundary Detection in Sensor Networks. 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2005) 2(C):902–913, DOI 10.1109/INFCOM.2005.1498320
- [Farah et al.(2008)Farah, Boulila, Ettabaa, Solaiman, and Ahmed] Farah IR, Boulila W, Ettabaa KS, Solaiman B, Ahmed MB (2008) Interpretation of multisensor remote sensing images: Multiapproach fusion of uncertain information. *IEEE Transactions on Geoscience and Remote Sensing* 46(12):4142–4152
- [Ferchichi et al.(2017)Ferchichi, Boulila, and Farah] Ferchichi A, Boulila W, Farah IR (2017) Propagating aleatory and epistemic uncertainty in land cover change prediction process. *Ecological Informatics* 37:24 – 37
- [Ghalem et al.(2019)Ghalem, Kechar, Bounceur, and Euler] Ghalem SK, Kechar B, Bounceur A, Euler R (2019) A probabilistic multivariate copula-based technique for faulty node diagnosis in wireless sensor networks. *Journal of Network and Computer Applications* 127(February 2018):9–25, DOI 10.1016/j.jnca.2018.11.009
- [Guenterberg et al.(2007)Guenterberg, Ghasemzadeh, Loseu, and Jafari] Guenterberg E, Ghasemzadeh H, Loseu V, Jafari R (2007) Separating the Wheat from the Chaff: Practical Anomaly Detection schemes in Ecological Applications of Distributed Sensor Networks. In: *Distributed Computing in Sensor Systems*, DOI 10.1007/978-3-540-73090-3\_15
- [Hawkins(1980)] Hawkins DM (1980) *Identification of Outliers*. Springer Netherlands, Dordrecht, DOI 10.1007/978-94-015-3994-4
- [Hodge and Austin(2004)] Hodge VJ, Austin J (2004) A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review* 22(1969):85–126, DOI 10.1007/s10462-004-4304-y, arXiv:1011.1669v3
- [Kandhari et al.(2009)Kandhari, Chandola, Banerjee, Kumar, and Kandhari] Kandhari R, Chandola V, Banerjee A, Kumar V, Kandhari R (2009) Anomaly detection. *ACM Computing Surveys* 41(3):1–6, DOI 10.1145/1541880.1541882
- [Luo et al.(2006)Luo, Dong, and Huang] Luo X, Dong M, Huang Y (2006) On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers* 55(1):58–70, DOI 10.1109/TC.2006.13
- [Ma and Yang(2007)] Ma M, Yang Y (2007) SenCar: an energy-efficient data gathering mechanism for large-scale multihop sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 18(10)
- [Mahmud et al.(2018)Mahmud, Kaiser, Hussain, and Vassanelli] Mahmud M, Kaiser MS, Hussain A, Vassanelli S (2018) Applications of deep learning and reinforcement learning to biological data. *IEEE Transactions on Neural Networks and Learning Systems* 29(6):2063–2079
- [Martincic and Schwiebert(2006)] Martincic F, Schwiebert L (2006) Distributed Event Detection in Sensor Networks. In: *2006 International Conference on Systems and Networks Communications*, p 43, DOI 10.1109/ICSNC.2006.32
- [Meratnia and Havinga(2010)] Meratnia N, Havinga P (2010) Outlier Detection Techniques for Wireless Sensor Networks: A Survey. *IEEE Communications Surveys & Tutorials* 12(2):159–170, DOI 10.1109/SURV.2010.021510.00088
- [Ord(1996)] Ord K (1996) Outliers in statistical data. *International Journal of Forecasting* 12(1):175–176, DOI 10.1016/0169-2070(95)00625-7
- [Pang-Ning et al.(2006)Pang-Ning, Steinbach, and Kumar] Pang-Ning T, Steinbach M, Kumar V (2006) *Introduction to data mining*. Library of Congress p 796, DOI 10.1016/0022-4405(81)90007-8
- [Rajasegarar et al.(2006)Rajasegarar, Leckie, Palaniswami, and Bezdek] Rajasegarar S, Leckie C, Palaniswami M, Bezdek JC (2006) Distributed anomaly detection in wireless sensor networks. In: *2006 IEEE Singapore International Conference on Communication Systems, ICCS 2006*, DOI 10.1109/ICCS.2006.301508
- [Rubia and ArulSelvan(2014)] Rubia, ArulSelvan M (2014) A Survey on Mobile Data Gathering in Wireless Sensor Networks - Bounded Relay. *International Journal of Engineering Trends and Technology* 7(5):205–208, DOI 10.14445/22315381/IJETT-V7P247

- [Saneja and Rani(2017)] Saneja B, Rani R (2017) An efficient approach for outlier detection in big sensor data of health care. *International Journal of Communication Systems* DOI 10.1002/dac.3352
- [Shahid et al.(2015)Shahid, Naqvi, and Qaisar] Shahid N, Naqvi IH, Qaisar SB (2015) One-class support vector machines: analysis of outlier detection for wireless sensor networks in harsh environments. *Artificial Intelligence Review* DOI 10.1007/s10462-013-9395-x
- [Sheng et al.(2007)Sheng, Li, Mao, and Jin] Sheng B, Li Q, Mao W, Jin W (2007) Outlier detection in sensor networks. *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing - MobiHoc '07* pp 219–228, DOI 10.1145/1288107.1288137
- [da Silva et al.(2005)da Silva, Martins, Rocha, Loureiro, Ruiz, and Wong] da Silva APR, Martins MHT, Rocha BPS, Loureiro AaF, Ruiz LB, Wong HC (2005) Decentralized intrusion detection in wireless sensor networks. *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks* pp 16–23, DOI 10.1145/1089761.1089765
- [Subramaniam et al.(2006)Subramaniam, Palpanas, Papadopoulos, Kalogeraki, and Gunopulos] Subramaniam S, Palpanas T, Papadopoulos D, Kalogeraki V, Gunopulos D (2006) Online outlier detection in sensor data using non-parametric models. *VLDB '06: Proceedings of the 32nd international conference on Very large data bases* pp 187–198, DOI <http://www.vldb.org/conf/2006/p187-subramaniam.pdf>
- [Wang and Liu(2011)] Wang F, Liu J (2011) Networked Wireless Sensor Data Collection: Issues, Challenges, and Approaches. *IEEE Communications Surveys & Tutorials* 13(4):673–687, DOI 10.1109/SURV.2011.060710.00066
- [Wu et al.(2007)Wu, Cheng, Ding, Xing, Liu, and Deng] Wu W, Cheng X, Ding M, Xing K, Liu F, Deng P (2007) Localized outlying and boundary data detection in sensor networks. *IEEE Transactions on Knowledge and Data Engineering* DOI 10.1109/TKDE.2007.1062
- [Yick et al.(2008)Yick, Mukherjee, and Ghosal] Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. *Computer Networks* 52(12):2292–2330, DOI 10.1016/j.comnet.2008.04.002, 1011.1529
- [Younis and Akkaya(2008)] Younis M, Akkaya K (2008) Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks* 6(4):621–655, DOI 10.1016/j.adhoc.2007.05.003
- [Zhang et al.(2018)Zhang, Huang, Zhang, and Hussain] Zhang S, Huang K, Zhang R, Hussain A (2018) Learning from few samples with memory network. *Cognitive Computation* 10(1):1522
- [Zhang et al.(2007)Zhang, Meratnia, and Havinga] Zhang Y, Meratnia N, Havinga P (2007) A taxonomy framework for unsupervised outlier detection techniques for multi-type data sets. *Computer* 49(3):355–363
- [Zhuang and Chen(2006)] Zhuang Y, Chen L (2006) In-network Outlier Cleaning for Data Collection in Sensor Networks. *Workshop in VLDB*