A probabilistic multivariate Copula-based technique for faulty node diagnosis in Wireless Sensor Networks

Sanaa Ghalem, Bouabdellah Kechar, Ahcène Bounceur, Reinhardt Euler

A probabilistic multivariate Copula-based technique for faulty node diagnosis in Wireless Sensor Networks

Sanaa Kawther Ghalem[a,*], Bouabdellah Kechar[a], Ahcène Bounceur[b], Reinhardt Euler[b]

[a] *Industrial computing and networks Laboratory (RIIR), Computer science Department, University of Oran1, Ahmed Benbella*
[b] *CNRS Lab-STICC Laboratory, UMR 6285, University of Western Britanny, 20 Avenue Victor Le Gorgeu, 29238, Brest, France*

Abstract

Wireless sensor networks (WSNs) find extensive applications in various sensitive domains such as tracking, monitoring, environmental data collection and border surveillance. In these cases, the collected data are considered as a critical resource and used to detect any anomalies or abnormal behavior, providing information about an occurring event or a node failure. An outlier detection process must be set up to ensure the proper functioning of the monitoring system. The existing approaches are limited by assumptions on a specific distribution or a predefined data range of the collected data. Often these assumptions do not hold in practice, the data distribution is not known or determining reliable upper and lower bounds for the set of data is not possible. To overcome this, we propose a new copula-based probabilistic multivariate outlier detection method for faulty node detection in wireless sensor networks (WSNs). The joint probability density function of the copula is constructed considering dependency among the captured $n-$sensed measures without making any assumptions on the distribution of the collected data. The samples having probabilities violating a predetermined control limit are classified to be faulty. The performance of the proposed technique is observed to be better than the existing statistical methods.

*Corresponding author. Tel.: +213 7 83 19 31 15; Fax.: +213 41 58 19 41.
Email address: ghalemsanaa@gmail.com (Sanaa Kawther Ghalem)

Keywords: Anomalies; Outlier detection; Probabilistic; Multivariate; Copula; Wireless Sensor Network; Dependency.

Introduction

In the last decade, technological advances in the field of wireless communication and micro-electro-mechanical systems (MEMS) have led to the emergence and widespread use of wireless sensor networks (WSNs) [1] [2]. They can be described as a group of small, low-cost wireless-enabled devices, with limited power. Their deployment provides a monitoring service in an efficient manner. Originally driven by military applications [3] [4] such as battlefield surveillance, their development have witnessed in recent years the expansion to other critical applications like biomedical health monitoring [5], environmental monitoring [6], underground communication and monitoring systems [7], itinerant applications Deploy and forget [8], irrigation facilities management [9]. Generally, autonomous sensors are spatially distributed at different locations to monitor physical or environmental condition, such as temperature, humidity, pollution, gas, sound, vibration or pressure, etc. However, the usefulness of WSNs for this type of application is primarily determined by the dependability on the collected data. The quality of a dataset may be affected by noise/error or duplicated data. On one hand, if a WSN fails to report an occurring event it prevents from carrying out the right decision. On the other hand, it can also report erroneous data considered as outliers, which can lead to unnecessary or even harmful actions, with a direct impact on human life, health, and security.

Critical and sensitive applications require efficient outlier detection mechanisms that allow to distinguish unusual natural events from incorrectly sensed measures as early as possible. For this purpose and a good decision making, an outlier detection process must be set up to ensure the correctness of the sensed data. These data must reflect the real state of the environment in the most faithful way and they should be treated quickly to enable correct decisions. Any decision based on erroneous data may cause wrong, harmful or costly actions.

To ensure the reliability of the sensed data, it is mandatory to detect erroneous measurements quickly. Generally considered as a deviation from the normal data model, these diverging measures can be generated by several factors, like for example, hardware damage or ageing, malicious attacks, environment (hot or cold), etc. An outlier can be generated from a malicious neighboring node affecting the security of the network, a case which, however, is beyond the scope of this paper. We only consider the outliers caused by natural events and errors, which can be described as follows:

- Natural events: this kind of outliers has a small probability to occur. In general, they indicate a significant change of the natural environment (e.g., temperature, humidity, pollution, etc.) and they can be useful to detect or even predict phenomena [10] such as an earthquake, forest fires, tsunamis, flooding, etc.

- Errors: this kind of outliers is considered to be quite common and local: it may be caused by the dysfunction, at the hardware or software level, of a sensor node, producing values that diverge significantly from the expectedly normal ones [11].

In this paper, we propose a new outlier detection approach based on multivariate statistical modeling using copulas. We first define a non-faulty distribution model that fits the typical behavior of the sensed data. Then, we estimate the deviation from the usual correlation structure in the multi-dimensional space. After to fix a certain threshold, it will then be possible to determine whether a new sensed value is an outlier or not.

Note that such a deviation is characterized by a low probability which is extracted from the defined model. The proposed approach allows for each sensor node to test the correctness of its sensed data independently from the rest of the network. Hence, no communication overhead is required. It is also possible to calculate the spatial-temporal correlation in order to determine efficiently the type of the anomaly (i.e., a natural event or an error).

The rest of the paper is organized as follows. In the next section, we give a summary of the most important research work concerning outlier detection in WSNs. A brief introduction to copula theory is presented in Section 3. In Section 4, we describe the proposed approach. Section 5 presents experimental results based on real-world datasets. Finally, Section 6 provides concluding remarks and some future work.

Outlier detection in WSNs

The concept of an outlier was originally used in the field of statistics [12] where it is referred to as a deviant or an anomaly. In [13], Johnson et al. define it as an observation that is inconsistent or which deviates significantly from the set of samples in which it occurs. In [14], Barnett et al. provide a similar definition:*"An outlier is an observation, or a subset of observations, which appears to be inconsistent with the remainder of that set"*. Similarly, Hawkins et al. [15] give a general definition to deal with various types of methods and data, where the outlier is described as an observation, or a subset of observations, that diverges so much from other observations that it is considered as a set generated by a mechanism other than the considered one.

Outlier detection has become an emerging branch of research in WSNs since it guarantees the event reporting, improves the data analysis and ensures the data quality. This task is very challenging due to the WSN characteristics: resource constraints, high communication cost, etc. Numerous approaches have been proposed most of which are based on statistical methods [16] [17] [18]. Their principal task is to approximate the sensor data distribution, which can be used to flag outliers by calculating probabilities, or metrics like mean, variance, quantile, correlations, etc.

In [19], two outlier cleaning techniques are presented. They exploit the spatio-temporal correlations of the sensed measures obtained from many sensor nodes. The first one uses wavelet analysis and the second one a dynamic time warping (DTW) method. Both of them require to fix a certain threshold to

detect the outliers.

Authors in [20] have proposed an algorithm to detect an outlying sensed measure, where each sensor node computes the median of its k nearest neighbors. The sensed measure is considered as an outlier if the deviation from the determined median is greater than a defined threshold. This method generates an important communication cost, and the detection process is biased if half of the neighboring sensed measures are erroneous.

The work in [21] is an improved version of a previous work [20], in which a temporal correlation is added to the previous detection algorithm. Any new sensed measure is compared to the median of its k nearest neighbors, and to the history of existing sensed measures saved locally in the corresponding sensor. While the accuracy of this method is improved, the algorithm requires an extra computational cost.

A histogram-based method to detect global outliers in WSN data collection applications has been proposed in [22]. Instead of sending all sensed data to the sink, each sensor node will maintain a summary of pertinent measures over a sliding window. Using these summaries, the sink extracts the data distribution and filters out the normal data. Outliers are pointed out if the measures exceed a fixed threshold value. The main drawback of this method is the accidental availability (e.g., shut-down, failure, etc.) at the sink which will stop the whole diagnosis system. Furthermore, this method is only applied in one-dimensional data outlier detection where spatial proximity among sensors is important.

The authors in [23] propose a density-based method ordering points to identify a clustering structure, called OPTICS. This nonparametric method computes for each point a metric of reachability distance and depending on the resulting value the point is labeled as a normal instance or an outlying measure. This divides the dataset into clusters with a minimum number of points in the neighbourhood of selected points with regard to a threshold-distance. The dataset is processed point by point which requires a high computation cost.

A fuzzy-based anomaly detection is proposed in [24]. This work uses a subtractive clustering method to point out outliers. The dataset is partitioned

into groups where the similarities inside the groups are bigger among its peers; then the anomaly detection is performed with the help of a Takagi–Sugeno fuzzy model for parameter selection and membership functions. This method relies on a distributed clustering wireless sensor network to spot anomalies and thus cannot be applied to other network architectures. Moreover, this only tackles the anomalies in 2d datasets and thus cannot detect outliers in higher dimensions.

In [17], an anomaly detection for healthcare applications is proposed, which is based on dynamic sequential minimal optimization regression (SMO) and correlation. In a first step, a correlation factor is calculated to sort the attributes (only the strongly correlated attributes are selected). Then in a second step, anomalous sensors are detected using dynamic SMO regression. Mainly designed for big data, the authors have implemented this technique on a Hadoop MapReduce framework to fasten the processing. A prediction model is constructed over a sliding window, the model must be kept updated and the detection is based on the difference between the predicted value and the received one. This mechanism generates a high computation cost which is not suitable for WSNs.

A cluster-based technique is used in [25], where the sensed measures are merged into clusters with a predefined width before being compared to those of other sensor nodes. This technique does not require any knowledge of the data distribution but generates a high communication overhead.

Detecting outliers can also be done by measuring the density of the sensed measures in a given area. This density calculation can be performed in a distributed way. In [16], a method, called Local Outlier Factor (LOF) and based on this principle, has been presented. It first draws a circle around at least k measures and based on the obtained level of density it assigns a parameter called "outlier metric" to each measure which will be used to decide if this measure is an outlier or not. To ensure a good level of accuracy, it might be necessary to execute the LOF method with many values of k, which increases the computational cost.

Many existing papers use the Principle Component Analysis in WSN outlier detection [26]. This method is useful when combined with a distributed

spatiotemporal model. To detect an outlier, this method compares the sensed measures with those of the neighbors, which is computationally expensive.

A Kernel density estimator based technique has been proposed in [27] for on-line deviation detection in real-time streaming of sensor data in wireless sensor networks, using a hierarchical network composed of low and high-capacity sensors according to processing power and communication range. A model of the most recent captured data in a sliding window is built using a Kernel density estimator. Thus, every new sensed measure will be marked as an outlier if it deviates significantly from the estimated data distribution. The low-capacity sensors are used to detect local outliers while high-capacity sensors are used for more spatially dispersed outliers by using an aggregation of the low-capacity sensors. The data distribution is estimated using the Kernel density. This method relies mainly on the definition of a threshold, which represents an important drawback since setting up a threshold for a d-dimensional dataset can be very tricky. Furthermore, the data model requires a constant update to be accurate, generating a computational cost as a consequence. A similar approach, based on the estimation of the distribution of the sensed data using the Kernel density function, has been presented in [28]. However, it does not deal with multidimensional data.

In [29], an outlier detection approach based on the temporal correlation has been presented, in which the network is represented by a specific tree topology. Any node can be either a leaf or a parent. The detection process is composed of two stages. In the first stage, each leaf marks its data as outlier or inlier candidates using the estimation of the probability distribution of the data and sends the outlier candidates to the parent node. In the second stage, each parent node decides to accept or reject the incoming candidates based on the data received from the leaves. This approach works only if a tree communication topology is maintained. Its communication overhead is important.

In [30], an outlier detection based on a naive distance similarity metric is proposed, where each sensor node uses this metric to identify local outliers in order to broadcast them to the neighboring nodes for verification. The neigh-

boring nodes repeat the same operation until the network validates the existence or not of global outliers. This method is energy consuming and not applicable to large sensor networks.

In [18], a one class support vector machine SVM outlier detection method is proposed. In this kind of classifier all training instances are considered to belong to the normal class, based on that a boundary is learned, and any test instance beyond this boundary is labeled as an outlier. Furthermore, a Kernel function is used to estimate the dot products among the mapped vectors in the feature space to find the hyperplane. This classifier can only be applied to 2-d datasets.

In the majority of the proposed outlier detection approaches there is a significant loss of generality when the data are assumed to have a Gaussian distribution. Most of these methods only consider the univariate case, and therefore, neglect more complicated arrangement involving three or more variables. Moreover, the data used for an outlying measure are often required in the detection step. In this paper, we present a new method for faulty node detection based on Copula theory. This method aims at addressing the common drawbacks of the existing outlier detection methods cited above. Our method offers important advantages, as it does not require any knowledge of the distribution of the captured variables and it enables us to model joint probability distributions over multiple (more than two) random variables with a parametric family of copula. The resulting probability densities can be used to quantify the probability of data occurrence, and if this probability exceeds a certain threshold the data is marked as an outlying measure. Finally, depending on the number of involved nodes, we can distinguish between an occurring event or a separated faulty node.

Definition and properties of copulas

Originating from Sklar's theorem [31], copulas are a way of describing complex dependency structures and to relate them to marginal distributions within a single function. The definition of a copula for a bivariate distribution is as

follows:

A d-dimensional copula $C : [0,1]^d \to [0,1]$ is a distribution function with uniform marginals. A copula can be considered as a probability function given by the Equation (1) [31]:

$$C(u_1, \ldots, u_d) = P(U_1 \leq u_1, \ldots, U_d \leq u_d) \tag{1}$$

where $U_1, \ldots, U_d$ are uniform distributions on the interval $[0,1]$.

A copula function C has the following properties [31]:

1. $C(u_1, \ldots, u_d)$ is increasing in each component u_i
2. $C(1, \ldots, 1, u_i, 1, \ldots, 1) = u_i$ for every $i \in \{1, \ldots, d\}, u_i \in [0,1]$
3. $C(u_1, \ldots, u_d) = 0$ if $u_i = 0$ for every $i \leq d$

From Equation (1), we can deduce that for every multivariate distribution function, a copula can separate the marginal distributions from their dependence structure, as is expressed by Sklar's theorem [31]. Let F be a d-dimensional joint distribution function with marginals $F_1, ..., F_d$. Then there exists a copula $C : [0,1]^d \to [0,1]$ such that,

$$F(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d)), \ \forall x_1, ..., x_d \tag{2}$$

Fréchet-Hoeffding bounds

Any copula function $C(u_1, \ldots, u_d)$ must lie within certain bounds, which are known as the Fréchet-Hoeffding bounds [31]:

$$\max\left(\sum_{i=1}^{d} u_i + 1 - d, 0\right) \leq C(u_1, \ldots, u_d) \leq \min\left(u_1, \ldots, u_d\right) \tag{3}$$

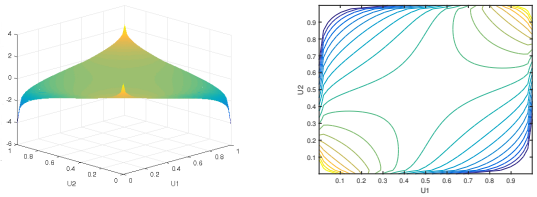

Figure 1: Surface and contour plots of the Gaussian copula with parameter ρ=0.3 (from right to left).

The upper bound is also called the co-monotonicity copula, and the lower one is a generalized form of the counter-monotonicity copula.

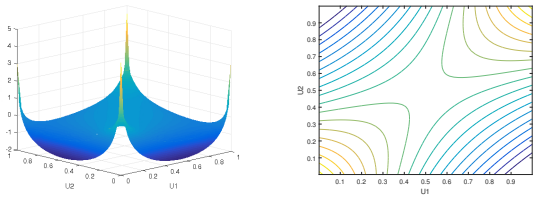

Figure 2: Surface and contour plots of the Student copula with parameter ρ=0.3 ν=3 (from right to left).

Usual Copula families

In this section, we present different classes of copulas, in particular fundamental, elliptical and Archimedean copulas. Some properties and visualizations will be provided for each family. We will use the term *copula family* to designate the copula function with one or more real parameters, and the term *copula class* to designate a collection of copula families that have the same properties.

The independence copula

One of the simplest form of a copula is the independence copula used whenever the random variables are independent. It is also called the product copula because of its form [32]:

$$C(u_1, \ldots, u_d) = \prod_{i=1}^{d} u_i \tag{4}$$

Elliptical Copulas

Elliptical copulas are based on existing multivariate elliptical distributions. In the following, we present two particular cases: the Gaussian and the Student copula.

Gaussian copula

This copula is obtained from the multivariate normal distribution, and it is expressed as follows [32]:

$$C_{\sigma}^{Ga}\left(u_1,\ldots,u_d\right)=\Phi_{\Sigma}(\Phi^{-1}\left(u_1\right),\ldots,\Phi^{-1}\left(u_d\right)),\quad \left(u_1,\ldots,u_d\right)\epsilon[0,1]^d \tag{5}$$

where Φ_{Σ} represents the Cumulative Distribution Function (CDF) of the multivariate normal distribution with a correlation matrix Σ, where Φ^{-1} is the quantile function of the normal distribution. Figure 1 illustrates the surface and contour plots of a Gaussian copula with a dependence parameter $\rho = 0.3$.

Note that the Gaussian copula is not only obtained from the multivariate normal distribution. It can also be obtained from any multivariate distribution.

Student copula

This copula is also called the desert island copula [33], because it exhibits the best fit among different families. It is based on the multivariate Student t-distribution.

$$C_{\upsilon,\sigma}^{t}(u_1,\ldots,u_d)=t_{\upsilon,\sigma}(t_{\upsilon}^{-1}(u_1),...,t_{\upsilon}^{-1}(u_d)) \tag{6}$$

where t_{υ} is the CDF of the univariate Student t-distribution with υ degrees of freedom and $t_{\upsilon,\sigma}$ is the CDF of the multivariate Student t-distribution with a correlation matrix Σ and υ degrees of freedom. Figure 2 illustrates the surface and contour plots of a Student copula with a dependence parameter $\rho = 0.3$ and a degree of freedom v=3.

Archimedean Copula

Archimedean copulas have a more regular form compared to the Gaussian and Student ones. They rely on a non-increasing function $\varphi[0,\infty) \rightarrow [0,1]$, the so-called generator of the copula, which satisfies the following conditions [34]:

$$\varphi(1)=0,\quad \varphi(0)=\infty \tag{7}$$

and which is strictly decreasing on $[0, \infty)$. A d-dimensional copula C is called Archimedean, if it permits the representation [34]

$$C(u_1, ..., u_d) = \varphi(\varphi^{-1}(u_1) + ... + \varphi^{-1}(u_d)), u_i \in [0, 1] \quad i = 1 \dots d \qquad (8)$$

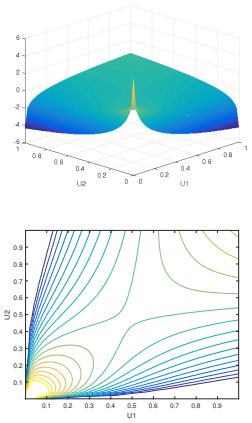

Figure 3: Surface and contour plots of the Clayton copula with $\theta = 0.7$.

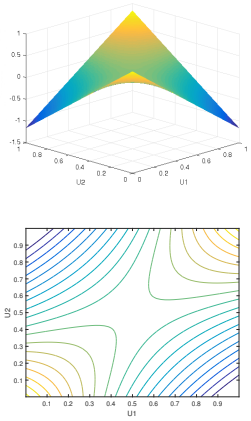

Figure 4: Surface and contour plots of the Frank copula with $\theta = 2$.

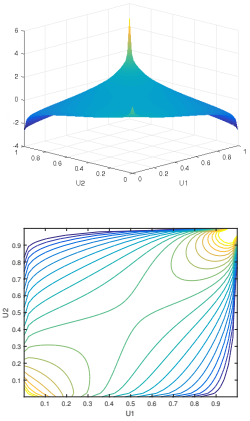

Figure 5: Surface and contour plots of the Gumbel copula with $\theta = 1.3$.

Depending on the used generator, we can find different types of Archimedean copulas. This generator can have one or more parameters. Using different values of the parameters, different copulas can be obtained. Nelsen [34] describes a total number of 22 different Archimedean copula families generated by a single generator. In the following, we will introduce the three copulas most frequently used in real applications. The Clayton copula is often used in the field of finance to study correlated risks. It exposes the dependence in the lower tail but cannot express a negative dependence. The Gumbel copula is able to model asymmetric dependencies of data, e.g., weaker lower and stronger upper dependence like in the Clayton copula, but it cannot represent a negative dependency. As opposed to the Clayton and Gumbel copulas, the Frank copula can be used to represent both positive and negative dependencies and it can describe symmetrical dependencies . These dependencies are visually described in Figures 3, 4 and 5 with different dependence parameters θ. The generators of these copulas are given in Table 1 [34].

Table 1: Archimedean copulas and their generators.

Copula	$\varphi(u)$	$\varphi^{-1}(u)$	θ
Clayton	$(1+u)^{\frac{-1}{\theta}}$	$u^{-\theta}-1$	$(0,\infty)$
Gumbel	$e^{-u^{\frac{1}{\theta}}}$	$(-\ln u)^{\theta}$	$[1,\infty)$
Frank	$-\frac{1}{\theta}\log(1-(1-e^{-\theta})e^{-u})$	$-\ln\frac{e^{-\theta u}-1}{e^{-\theta}-1}$	$(0,\infty)$

The proposed approach

To detect abnormal events in an unknown environment, we first have to start from an initial model of the environment on-line. Copulas represent a straightforward concept for modelling different dependence structures. In recent years, Copula theory has received a high interest from the research community. This method has been increasingly used in statistical modelling offering a great flexibility as it can separate marginal distributions from the distribution of the dependency of a dataset. It represents an important and accurate tool that deals with uncertainty issues and multidimensionality in practical situations.

The purpose of this work is to define a Copula-based approach for outlier detection, which could be used for the detection of faulty nodes in wireless sensor networks. We consider a set of n time-synchronized sensor nodes $S = \{s_1, s_2, ..., s_n\}$ organized into a cluster topology as illustrated by Figure 6. Each cluster is composed of a cluster head CH and a group of nodes $s_i \subseteq S$ for $i = 1, \ldots, n$. In this work we assume that the clusters of the network are already predefined.

Our approach can be described by three main stages, as illustrated by the basic functional blocks of Figure 7:

- Estimator stage: the base station uses the sensor data captured by the nodes of the network, where we assume that there is no outlier in the obtained samples, in order to produce a unique classifier. Such a classifier, represented by an anomaly detection function, is determined for each sensor node and sent to it.

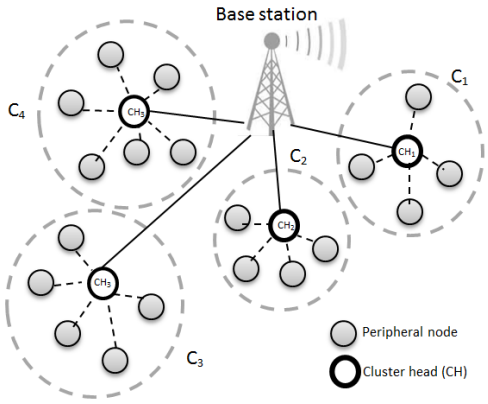

Figure 6: WSN cluster topology.

- Detection stage: each sensor node uses the received anomaly detection function to classify its new samples as normal or as outliers. This step is performed in a totally distributed manner and without any inter-node communication.

- Outlier classification stage: according to the contiguity and frequency of detected outliers, the cluster head can classify them into events or into local outliers.

These main steps will be described in more detail in the following subsections.

Expressing dependence through copula

The estimation of the copula model between every pair of sensor measurements is computationally expensive if it is done by the sensor nodes. The most reliable and low-cost option is to estimate the copula at the base station. This choice allows the use of more powerful devices, to analyze the dataset and to build the model.

However, these estimated models are valid only when the relationships remain stable through time. This may not always be true since sensor nodes are receiving new measurements continually, and some additional information can

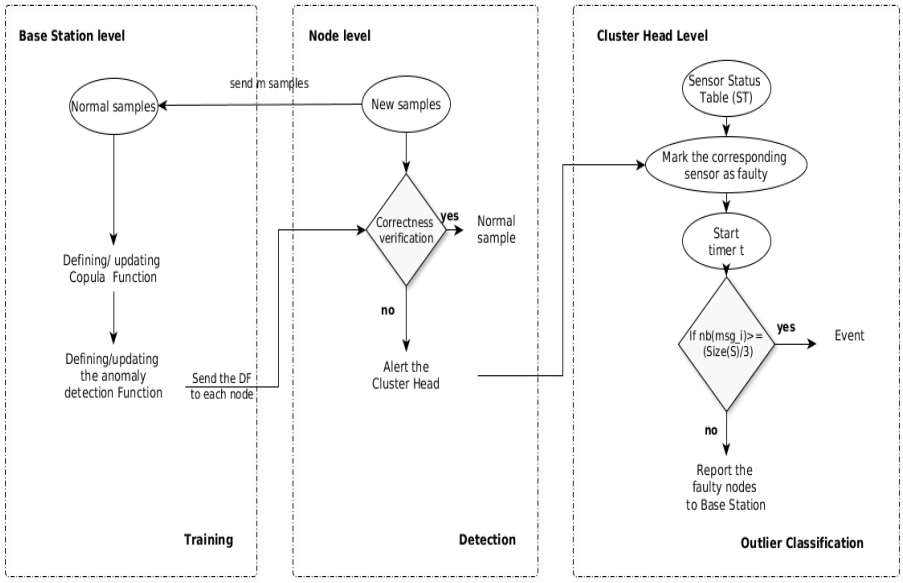

Figure 7: Details of the algorithm used in the approach.

be extracted from newly collected measurements. The design of an updating criterion is not considered in this paper.

Our objective is to propose the use of a copula function for a more general and accurate modelling of data dependencies. As opposed to the Gaussian-based model, the correlation modelling will be used without making any assumption on the data distribution. Consequently, the statistical properties of the reading dependencies of each sensor are represented more accurately. In the following, we consider the case of one sensor node within the sensor network. During each time interval Δ_k, the sensor node produces a vector of d measurements $X^k = (x_1^k, \ldots, x_d^k)$, where each X^k represents the vector of physical measurements of the node at time instance k. The variable j represents one of the d physical values monitored by the sensor node, for instance, vibration, temperature, pressure, etc.

We assume that the first m measurements of the sensor node are outlier free, and therefore, they will be used by the base station as a training

dataset to define the copula function. These m independent samples $X^1 = (x_1^1, \ldots, x_d^1), \ldots, X^m = (x_1^m, ..., x_d^m)$ are contained in the d-dimensional vector X.

Unlike SVM, which assumes a defined distribution for the data, we do not have to introduce any distributional assumptions for the marginals. Instead, the copula fitting is carried out using a rank transformation of the vector X. Figure 8 shows an example of this transformation, where the inference is based on the so-called pseudo-samples $u^1 = (u_1^1, \ldots, u_d^1), \ldots, u^m = (u_1^m, \ldots, u_d^m)$ from the pseudo-vector U, where [35]

$$u^k = (u_1^k, \ldots, u_d^k) = (\frac{r_1^k}{(m+1)}, \ldots, \frac{r_d^k}{(m+1)}) \tag{9}$$

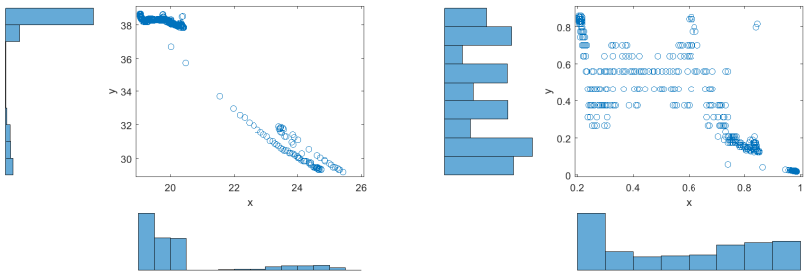

Figure 8: Normal scaled and ranked scale data for the first 600 humidity-temperature measurement-mote 1 (from right to left).

r_j^k representing the rank of x_j^k amongst $(x_j^1, ..., x_j^m)$. The rank is defined after sorting each group of observations in ascending order where the smallest value has rank 1 and the largest one rank m (the highest rank). This transformation highlights the dependence structure, where all different random variables are transformed to a common domain. Therefore, they are considered as samples from the underlying copula C. This rank transformation also introduces dependence and $(u^1, \ldots, u^n)$ are no longer independent samples.

Let $U_1, U_2, ..., U_m$ be their marginal distribution functions such that $U_j = F_j(u_j)$. According to Sklar's theorem a unique d dimensional copula C exists,

such that [31]:

$$F\left(u_1, u_2..., u_d\right) = C\left(U_1, ..., U_d\right) \tag{10}$$

In the literature, there exists a large family of copula functions. For our approach, we have tested some copulas among a dozen of suitable ones. An incorrect choice of the representative copula can lead to biased results; the selection of the copula is therefore a crucial step. Different methods have been proposed to deal with the uncertainty in real dependence structures underlying the studied phenomena. One way is to use the Akaike Criteria (AIC), where a set of copula types are fitted using the maximum likelihood estimation, and the AIC criteria are computed for each resulting copula. The copula with the minimum AIC value is selected as follows [36]. For u_j^k, with $k \in \{1, ..., m\}$ and $j \in \{1, 2, ..., d\}$, the AIC of a bivariate copula family C with parameter θ is defined as:

$$AIC \ = \ -2 \sum_{i=1}^{n} ln(C(u_1, u_2, ..., u_d|\theta)) \tag{11}$$

The process of fitting a copula can then be given as follows:

1. Start with the vector $D_{tr}^i = \left\{X_1^i, X_2^i, \ldots, X_m^i\right\}$ of the first m measurements of a sensor node i, where $i \in \{1, ..., n\}$.
2. Convert D_{tr}^i into pseudo-variables u_j^k such that $u_j^k \in [0, 1]$, $\forall i \in \{1, ..., n\}$, $\forall j \in \{1, ..., d\}$, $\forall k \in \{1, ..., m\}$.
3. Estimate the parameters associated with the type of copulas: Gaussian, Student, Gumbel or Frank.
4. Calculate the AIC of the resulting copula using Equation (11).
5. Define the optimal copula function according to AIC.

Defining the one-class Copula classifier

As mentioned above, the aim of this work is to introduce the use of copula functions into a semi-supervised outlier detection process. According to Sklar's theorem [31], we can employ a copula function within a probabilistic classifier, such as a Bayesian classifier. In this section, we will describe a probabilistic

model based on a d-empirical distribution function and a d-variate dimensional copula function. Bayes'theorem gives the following [37]:

$$P(\Omega = k \mid x_1, \ldots, x_d) = \frac{P(x_1, \ldots, x_d \mid \Omega = k)P(\Omega = k)}{P(x_1, \ldots, x_d)} \tag{12}$$

where Ω is the set of classes, $P(\Omega = k \mid x_1, ..., x_d)$ is the posterior probability of the class k , $P(x_1, ..., x_d \mid \Omega = k)$ is the likelihood function of the class k, $P(\Omega = k)$ is the prior probability and $P(x_1, ..., x_d)$ is the data probability.

For continuous attributes, a density function can be used to model the dependence structure in the likelihood function. In this case, Equation (12) can be written as follows [37]:

$$P(\Omega = k|x_1, \ldots, x_d) = \frac{f(x_1, \ldots, x_d|\Omega = k)P(\Omega = k)}{f(x_1, \ldots, x_d)} \tag{13}$$

where $f(x_1, \ldots, x_d|\Omega = k)$ expresses the likelihood of the vector $(x_1, \ldots, x_d)$ given the class k. According to Equation (2), any joint distribution function F with continuous marginals $F_1, F_2, \ldots, F_d$ can be expressed through a copula function C, which is a function of the marginal distributions $F_1, F_2, \ldots, F_d$. Using Sklar's theorem enables us to highlight the existing relationship between the d-dimensional joint Probability Density Function (PDF) c and the marginal PDFs $f_1, f_2, \ldots, f_d$ as shown by Equation (14). Then the likelihood function in (13) can be expressed with a copula. In [38], a Gaussian copula has been used to construct the Bayesian classifier, but in order to deal with the different existing dependencies in environmental data, we will not restrict our work to the Gaussian copula.

$$f(x_1, ..., x_d|\Omega = k) = c(F_1(x_1), \ldots, F_d(x_d)|\Omega = k) \prod_{i=1}^{d} f_i(x_i|\Omega = k) \tag{14}$$

By replacing Equation (14) in Equation (13), we obtain:

$$P(\Omega = k|x_1, \ldots, x_d) = \frac{c(F1(x_1), \ldots, F_d(x_d)|\Omega = k) \prod_{i=1}^{d} f_i(x_i|\Omega = k)P(\Omega = k)}{f(x_1, \ldots, x_d)} \tag{15}$$

From Equation (15) we can see that a copula allows to model flexibility by separating the marginal distributions from their dependence structure. Bayes'

rule enables us to model posterior as a product of likelihood and prior by ignoring the normalizing constant in Equation (15) [39], and we can construct a Maximum A posteriori Probability density classifier based on copulas as follows:

$$arg \max_{k \in \Omega} c(F_1(x_1), \ldots, F_d(x_d)|\Omega = k) \prod_{i=1}^{d} f_i(x_i|\Omega = k) P(\Omega = k) \qquad (16)$$

where $c(F_1(x_1), \ldots, F_d(x_d)|\Omega = k) \prod_{i=1}^{d} f_i(x_i|\Omega = k)$ is the likelihood function for each class k modeled by the copula density function. We can use the resulting classifier in the one class classifier, where Equation (16) becomes the discriminant function of the initial class N represented by the following equation:

$$g(x_1, \ldots, x_n \mid N) = c(F_1(x_1), \ldots, F_d(x_d)|N) \prod_{i=1}^{d} f_i(x_i|N) P(N) \qquad (17)$$

We can assume that an outlier can be characterized as an observation that lies in a low-density region compared to other values in the dataset [40]. Before highlighting an outlier, it is necessary to describe what is considered as a normal observation. Every new sample is evaluated with the discriminant function given by Equation (17) to test its correctness on the basis of the calculated value $g(u)$. The sample is labelled as normal or as an outlier using a threshold, which must be defined in advance by the user. The detection process is described by Algorithm 1.

Algorithm 1 Copula based outlier detection.

1: **Input:** $x_1, \ldots, x_n$, the n^{th} vector of measurements of the sensor node.
2: **if** $(g(x_1, \ldots, x_n \mid N) < threshold)$ **then**
3: **return** outlying measurement
4: **else**
5: **return** normal measurement
6: **end if**

In any anomaly detection algorithm, one of the main difficulties lies in the selection of a correct threshold which maximizes the detection rate and at the

same time minimizes the false positives (or false alarms). In our case, a probabilistic control limit is determined to define the normal region by using F1-score on cross-validation ground truth dataset. This metric is calculated using Equation(18), where r is recall and p is precision. We use a rule of thumb in order to optimize the F1-score. This latter finds the right balance between the number of anomalies detected by the algorithm and the misclassified normal instances. In this case the defined threshold is called "discrimination threshold".

$$F1 = 2\frac{1}{\frac{1}{r} + \frac{1}{p}} \tag{18}$$

In our proposed approach, the detection method is done by each sensor node independently from the rest of the network. A cluster head (CH) is alerted as soon as an outlying measurement is detected. The copula represents the multivariate cumulative distribution function of the data. Its density illustrates the strength of dependence between the margins. Abnormal measures can be identified by combining the resulting copula with Bayes' theory which links the probability density function of the given class of data to the posterior probability of the class given the data.

The most straightforward method to obtain a one-class classifier is to estimate the density of the training data [41] and to set a threshold on this density, assuming that a good probability model is used (one for which the bias is small) and that the threshold is optimized. The big advantage of a copula is that it can characterize a multivariate dataset without making any assumption on the distribution of its margins and thus the bias is kept to the minimum.

Outlier classification

Outliers can be engendered in different ways. According to the number and contiguity of the involved nodes, we can classify them into local and global outliers. Local outliers are abnormal measurements involving an isolated sensor node due to a dysfunction that can have various origins. The global outlier or event is an anomaly which can reveal an occurring phenomenon such as an

earthquake or a fire. An event is detected by a certain number of network nodes that are spatially and temporally correlated. We consider that each cluster head CH maintains a Sensor State Table ST which initially contains the id of the sensors in the cluster with outlier-free data. Figure 9 provides an illustrative example of the copula cluster based outlier detection in a sensor network with a two-level hierarchical topology.

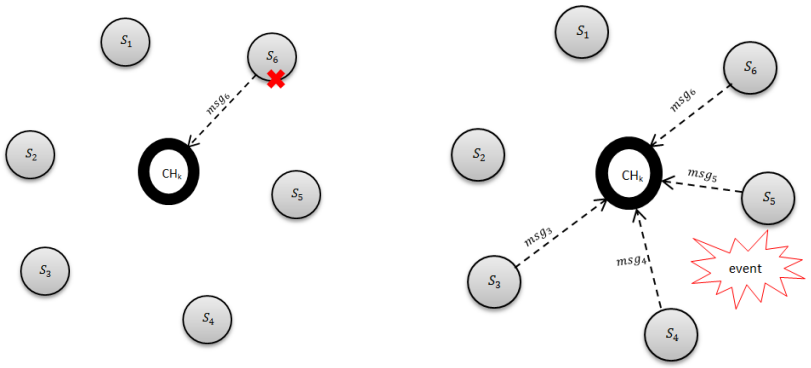

Figure 9: Local outlier and Global outlier.

It is possible to differentiate between global and local anomalies at the cluster head level. When the first alert is detected by the sensor node s_i, it will be sent to its cluster head, which will trigger its timer, and set to faulty the state of that sensor. At the end of the timer, if the number of faulty sensor nodes is greater than half of the size of the cluster, which means that an event has occurred, then the base station will be informed. Otherwise, the list of faulty sensor nodes will be sent to the base station. The steps of detecting outliers based on copulas and clusters are summarized in Algorithm 2.

Experiments and results

To validate the efficiency of our approach, we have tested it on two real-life WSN datasets: we have used the public Intel Berkeley Research Lab dataset IBRL [42]. This dataset contains the original readings of 54 Crossbow Mica2Dot

Algorithm 2 Copula cluster based outlier detection classification.

Input:
size(S) : the size of the node's cluster set S.
t: timer.
msg_i: alert message from the sensor node s_i.
begin *for each message* msg_i
Receive (msg_i);
Mark the sensor node as faulty;
Start (t);
while not null(t) **do**
    if Receive(msg_k) **then**
        mark the sensor node s_k as faulty ;
    end if
end while
if number(faulty)≥(size(S)/3) **then**
    report the event to the base station ;
else report the faulty nodes to the base station ;
end if

sensors, with weatherboard implemented and deployed in the Intel Berkeley Research Lab between February 28^{th} and April 5^{th}, 2004. The collected data include the measurements of temperature, humidity, light and voltage along with the timestamps and the topology information. The second one is the publicly available real-world SensorScope Grand Saint Bernard [43]. It contains the readings of a multi-hop network of 19 weather stations, deployed from September 13^{th}, 2007, to October 26^{th}, 2007, on a 900-meter long line located between Switzerland and Italy at 2400 meter height. The collected weather data include the measurements of temperature, solar radiation, humidity, moisture, etc. Along with the timestamps and the topology information. This WSN had been deployed to prevent avalanches.

For each of the above datasets, we apply our approach and compare its result to other outlier detection approaches. The training and testing subsets are sampled by Monte Carlo cross-validation (five runs), with 70% of the dataset used for training and the other 30% used for testing. We report here the plots of ROC curves and averages of AUC and prediction time over the runs. Our experiments contain more than 80 runs and 240 plots, that we are going to describe in the following figures. In a first step we will give the hole figures produced during two typical runs in each of the datasets cited above (Intel Berkeley and Saint Bernard), and this for two cases: the outlier free training dataset and the contaminated training dataset. In a second step, we will summarize the produced results in our experiments through charts and tables. The experiments are performed on a Windows 7 platform and powered by an Intel(R)Core(TM) i5 CPU (2.27GHz) and 4GB RAM. We used the RStudio software of the R version 3.2.2 and Matlab R2016b.

The performance evaluation of our approach is based on the calculation of TPr/FPr as summarized in Table 2.

The verification outcomes are samples labeled as positive P or negative N. The verification can have four possible outcomes: if the predicted label is P and the actual value is also P, then it is defined as a True Positive (TP). However, if the actual value is N, then it is defined as a False Positive (FP). In the same

Table 2: *Confusion matrix.*

	Object from target class	**Object from outlier class**
Classified as target object	True positive (TP)	False positive (FP)
Classified as outlier object	False negative (FN)	True negative (TN)

manner, if a normal sample is correctly identified as negative, it is called a True Negative (TN), and if it is wrongly identified as positive then it will be called a False Negative (FN). These outcomes are used to calculate different metrics to compare the performance of our approach to those present in the literature. To point out the accuracy of our proposed method, we have used the receiver operating characteristic (ROC) curve, true positive rate versus false positive rate and the two metrics being computed using Equation (19) and (20) [44]. The comparison is then based on the area under the curve (AUC) which is estimated using the trapezoidal rule, and the average perdition time of each of the implemented methods.

$$TPr = \frac{\sum TP}{\sum (TP + FP)} \tag{19}$$

$$FPr = \frac{\sum FP}{\sum (FP + FN)} \tag{20}$$

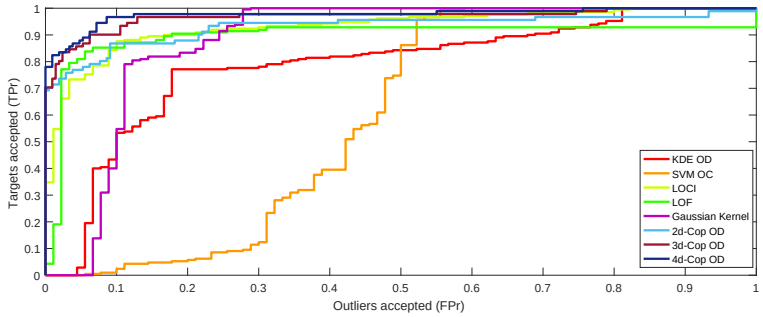

Figure 10: ROC Performance curve for the Intel Berkeley dataset with 30% of outliers in testing dataset.

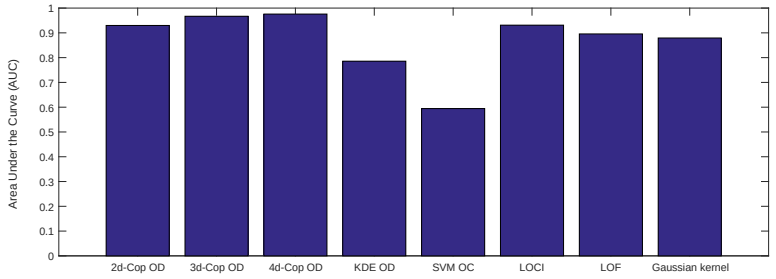

Figure 11: AUC for the Intel Berkeley dataset with 30% of outliers in testing dataset.

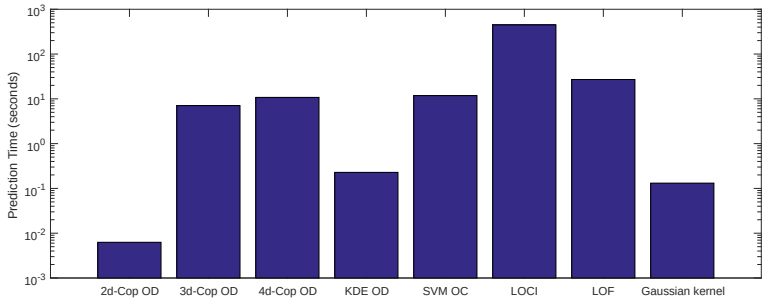

Figure 12: Prediction time for the Intel Berkeley dataset with 30% of outliers in testing dataset.

In order to assess the quality of our proposed method, we have investigated its performance against the 5 known outlier detection methods KDE, SVM with a linear Kernel, LOF, LOCI and the Gaussian Kernel based outlier detection. None of the cited methods can detect outliers in higher than 2 dimensions. Each experiment was performed using the Monte Carlo cross-validation with 70% of the dataset used for training and 30% of the dataset used for testing. Figure 10 plots the ROC Curve for the first 1000 measures captured by the mote 1 of the Intel Berkeley dataset. The training dataset is kept clean while we have added 30% of outliers into the testing dataset. Our method shows a 92.98%, 96.69%, 99.58% of average AUC for, respectively, dimensions 2, 3 and 4, and, as pointed out in Figure 11, outperforming each of KDE, SVM, LOF, LOCI and the Gaussian Kernel method. The SVM linear one class classifier is scoring the least performance with only 59.43%. LOCI's performance is a little less than our method's performance with 93.10%, but the prediction time is 447.43 seconds which represents more than 40 times the prediction time of our method in the 4-d copula. This last one is the worst time recorded for the copula outlier detection with 10.77 seconds of prediction time.

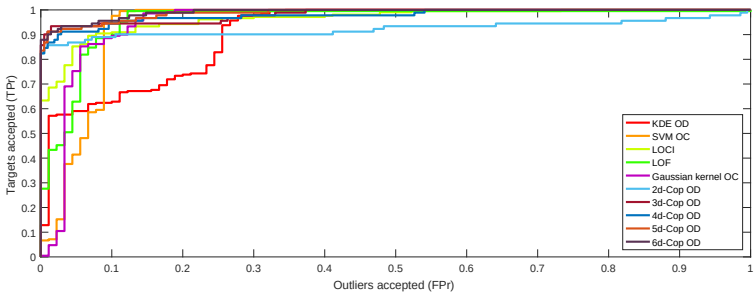

Figure 13: ROC plot for the Saint Bernard dataset with 30% of outliers in testing dataset.

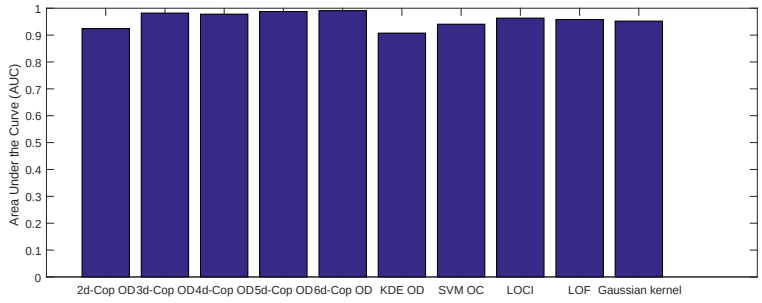

Figure 14: AUC for the Saint Bernard dataset with 30% of outliers in testing dataset.

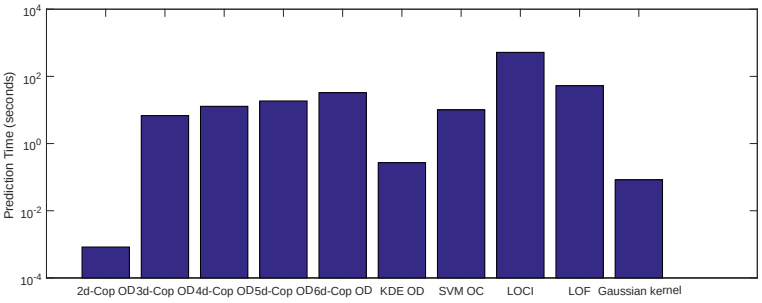

Figure 15: Prediction time for the Saint Bernard dataset with 30% of outliers in testing dataset.

To enlarge the number of dimensions that could be handled by our copula-based outlier classifier, we have tested it on the Saint Bernard dataset with the data captured by the weather station 17. We have tested our classifier for up to 6 dimensions, and the performance results are plotted in Figure 13. As shown in Figure 14, our method gives relatively good results with 92.02% of AUC for the 2 d-Cop OD and 98.89%, 98.60%, 99.0% and 99.30% for the 3 to 6 d-Cop OD, respectively. The other methods perform well with 90.35% for KDE, 93.51% for SVM; LOF and LOCI show similar results with 95.19% and 95.13%, respectively. LOF shows a 98.38% of AUC which is 0.92% less than the scored performance of our method but its prediction time is 20 seconds more than the

6d-Cop OD which is the highest prediction time recorded for our d-Cop OD.

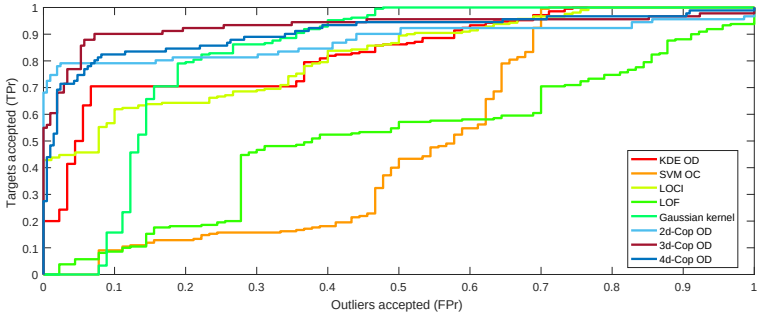

Figure 16: ROC plot for the Berkeley Intel Lab dataset with 30% of outliers in both the training and testing datasets.

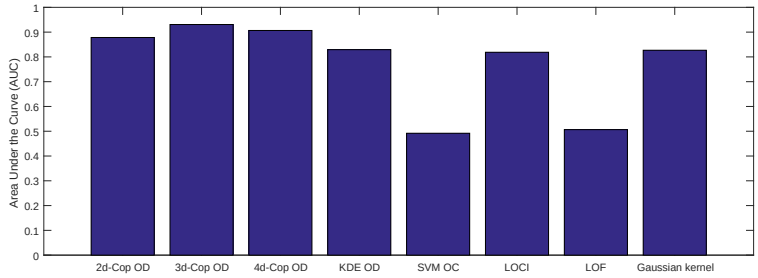

Figure 17: AUC for the Berkeley Intel Lab dataset with 30% of outliers in both the training and the testing datasets.

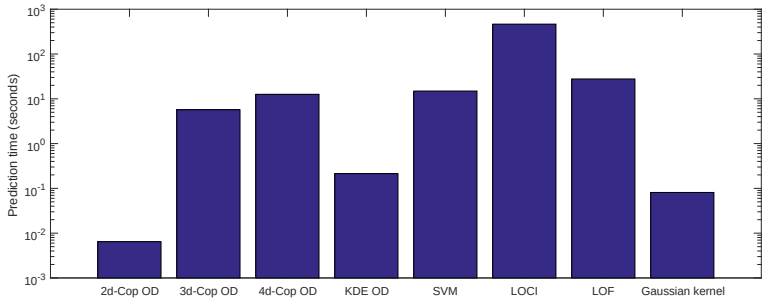

Figure 18: Prediction time for the Berkeley Intel Lab dataset with 30% of outliers in both the training and the testing datasets.

The captured data used for training are not always outlier free, which could seriously corrupt the performances of the learned data model, and lead to a low detection rate. To investigate this aspect statistically, we have increased the rate of outliers present in the training datasets to 30%. This experience aims at testing the robustness of our proposed method against other existing methods. The Figures 16 to 21 present the performances of our proposed method along with existing ones when the training dataset is contaminated by outliers.

Figure 16 plots the performances of our proposed method in the case where 30% of the training dataset is contaminated by outliers. The used dataset is the Intel Berkeley dataset, and the calculated AUC of the resulting ROC curve is plotted in Figure 17. Our method's performances range from 87.83% to 93.07% where 3d-Cop OD scores the highest rate of AUC. The closest performance is the Gaussian Kernel OD with 82.69% followed by KDE OD with 82.92%. The proposed nd-Cop OD prediction times vary by 64 milliseconds for the 2d-Cop OD, 5.6973 and 12.6077 seconds for the 3d and 4d Cop OD, which represents a good performance since the prediction time of the most efficient competitive outlier detection method KDE OD is 2133 milliseconds with an 82.92% AUC, which is 4.91% less than the 2d-Cop OD and 33 times its prediction time.

For the second dataset (Saint Bernard) the d-Cop OD scores a better AUC than for the first dataset with a maximum of 97.13% AUC and 24.91 seconds pre-

diction time for the 6d-Cop OD, outperforming the rest of the existing methods. The maximum AUC rate is obtained by LOCI with 89.90%, but its corresponding prediction time is really high with 573.29 seconds, which represents 23 times the prediction time of the 6d-Cop OD.

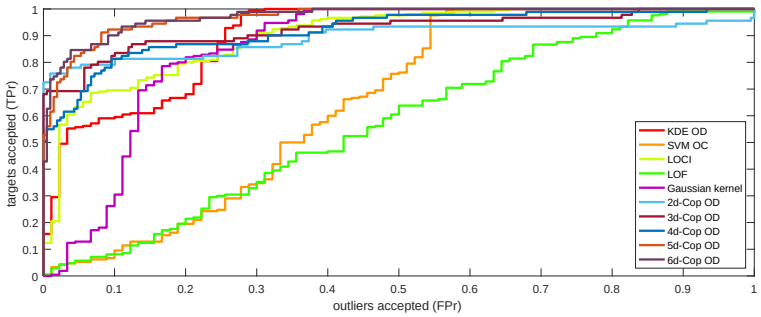

Figure 19: ROC plot for the Saint Bernard dataset with 30% of outliers in both the training and the testing datasets.

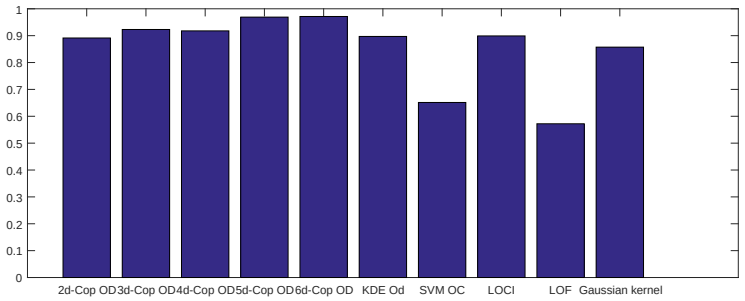

Figure 20: AUC plot for the Saint Bernard dataset with 30% of outliers in both the training and the testing datasets.

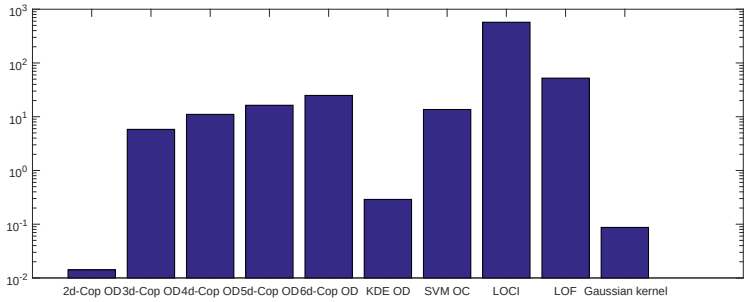

Figure 21: Prediction time for the Saint Bernard dataset with 30% of outliers in both the training and the testing datasets.

In the following tables and figures, we will summarize the rest of our results in form of charts and tables. To appreciate the impact of outliers on the performance of outlier detection methods, we introduce the following metrics, where P_{Avg}, t_{Avg} represent, respectively, the average of the recorded performances AUC and the prediction time with different rates of outliers in the training/testing datasets; P_{Diff}, T_{Diff} represent the difference between the recorded AUC and prediction time, respectively, in lowest and highest outlier rates. This metric is positive when the performance/prediction time is increasing and negative when the performance/prediction time is decreasing .

$$P_{Avg} = \frac{\sum AUCs}{N} \tag{21}$$

$$P_{Diff} = Max(AUCs) - Min(AUCs) \tag{22}$$

$$t_{Avg} = \frac{\sum t_{pred}}{N} \tag{23}$$

$$t_{Diff} = Max(t_{pred}) - Min(t_{pred}) \tag{24}$$

where N is the number of test cases.

For clarity of presentation and to have a better vision of our obtained results, we have plotted the results of Tables 3 to 6 in Figures 22 to 25, respectively.

Table 3 shows the average of five run cross-validation with different rates of outliers in the testing dataset and 0% of outliers in the training dataset. As plotted in Figure 22, we can notice that the 4d-Cop OD exposes the highest AUC performance with 97.83% in the case where 40% of the testing dataset contains outliers; and a P_{Diff}=-2.05% which is the lowest rate recorded in the d-Cop OD. Other methods than the KDE show an increase in the performance with P_{Diff}= +9.84%; We assume that this is due to the fact that this kind of outlier detection method performs better in the case where the rate of healthy and outlying instances are equally balanced in the testing dataset. For the prediction time, we can observe that the increasing rate of outliers influences the prediction time in all of the tested outlier detection methods.

As reported in Table 3, the most influenced methods are LOCI and LOF with +270.82 and +15.15, respectively. The reported prediction time for the 4d-Cop OD is +7.27 seconds, and the highest prediction time is recorded by LOCI, the lowest by 2d-Cop OD.

Table 3: Performance measure averages for the Intel Berkeley Lab dataset with different rates of outliers in testing dataset, five run Monte Carlo cross validation.

	AUC		Prediction time	
	P_{Avg}	P_{Diff}	t_{Avg}	t_{Diff}
2d-Cop OD	92.65	-3.45	0.0072	+0.0047
3d-Cop OD	96.24	-2.69	5.44	+3.79
4d-Cop OD	97.83	-2.05	9.67	+7.27
KDE OD	80.74	+9.84	0.22	+0.18
SVM OC	68.06	-4.08	8.41	+6.12
LOCI	95.89	+1.27	368.16	+270.82
LOF	91.23	-0.73	29.89	+15.15
Gaussian Kernel	91.34	+2.1	0.09	+0.05

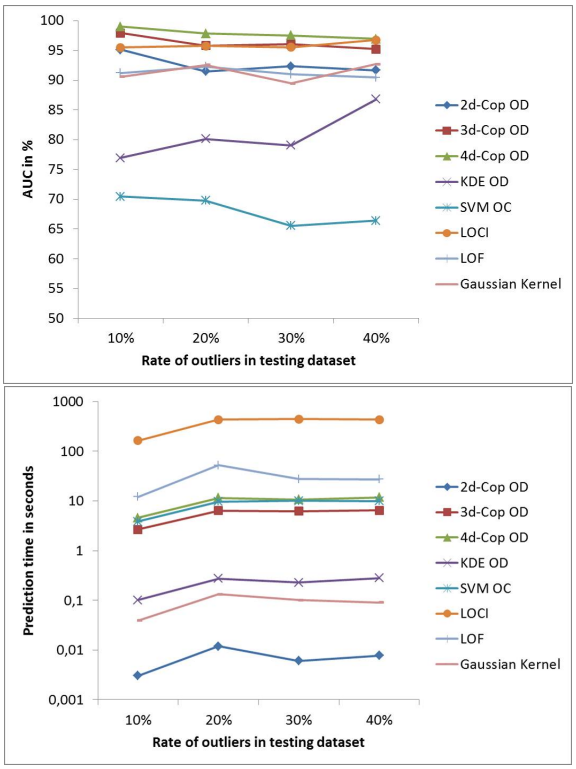

Figure 22: Average calculated AUC and prediction time for the Intel Berkeley dataset with different rates of outliers in testing dataset, five runs of Monte Carlo cross validation.

In order to investigate the influence of a contaminated training dataset with outliers, we have performed four tests on the Intel Berkeley dataset with different rates of outliers in the training dataset (10%, 20%, 30% and 40%) and a stable rate of outliers in the testing dataset (30%). The averages of five Monte Carlo cross-validations are reported in Table 4 and their corresponding plots are illustrated in Figure 23. The best P_{Avg} is performed by the 4d-Cop OD with 91.12% followed by 3d-Cop OD with 90.38%, 2d-Cop OD with 87.38%, LOCI with 86.09% and Gaussian Kernel with 85.23%. The presence of outliers in the training dataset has dropped their corresponding performances with -12.25% down to -13.67% for the nd-Cop OD. The highest drop of performance is recorded by LOCI with -15.2% and the lowest one by Gaussian Kernel with

-6.84%.

Table 4: Performance measure averages for the Intel Berkeley Lab dataset with different rates of outliers in training dataset, five runs of Monte Carlo cross validation.

	AUC		Prediction time	
	P_{Avg}	P_{Diff}	t_{Avg}	t_{Diff}
2d-Cop OD	87.38	-12.25	0.0072	-0.0006
3d-Cop OD	90.38	-13.32	6.64	+1.54
4d-Cop OD	91.12	-13.67	11.14	+1.04
KDE OD	84.47	-2.28	0.27	+0.03
SVM OC	52.82	-6.15	13.03	+2.51
LOCI	86.09	-15.2	467.2	+53.55
LOF	54.56	-14.14	34.55	+6.64
Gaussian Kernel	85.23	-6.84	0.1	-0.03

Despite a relatively limited drop of performance, this drawback is recovered by the low prediction time. As reported in Table 4, the nd-Cop OD scored a 72.41 milliseconds 6.64 and 11.14 seconds average prediction time for the 2d, 3d, and 4d Cop OD, respectively. The presence of outliers in the training dataset is of little influence on the recorded prediction, which has an increase of +1.54 seconds in the worst case for the 4d-Cop OD.

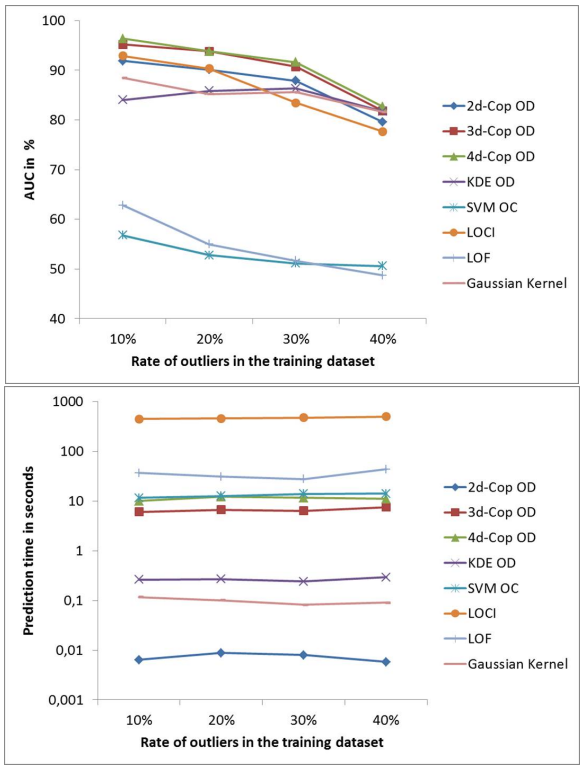

Figure 23: Average calculated AUC and prediction time for the Intel Berkeley dataset with different rates of outliers in training dataset (30% of outliers are used in the testing dataset), five runs of Monte Carlo cross validation.

In order to test our copula-based outlier detection in higher dimensions, we have used the Saint Bernard dataset. Table 5 reports the results of five run Monte Carlo cross-validation with different rates of outliers in the testing dataset. The 6d-Cop scores the highest AUC as plotted in Figure 24 with an average AUC of 97.75% and 28.13 seconds of average prediction time, outperforming the rest of the proposed methods. The closest performance is obtained by LOF with a $P_{Avg} = 96.1\%$ and $t_{Avg} = 53.11$ seconds, which represents almost 2 times the prediction time recorded for the 6d-Cop OD. This result reveals that increasing the rate of outliers in the testing dataset improves the performance of our proposed method, with a top of 2.05% for the 4d-Cop OD. In the rest of the cases the performance methods are decreasing with a top of 3.05% for KDE

OD.

Table 5: Performance measure averages for the Saint Bernard dataset with different rates of outliers in the testing dataset, five run Monte Carlo cross validation.

	AUC		Prediction time	
	P_{Avg}	P_{Diff}	t_{Avg}	t_{Diff}
2d-Cop OD	92.36	+0.93	0.0009	0
3d-Cop OD	97.61	+0.33	7.26	+0.05
4d-Cop OD	96.76	+1.38	12.89	+0.2
5d-Cop OD	97.26	+2.4	19.78	-0.68
6d-Cop OD	97.75	+0.74	28.13	+1.79
KDE OD	89.5	-3.05	0.31	+0.07
SVM OC	91.62	-2.3	9.8	-0.12
LOCI	93.23	-2.66	531.8	-9.62
LOF	96.1	-0.42	53.11	-1.42
Gaussian Kernel	93.27	-0.77	0.12	-0.01

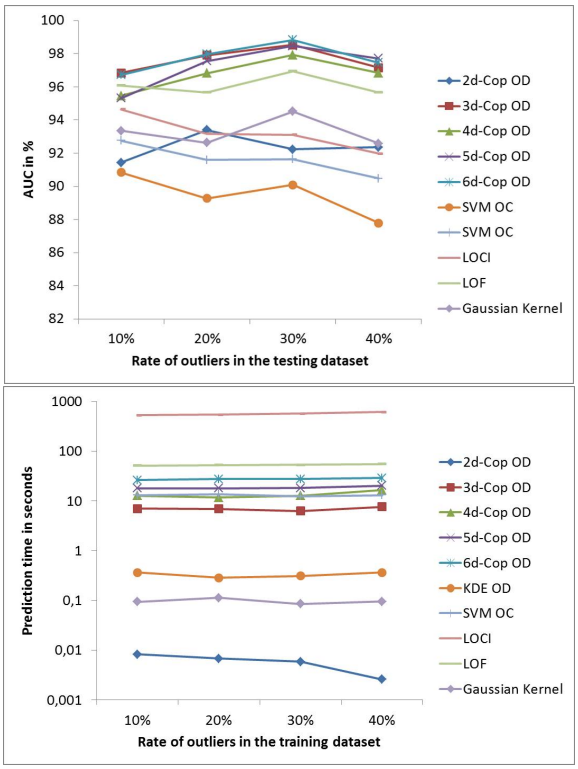

Figure 24: Average calculated AUC and Prediction time for the Saint Bernard dataset with different rates of outliers in the testing dataset, five run Monte Carlo cross validation.

Table 6 shows the results of the experiments performed on the Saint Bernard dataset when the training dataset is composed with a certain rate of outliers. Increasing this later gives us an overview of the impact of a contaminated training dataset on our method. As pointed out in Figure 25, the best performing methods are the 5d and 6d-Cop OD with 94.38% and 94.43% P_{Avg} for each of them, respectively, which represents a slight difference of 0.05%. The highest drop in performance is held by the d2-Cop OD with $P_{Diff} = -13.62\%$, followed by 4d-Cop OD with 12.41%. In contrast to the Intel Berkeley dataset, the presence of outliers in the training dataset has more influence on the performance of our proposed method. In summary, in all of the results, the least time-consuming approach is the 2d-Cop OD. The prediction time is rapidly increasing in higher

dimensions, compared to the 2d one, but it still manages to keep a much lower prediction time than LOCI. In all of the reported results, our method scores the highest performance in terms of AUC, and keeps a relatively good one compared to the rest of existing methods when the training dataset is contaminated.

Table 6: Performance measure averages for the Saint Bernard dataset with different rates of outliers in the training dataset, five run Monte Carlo cross validation

	AUC		Prediction time	
	P_{Avg}	P_{Diff}	t_{Avg}	t_{Diff}
2d-Cop OD	87.05	-13.62	0.0059	-0.0057
3d-Cop OD	93.25	-7.08	6.9525	+0.67
4d-Cop OD	90.73	-12.41	13.42	+3.74
5d-Cop OD	94.38	-6.87	18.5725	+2.21
6d-Cop OD	94.43	-7.77	27.5375	+2.41
KDE OD	90.16	-0.01	0.33	0
SVM OC	63.8	-10.43	13.045	-0.01
LOCI	87.29	-9.05	563.04	-87.82
LOF	58.8	-4.41	52.995	+4.03
Gaussian Kernel	86.27	-8.31	0.0925	0

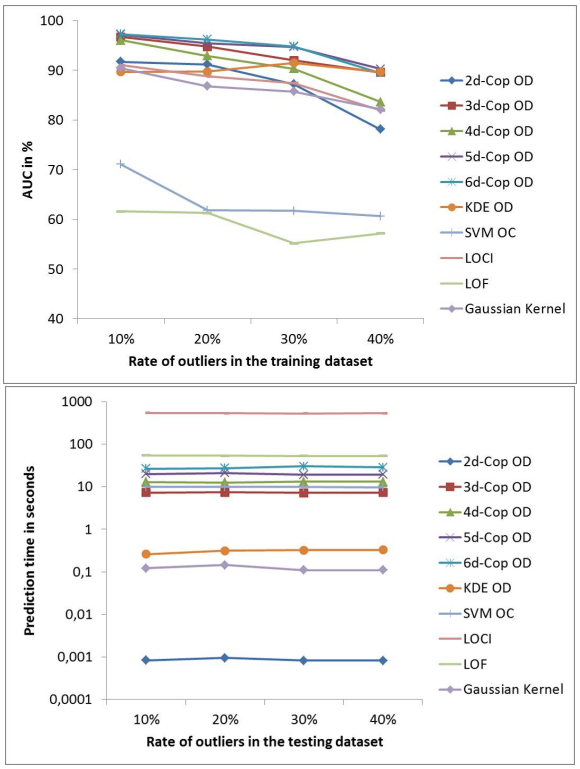

Figure 25: Average calculated AUC and prediction time for the Saint Bernard dataset with different rates of outliers in the training dataset (30% of outliers are used in the testing dataset), five run Monte Carlo cross validation.

Conclusion

Sensor data faults are very common in data collection using wireless sensor networks. Detecting such faults is sometimes a challenging task. This paper proposes a probabilistic outlier detection method based on copula theory. We have tested our approach with two real-life datasets, exposing promising results in terms of AUC performance and prediction time in comparison with some existing ones. The proposed method also shows a valuable resistance to outlier polluted training datasets. Copulas also enable us to detect these deviations in higher dimensions (3d or more), by modeling the unknown dependence structures with a known copula. No communication is needed throughout the whole

process of detection. Hence, it is energy efficient. Moreover, this approach can be very satisfactory for monitoring applications where the base station has to be alerted only when an abnormal event occurs. Most of the existing detection models consider that the marginal distributions of the sensors are also Gaussian. This assumption, however, does not hold in practice.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer networks 38 (4) (2002) 393–422.

[2] P. Rawat, K. D. Singh, H. Chaouchi, J. M. Bonnin, Wireless sensor networks: a survey on recent developments and potential synergies, The Journal of Supercomputing 68 (1) (2014) 1–48.

[3] G. Simon, M. Maróti, A. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, K. Frampton, Sensor network-based countersniper system, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04, ACM, New York, NY, USA, 2004, pp. 1–12. doi:10.1145/1031495.1031497.

[4] S. Vasuhi, V. Vaidehi, Target tracking using interactive multiple model for wireless sensor network, Inf. Fusion 27 (2016) 41–53. doi:10.1016/j.inffus.2015.05.004.

[5] M.-T. Vo, T. T. Thanh Nghi, V.-S. Tran, L. Mai, C.-T. Le, Wireless sensor network for real time healthcare monitoring: Network design and performance evaluation simulation, in: V. V. Toi, T. H. Lien Phuong (Eds.), 5th International Conference on Biomedical Engineering in Vietnam, Springer International Publishing, 2015, pp. 87–91.

[6] L. Muduli, D. P. Mishra, P. K. Jana, Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review, Journal of Network and Computer Applications (2017) pp. 48–67. doi:https://doi.org/10.1016/j.jnca.2017.12.022.

[7] M. A. Moridi, M. Sharifzadeh, Y. Kawamura, H. D. Jang, Development of wireless sensor networks for underground communication and monitoring systems (the cases of underground mine environments), Tunnelling and Underground Space Technology 73 (2018) 127 – 138. doi:https://doi.org/10.1016/j.tust.2017.12.015.

[8] D. Todolí-Ferrandis, J. Silvestre-Blanes, S. Santonja-Climent, V. Sempere-Paya, J. Vera-Pérez, Deploy & forget wireless sensor networks for itinerant applications, Computer Standards & Interfaces 56 (2018) 27 – 40. doi:https://doi.org/10.1016/j.csi.2017.09.002.

[9] W.-H. Nam, T. Kim, E.-M. Hong, J.-Y. Choi, J.-T. Kim, A wireless sensor network (wsn) application for irrigation facilities management based on information and communication technologies (icts), Computers and Electronics in Agriculture 143 (2017) 185 – 192. doi:https://doi.org/10.1016/j.compag.2017.10.007.

[10] Y. Zhang, N. Meratnia, P. Havinga, Outlier detection techniques for wireless sensor networks: A survey, IEEE Communications Surveys Tutorials 12 (2) (2010) 159–170. doi:10.1109/SURV.2010.021510.00088.

[11] F. Martincic, L. Schwiebert, Distributed event detection in sensor networks, in: ICSNC '06. International Conference on Systems and Networks Communications, Tahiti, French Polynesia, 2006, pp. 43–43. doi:10.1109/ICSNC.2006.32.

[12] V. Hodge, J. Austin, A survey of outlier detection methodologies, Artificial Intelligence Review 22 (2) (2004) 85–126. doi:10.1023/B:AIRE.0000045502.10941.a9.

[13] R. A. Johnson, D. W. Wichern, et al., Applied multivariate statistical analysis, Vol. 5, Prentice Hall Upper Saddle River, NJ, 2002.

[14] V. Barnett, T. Lewis, Outliers in statistical data, Wiley, 1974.

[15] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier detection using replicator neural networks, in: Y. Kambayashi, W. Winiwarter, M. Arikawa (Eds.), Data Warehousing and Knowledge Discovery: 4th International Conference, DaWaK 2002 Aix-en-Provence, France, September 4-6, 2002 Proceedings, Springer Berlin Heidelberg, 2002, pp. 170–180. doi:10.1007/3-540-46145-0_17.

[16] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: Identifying density-based local outliers, in: SIGMOD Rec., Vol. 29, ACM, New York, NY, USA, 2000, pp. 93–104. doi:10.1145/335191.335388.

[17] B. Saneja, R. Rani, An efficient approach for outlier detection in big sensor data of health care, International Journal of Communication Systems 30 (17) (2017) 1–10. doi:10.1002/dac.3352.
URL http://dx.doi.org/10.1002/dac.3352

[18] N. Shahid, I. H. Naqvi, S. B. Qaisar, One-class support vector machines: analysis of outlier detection for wireless sensor networks in harsh environments, Artificial Intelligence Review 43 (4) (2015) 515–563.

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

[20] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, P. Deng, Localized outlying and boundary data detection in sensor networks, IEEE Transactions on Knowledge and Data Engineering 19 (8) (2007) 1145–1157. doi:10.1109/TKDE.2007.1067.

[21] L. M. A. Bettencourt, A. A. Hagberg, L. B. Larkey, Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks, in: J. Aspnes, C. Scheideler, A. Arora, S. Madden (Eds.), Distributed Computing in Sensor Systems: Third IEEE International Conference, DCOSS 2007, Santa Fe, NM, USA,

June 18-20, 2007. Proceedings, Springer Berlin Heidelberg, 2007, pp. 223–239. doi:10.1007/978-3-540-73090-3_15.

[22] B. Sheng, Q. Li, W. Mao, W. Jin, Outlier detection in sensor networks, in: Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '07, ACM, New York, NY, USA, 2007, pp. 219–228. doi:10.1145/1288107.1288137.

[23] A. Abid, A. Masmoudi, A. Kachouri, A. Mahfoudhi, Outlier detection in wireless sensor networks based on optics method for events and errors identification, Wireless Personal Communications 97 (1) (2017) 1503–1515. doi:10.1007/s11277-017-4583-7.
URL https://doi.org/10.1007/s11277-017-4583-7

[24] U. B. Nisha, N. U. Maheswari, R. Venkatesh, R. Y. Abdullah, Fuzzy-based flat anomaly diagnosis and relief measures in distributed wireless sensor network, International Journal of Fuzzy Systems 19 (5) (2017) 1528–1545.

[25] S. Rajasegarar, C. Leckie, M. Palaniswami, J. C. Bezdek, Distributed anomaly detection in wireless sensor networks, in: 10th IEEE Singapore International Conference on Communication Systems, Singapore, Singapore, 2006, pp. 1–5. doi:10.1109/ICCS.2006.301508.

[26] V. Chatzigiannakis, S. Papavassiliou, M. Grammatikou, B. Maglaris, Hierarchical anomaly detection in distributed large-scale sensor networks, in: 11th IEEE Symposium on Computers and Communications (ISCC'06), Cagliari, Italy, 2006, pp. 761–767. doi:10.1109/ISCC.2006.1691116.

[27] T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos, Distributed deviation detection in sensor networks, SIGMOD Rec. 32 (4) (2003) 77–82. doi:10.1145/959060.959074.

[28] V. S. K. Samparthi, H. K. Verma, Outlier Detection of Data in Wireless Sensor Networks Using Kernel Density Estimation, International Journal of Computer Applications 5 (6) (2010) 28–32. doi:10.5120/924-1302.

[29] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos, Online outlier detection in sensor data using non-parametric models, in: Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB '06, VLDB Endowment, Seoul, Korea, 2006, pp. 187–198.

[30] J. W. Branch, C. Giannella, B. Szymanski, R. Wolff, H. Kargupta, In-network outlier detection in wireless sensor networks, Knowledge and Information Systems 34 (1) (2013) 23–54. doi:10.1007/s10115-011-0474-5.

[31] C. Genest, J. Mackay, The joy of copulas: Bivariate distributions with uniform marginals, The American Statistician 40 (4) (1986) 280–283. doi:10.1080/00031305.1986.10475414.

[32] P. K. Trivedi, D. M. Zimmer, Copula Modeling: An Introduction for Practitioners, Vol. 1, Foundations and Trends in Econometrics, 2007. doi:10.1561/0800000005.

[33] M. J. Fischer, C. Köck, S. Schlüter, F. Weigert, Multivariate Copula Models at Work: Outperforming the desert island copula?, Discussion Papers 79/2007, Friedrich-Alexander University Erlangen-Nuremberg, Chair of Statistics and Econometrics (2007).
URL https://ideas.repec.org/p/zbw/faucse/792007.html

[34] R. B. Nelsen, An introduction to copulas, Vol. 139, Springer Science & Business Media, 2013.

[35] W. J. Conover, R. L. Iman, Rank transformations as a bridge between parametric and nonparametric statistics, The American Statistician 35 (3) (1981) 124–129. doi:10.1080/00031305.1981.10479327.

[36] G. Tolle, D. Culler, Design of an application-cooperative management system for wireless sensor networks, in: Proceeedings of the Second European Workshop on Wireless Sensor Networks., 2005, pp. 121–132. doi:10.1109/EWSN.2005.1462004.

[37] I. Rish, et al., An empirical study of the naive bayes classifier, in: IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Vol. 3-22, IBM New York, 2001, pp. 41–46.

[38] R. Salinas-Gutiérrez, A. Hernández-Aguirre, M. J. J. Rivera-Meraz, E. R. Villa-Diharce, Using gaussian copulas in supervised probabilistic classification, in: O. Castillo, J. Kacprzyk, W. Pedrycz (Eds.), Soft Computing for Intelligent Control and Mobile Robotics, Springer, Berlin, Heidelberg, 2011, pp. 355–372. doi:10.1007/978-3-642-15534-5_22.

[39] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[40] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 1–58. doi:10.1145/1541880.1541882.

[41] L. Tarassenko, P. Hayton, N. Cerneaz, M. Brady, Novelty detection for the identification of masses in mammograms, In Proc. of the Fourth International IEE Conference on Artificial Neural Networks 409 (1995) 442—-447.

[42] S. Madden, Intel lab data, Last accessed on 28-06-2017.
URL http://db.csail.mit.edu/labdata/labdata.html

[43] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, M. Parlange, Sensorscope: Application-specific sensor network for environmental monitoring, ACM Transaction. 6 (2) (2010) 1–32. doi:10.1145/1689239.1689247.

[44] C. E. Metz, Basic principles of roc analysis, in: Seminars in nuclear medicine, Vol. 8, Elsevier, 1978, pp. 283–298.