Global outliers detection in wireless sensor networks: A novel approach integrating time-series analysis, entropy, and random forest-based classification.

Mahmood Safaei; Maha Driss; Wadii Boulila; Elankovan A. Sundararajan; Mitra Safaei

Abstract

Wireless sensor networks (WSNs) have recently attracted greater attention worldwide due to their practicality in monitoring, communicating, and reporting specific physical phenomena. The data collected by WSNs is often inaccurate as a result of unavoidable environmental factors, which may include noise, signal weakness, or intrusion attacks depending on the specific situation. Sending high-noise data has negative effects not just on data accuracy and network reliability, but also regarding the decision-making processes in the base station. Anomaly detection, or outlier detection, is the process of detecting noisy data amidst the contexts thus described. The literature contains relatively few noise detection techniques in the context of WSNs, particularly for outlier-detection algorithms applying time series analysis, which considers the effective neighbors to ensure a global-collaborative detection. Hence, the research presented in this article is intended to design and implement a global outlier-detection approach, which allows us to find and select appropriate neighbors to ensure an adaptive collaborative detection based on time-series analysis and entropy techniques. The proposed approach applies a random forest algorithm for identifying the best results. To measure the effectiveness and efficiency of the proposed approach, a comprehensive and real scenario provided by the Intel Berkeley Research Laboratory has been simulated. Noisy data have been injected into the collected data randomly. The results obtained from the experiment then con-ducted experimentation demonstrate that our approach can detect anomalies with up to 99% accuracy.

Keywords

Anomaly detection, entropy, outlier detection, random forest, time series analysis, wireless sensor network

1 Introduction

Wireless sensor networks (WSNs) are drawing great interest worldwide, especially with the considerable progress of technologies that are leading to the apparition and enhancement of small smart sensors. With their reduced size, limited computing units, and condensed processing resources, these sensors are cheaper than their traditional counterparts. The nodes embedded in smart sensors enable them to detect data, measure it, and collect it from various points in the target environment. In addition, these nodes transfer sensory data into the sink, or base station, of the sensor, where decisions are processed and made.

These capacities mean that smart sensor nodes have low power requirements and are relatively simple devices despite their complex functions: most consists of the nodes themselves plus a power supply, processor, radio transmitter, memory, and actuator.[1]

A WSN is composed of multiple such wireless sensor devices, sometimes hundreds or thousands, implemented in a location determined by the user.[2] With WSNs, reliable communication is very important, and the literature has proposed several algorithms intended to guarantee a WSN's reception of reliable, less noisy data. Outlier detection algorithms have been listed in parts of the literature, but they have not been studied in as much depth as some other options.

An outlier is defined by[3] as "an observation that diverges to a large extent from other observations to give rise to doubts that it was produced by a separate method". In Reference 4, an outlier represents "an observation (or a set of observations) that seems to be inconsistent with the rest of the data in that set". Another definition of outliers as they relate to WSNs has also been provided by,[5] which is "the measurements that show significant deviation from the typical pattern of sensed data". There are several sources of outliers, which are detected in the data collected by WSNs such as event detection,[6-9] fault detection,[10,11] and intrusion detection.[12,13]

In general, outliers can be classified into two different categories, local or global.[14] The category that any particular outlier falls into can be determined based on the types and range of data surveyed and utilized in the process of detecting it.[15] The detection of local outliers is performed by considering a single sensor node and carried out either by identifying irregular values at the considered node on the basis of its own values collected previously or by using data from that node's neighbors. The outlier detection process in the second approach offers greater accuracy than the processes of the first; since this second approach takes into account the benefits that are gained from spatiotemporal correlations among the overall collected sensor data.[16,17] In addition, the second approach detects outliers in a more global perspective, which it accomplishes by considering the whole network. This also makes it possible to detect global noisy data at distinct network levels by considering the network typology.[18] In the case of centralized network architecture, all data are collected in the main sink node, which is where the outliers' detection is also performed. The main drawback of this latter method is the way in which it both increases overall response time and also generates additional costs for communication.[15]

In much of the related literature, several other methods have also been proposed for implementing outlier detection, which have included statistical modeling, information theory, Z-Score, and data mining-based methods.[5] The data mining-based method denotes the discovery of valuable and interesting information from extensive sources of data, and in this context, outlier detection in WSNs would be an appropriate area of application of this method.[19,20]

In recent years, the possibility for a quick, efficient, and accurate means of detecting outliers in WSNs has become of great interest to researchers since it can guarantee robust functionality of the affected network, the reliability of data thus collected and analyzed, and the generation of real-time event reports.[21] In addition, the detection of outliers in WSNs guarantees the analysis of the validity of the data and therefore reduces the communication costs of incorrect data. Furthermore, potential attacks on the network can be identified through the detection of outliers, which in turn can lead to an improvement of the network security.

In this article, we suggest a new approach to outlier detection, one with its basis in time-series modeling and forecasting with neighbors' collaboration. First, we start by extracting features allowing the time-series modeling and forecasting. Then, an adaptive entropy-based method is proposed to determine neighbor spatial-correlation. The third step of the proposed approach aims to determine the outliers and the anomaly data in each sensor node by performing a random forest classification algorithm.

The main contributions of the present article can be summarized in the following three points:

- The formulation of the problem of outlier detection in WSNs as a time-series analysis problem by considering the historically collected data;
- The proposition of an entropy-based method to select the best neighbor related to a considered sensor in order to ensure a spatiotemporal correlation useful for the outlier detection. This article focuses on evaluating the importance
  of temporal features and the correlation among the data of time-series data for the detection of outliers. The spatiotemporal correlation can be exploited to improve the overall network performance. The characteristics of the correlation in the WSN context can be classified into spatial and temporal correlations.[22] The first one relies on multiple sensors recording the same event. In this case, data are highly correlated with the recorded observations. For the second case, temporal correlations are recorded for many WSNs applications such as event tracking or area monitoring, especially when nodes periodically transmit observations about event features. Moreover, spatiotemporal correlation can bring important advantages when developing efficient communication protocols for the considered WSNs. For instance, data coming from spatially separated sensors are more important to the sink than highly correlated data from nodes in proximity.[22] Additionally, in the case of event tracking, temporal correlations play an important role in adjusting the frequency of measurement reporting which is essential in order to minimize energy expenditure. To the best of our knowledge, numerous research studies have been conducted about the outlier detection problem in WSNs but most of them mainly detect anomalies using offline data and few studies detect outliers using stream data. Offline anomaly detection can affect real-time decision-making, which conflicts with the WSN reliability concept. In addition, traditional outlier detection methods such as those based on a fixed threshold are not efficient since space and temporal conditions are changing dynamically. Therefore, reading data from neighbor nodes for spatial data will increase the accuracy of the proposed algorithm;
- The developmentandapplicationof a random forest-basedalgorithmusing time-seriesdata to globally identifyoutliers in each sensor node. This algorithm prevents from making incorrect decisions on the base station and also increases the lifetime of the network.

The remainder of this article is structured in the following way: in Section 2, the relevant literature and research on outlier and anomaly detection in WSNs are reviewed. In Section 3, the approach we propose is described in greater detail. In Section 4, experimental results carried out on a synthetic and real-world dataset (provided by Intel Berkeley Research Laboratory) are reported and analyzed. Section 5features concluding remarks on our results and consideration of future directions for related work.

2 RELATED WORKS

Detecting outliers in WSNs is a challenging problem due to certain characteristics of sensors: resource constraints (e.g., memory and computational speed), high costs of communication, and limited lifetime. The related literature has recommended different methods, most of which have been based on statistical or similar approaches.[23,24] The main objective of such approaches tends to concern approximating the distribution of sensor data, which in turn can be used to report outliers by computing probabilities or metrics like variance, correlations, mean, and so forth.[25]

Rajasegarar et al. in Reference 26 used a cluster-based method, where sensory data were combined into clusters utilizing a static width before using this set-up as the basis of comparison for other sensor nodes. This method did not require any in-depth knowledge of how data was distributed, but it did generate high additional costs in terms of communication.

Zhuang and Chen in Reference 27 proposed two outlier detection techniques. They extract the spatiotemporal correlations of measures that had been detected and attained by several sensor nodes. Rajasegarar et al.'s technique applies a wavelet analysis while Zhuang and Chen's technique uses a method of dynamic time warping. However, both techniques needed to set a specified threshold in order to detect the anomalies.

For the detection of outlying sensors and event boundary in SNs, Wu et al. in Reference 28 propose two algorithms. The first algorithm starts by calculating, for each sensor, the difference between its reading value and the median reading value obtained from its neighboring reading values. Then, each sensor node collects the differences from its neighborhood and standardizes them. The last step permits the decision of whether the sensor considered is an outlier or not, which is done by comparing the absolute value of its standardized difference with a fixed threshold. If this value is larger than this threshold, the considered sensor is then identified as an outlier. This algorithm is exploited in the second proposed algorithm to localize event sensors at an event boundary. The approach proposed in this article depends on the specific characteristics/constraints of the communication network and the proposed detection algorithms are based on semidetected and sometimes incorrect data, which are collected from a randomly selected neighbor. An enhanced version of the proposed approach in Reference 28 is presented in Reference 29. In this article, the outlying sensor detection algorithm is enhanced by considering a temporal correlation between sensor nodes. The proposed algorithm in Reference 29 uses the median of the k nearest neighbors for each sensed data and compare it with the locally saved data in the corresponding sensor. The proposed method improves the accuracy of the detection algorithm but in return, the new proposed algorithm requires additional computational costs.

Sheng et al. in Reference 30 proposed a histogram-based technique that would ensure global outlier detection in WSNs. Rather than sending out all sensory data to the base station, with this method each sensor node kept a summary containing the relevant sensed data on a separate sliding window. Then, using the elaborated summaries collected this way, the base station could extract the distribution of data and filter for typical data only. With this method, outliers tend to be remarked if their measures passed a static threshold value. The principal disadvantage of this article, though, is found in the availability that can occur at unplanned intervals in the base station, and which can cause the shutdown of the entire analysis system.

Moreover, this method is limited to applications to one-dimensional data where the spatial distance between the sensor nodes is important.

The research conducted by Abid et al. in Reference 31 proposes a density-based clustering method ordering points for ensuring outlier detection. This method is performed without knowing in advance the number or the labels of the clusters, and it is applied independently of certain constraints related to the considered network (e.g., the topology, the change in scalability, and the form of the collected data). In this article, the "Ordering Points To Identify the Clustering Structure" (OPTICS) method is used to analyze the collected data by applying a density-based clustering algorithm, which ensures the classification of data into events and errors. The limitations of Reference 31 consist in two major points: (1) the proposed method has a handicap to detect an outlier in a huge number of normal values, and (2) it is more robust to detect possible outliers if the learning window is not very big.

Barakkath et al. in Reference 32 proposed a fuzzy-based approach for outlier detection. This article applied a subtractive clustering method. The dataset, which is used for the provided experiments, is divided into multiple sets in which the likenesses within sets are greater than those between the peers. Here, outlier detection is performed by adopting a Takagi–Sugeno fuzzy model to account for the function and selection of parameter membership. In this article, the suggested approach has been applied to a WSN that is divided into clusters and thus is unavailable for application to other networks architectures. In addition, our approach tackles outliers in 2D datasets only, and therefore cannot identify anomalies in datasets with greater dimensions.

The outlier detection in healthcare applications is studied by Saneja and Rani in Reference 24. In this article, the authors proposed an approach to outlier detection that was based on the sequential minimalization optimization (SMO) derived from correlation and dynamic regressions. During the initial stage, the values of the correlation coefficient are computed and sorted in order to identify the pairs of strongly correlated sensor nodes. In the second stage, anomalies in individual sensors are identified by applying the sequential minimal optimization regression algorithm (SMOReg). To speeduptheprocessingofbigdata,theproposedapproachinReference24reliesonaHadoopMapReduceframework.[33,34] Despite the high scalability of the proposed approach, the latter is applicable only to data that have linear correlation among the considered attributes, which is not true in certain areas of WSNs where measurements cannot be presented linearly.

Identifying outliers may also be performed by calculating the density associated with sensory data measures within a target area. This calculation of density can be executed in an evenly distributed manner. In Reference 35, a Local Outlier Factor (LOF) method is proposed. This method consists of drawing a circle around "k" measures, where depending on the density level obtained, it attributes an "outlier metric" parameter to each measure, which determines whether or not each such measure should be defined as an outlier. To guarantee a high level of accuracy, it may be necessary to execute the LOF method with numerous values of "k", which in turn may lead to increases in the cost of computation.

In Reference 36, Qiao et al. propose a method combining deep belief network and online quarter-sphere one-class support vector machine to perform outlier detection for large-scale and high-dimensional datasets of WSNs. First, a training process that learns the radius of the quarter sphere is applied. Then, online testing is proposed to perform online outlier detection without supervision. To validate the proposed method, four large-scale datasets having dimensions ranging from 54 to 561 are used. The

proposed method is compared with three competitive methods using two metrics, which are classification accuracy and computational time. In this article, the performance of the proposed method should be demonstratedbyitscomparisonwithotheroutlierdetectionmethodsthroughthecomputationofadditionalperformance metrics.

In Reference 17, Safaei et al. proposed a local outlier detection algorithm that would run on each individual sensor nodeofthewirelessnetworkunderconsideration.Theproposedapproachofferedthreeadvantages:(1)areductionmechanism allowing to eliminate the noneffective features; (2) a prior determination of what size the resulting data histogram memory would be, to ensure efficient use of the available memory; and finally (3) the adaptive Bayesian-network-based classification applied to predict noisy data. Experiments were conducted on real datasets and depicted good accuracy of outlier prediction compared to the existing state-of-the-art methods. This article is applied to ensure only the local outlier detection and the presented experimentation is not extended to include the global outlier detection.

Gupta et al. in Reference 37 employ the outlierness factor-based on neighborhood (OFN) technique for outlier detection and analysis in sensor networks. In the proposed approach, the neighborhood points are first determined. Then, the weight of the neighborhood data is calculated. The OFN technique is employed to classify the outlier data points as events and errors based on spatial and temporal correlations, which are neighborhood readings and timestamps of readings, respectively. The main disadvantage of the proposed approach is that the experiments presented in this article are conducted using only low dimensional datasets containing between 50 and 100 r-neighbors, which are the nearest neighbors for specific data.

A time-series denoising autoencoder (TSDA) network is proposed by Wang et al. in Reference 38 to compress the discriminative high-dimensional monitoring data to ensure the representation of the temporal and spatial features of the detection points. In addition, a Gaussian model is used for anomaly point detection in wireless sensor networks. This model is based on auxiliary target variables to gain the anomaly points by employing an objective function of region partitioning. The limitation of the proposed approach is that it performs a slight disadvantage with low-dimensional datasets presenting a limited number of spatial–temporal features.

In the next section, we detail our proposed approach for global outlier detection in WSNs.

3 PROPOSED APPROACH FOR GLOBAL OUTLIER DETECTION IN WSNS

The approach we propose to global outlier detection in WSNs is depicted in Figure 1. It is modeled as a process that consists of three sequential steps. The first step includes three parallel substeps, which are: (1) reading the actual data that are collected from the considered sensor *S*, (2) reading the historical data stored in the memory of *S*, and (3) searching neighbors of *S*, selecting the best neighbor, and reading the actual data from the selected neighbor. The second step aims to ensure the computation of features by using the collected data (i.e., actual data collected from *S*, historical data stored in *S*, and actual data collected from the best neighbor of *S*). The last step applies the outlier detection algorithm to determine outlier data and normal/healthy data.

3.1 Reading sensory data

This step aims to prepare sensor data to be evaluated for outlier detection. Two types of data are distinguished: data that are specific to a selected sensor and those that are specific to the best neighbor of the considered sensor. Indeed, to detect noise globally, it is necessary to select neighbors that can potentially collaborate with the considered sensor. It is important to determine how many neighbors must be selected and which sensor is the most effective for the collaboration. Hence, for the neighbor selection, a simulation of the Monte Carlo algorithm is conducted. This choice is justified by the fact that this algorithm has shown its usefulness in this context of use, which has been proven in the recently conducted research.[35] After the neighbor selection phase, adaptive entropy and a greedy algorithm are applied to the data that have the same timestamp as their neighbors, and this in order to select the sensor that can collaborate more effectively with the considered sensor to globally detect noise in the considered network.

The following subsections outline our process for searching the sensor neighbors and selecting the best one.

3.1.1 Searching sensor neighbors

To detect the outlier data, every local sensor has to find the best neighbors in order to collaborate with them. This is performed by applying a Monte Carlo simulation. For this purpose, a range of neighbors from 1 to 10 has been selected and a matrix *ne* has been created: *ne* = {1,2,3,4,5,6,7,8,9,10}. In addition, 10 sensors are, randomly selected based on their distance and coverage area and a matrix D_s has been created: $D_s = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}\}$. The result of the simulation shows that the best number of neighbors ensuring an effective spatial collaboration is four. The sensor nodes with the nearest distance are more reliable and more accurate compared with others at greater distances.

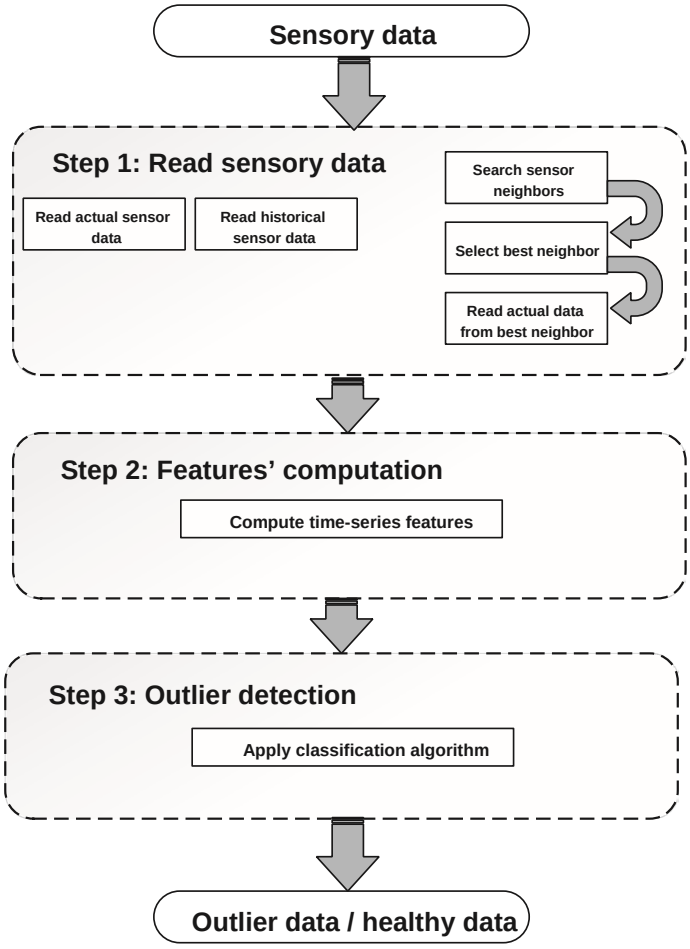

FIGURE 1 Steps of the proposed approach

3.1.2 Selecting the best neighbor

After searching neighbors for collaboration, the next step is to calculate and identify the best neighbor. This latter will participate with the local sensor data in the classification algorithm. For this matter, sensors will keep the latest 10 data from the selected neighbors.[17] The input data frame is shown below:

$$D = \begin{pmatrix} d_{S_1}^{t_n} & d_{S_1}^{t_{n-1}} & d_{S_1}^{t_{n-2}} & d_{S_1}^{t_{n-3}} & d_{S_1}^{t_{n-4}} & d_{S_1}^{t_{n-5}} & d_{S_1}^{t_{n-6}} & d_{S_1}^{t_{n-7}} & d_{S_1}^{t_{n-8}} & d_{S_1}^{t_{n-9}} \\ d_{S_2}^{t_n} & d_{S_2}^{t_{n-1}} & d_{S_2}^{t_{n-2}} & d_{S_2}^{t_{n-3}} & d_{S_2}^{t_{n-4}} & d_{S_2}^{t_{n-5}} & d_{S_2}^{t_{n-6}} & d_{S_2}^{t_{n-7}} & d_{S_2}^{t_{n-8}} & d_{S_2}^{t_{n-9}} \\ d_{S_3}^{t_n} & d_{S_3}^{t_{n-1}} & d_{S_3}^{t_{n-2}} & d_{S_3}^{t_{n-3}} & d_{S_3}^{t_{n-4}} & d_{S_3}^{t_{n-5}} & d_{S_3}^{t_{n-6}} & d_{S_3}^{t_{n-7}} & d_{S_3}^{t_{n-8}} & d_{S_3}^{t_{n-9}} \\ d_{S_4}^{t_n} & d_{S_4}^{t_{n-1}} & d_{S_4}^{t_{n-2}} & d_{S_4}^{t_{n-3}} & d_{S_4}^{t_{n-4}} & d_{S_4}^{t_{n-5}} & d_{S_4}^{t_{n-6}} & d_{S_4}^{t_{n-7}} & d_{S_4}^{t_{n-8}} & d_{S_4}^{t_{n-9}} \end{pmatrix}. \quad (1)$$

The identification of the best neighbor is based on an adaptive entropy function. This function aims to calculate the weight of each sensor node in order to select the best neighbor. This function is deducted from the following equations: 2,3,4,5,6,7,8,9,10,11, and 12.

$$d = \{d_{S_x}^{t_n}, d_{S_x}^{t_{n-1}}, d_{S_x}^{t_{n-2}}, d_{S_x}^{t_{n-3}}, d_{S_x}^{t_{n-4}}, d_{S_x}^{t_{n-5}}, d_{S_x}^{t_{n-6}}, d_{S_x}^{t_{n-7}}, d_{S_x}^{t_{n-8}}, d_{S_x}^{t_{n-9}}\}, \quad (2)$$

where d is the last 10 history data of each neighbor sensor s_x and t_n is the current time.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{10} d_i, \quad n = 10, \quad (3)$$

where $\bar{x}$ is the mean of the history data of d.

$$e = \frac{d_{S_x}^{t_n} - \bar{x}}{\bar{x}}, \quad (4)$$

e is the deviation of $d^{t}{}_{s}{}^{n}{}_{x}$ from x, where t_n is the current time and s_x is the neighbor sensor.

$$f_i = \begin{cases} h_i = -2, & \text{if } e \le -0.5 \\ h_i = -1, & \text{if } -0.5 < e \le 0 \\ h_i = 1, & \text{if } 0 < e \le 0.5 \\ h_i = 2, & \text{if } e > 0.5 \end{cases}, \quad (5)$$

h_i is the classification of each e value based on the defined condition.

$$a_0 = \sum_{i=1}^{n} h_i \Rightarrow h_i = -2, \quad (6)$$

$$a_1 = \sum_{i=1}^{n} h_i \Rightarrow h_i = -1, \quad (7)$$

$$a_2 = \sum_{i=1}^{n} h_i \Rightarrow h_i = 1, \quad (8)$$

$$a_3 = \sum_{i=1}^{n} h_i \Rightarrow h_i = 2, \quad (9)$$

where a_0,a_1,a_2, and a_3 are the total number of h_i values.

$$s = \sum(a_0, a_1, a_2, a_3), \tag{10}$$

s is the sum of all the (a_0,a_1,a_2,a_3) variables.

$$en_t = \begin{cases} en_0 = -(\frac{a_0}{s}) \times \log(\frac{a_0}{s}) \\ en_1 = -(\frac{a_1}{s}) \times \log(\frac{a_1}{s}) \\ en_2 = -(\frac{a_2}{s}) \times \log(\frac{a_2}{s}) \\ en_3 = -(\frac{a_3}{s}) \times \log(\frac{a_4}{s}) \end{cases}, \tag{11}$$

where en_t is the calculated weight for each variable a.

$$N_b = \max(en_t), \tag{12}$$

N_b is the best selected neighbor obtained by choosing the maximum value of en_t.

After selecting the best neighbor, we propose to calculate the corresponding features and build the feature matrix to be used by the classification algorithm. The major problem in WSNs is the limitation of resources such as dependence on batteries as power sources, very limited central processing unit (CPU) and memory capacity, and so forth. Certainly, increasing the number of features has a direct effect on the outlier detection algorithm's accuracy. However, realistically, it is not feasible to consider multiple features for the case of a single sensor node, and this is due to the previously mentioned limitation of WSNs. In this article, the feature matrix is composed of variables taken from the actual data of the best neighbor (e.g., temperature, pressure, humidity, etc.) and four features computed based on the actual and historical data of the considered sensor.

3.2 Features' computation

This step is intended to compute a set of features based on collected data (actual data collected from a chosen sensor *S* and historical data stored in *S*). In this article, four features are computed. These features are Pearson correlation, Spearman ranking correlation, distance correlation, and correlation relationship. These features have been considered in several previous related works and they have provided good results.[39-45]

3.2.1 Pearson correlation feature

Examining the relationships between variables is very important in classification algorithms. In this article, we propose to use the "Pearson correlation coefficient", also known as the "product-moment correlation coefficient". This statistical coefficient, denoted in our case by *r*, helps to estimate the relationship

between two variables. A value that is close to 0 indicates that there is no relationship between variables, whereas an absolute value that is close to 1 indicates a strong relationship. Generally, the Pearson coefficient is affected by nonlinear behavior. Hence, in our article, the Pearson correlation is measured using an adaptive entropy function to overcome the problem of nonlinear behavior.

Let us suppose two variables *x* and *y*. Equation 13 demonstrates how the Pearson correlation coefficient between this *x* and *y* is calculated:

$$r = \frac{SS_{xy}}{\sqrt{SS_x SS_y}}, \tag{13}$$

SS_x and SS_y represent the sums of the squared scores of *x* and *y*, respectively. Whereas, SS_{xy} represent the sum of the products of the squared scores of *x* and *y*.

SS_x is calculated using Equation 14, where $\bar{x}$ is the mean of the *x* sample and *n* is the size of this sample.

$$SS_x = \sum_{i=1}^{n} (x_i - \bar{x})^2. \tag{14}$$

The main challenge when calculating SS_x using Equation 14 is the computing time in case of considering a big dataset. Therefore, the sum of squares can be also calculated using Equation 15 in order to overcome the problem of time-consuming computation.

$$SS_x = \sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}. \tag{15}$$

Following the same process, we can calculate the sum of squares for *y* by modifying *x* by *y* in Equation 15.

The sum of the products of the squared scores of *x* and *y* is computed using Equation 16.

$$SS_{xy} = \sum_{i=1}^{n} (x_i y_i) - \frac{\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n}. \tag{16}$$

3.2.2 Spearman ranking correlation feature

The Spearman ranking correlation is a nonparametric coefficient that is used to measure the level of relationship between two variables. This coefficient is suitable for correlation analysis once the variables' values are converted into ordinal scales. Equation 17 is used to calculate the Spearman ranking correlation coefficient:

$$SS_{xy} = \sum_{i=1}^{n}(x_i y_i) - \frac{\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n}. \quad (16)$$

The ρ values are between −1 and +1.

When the ρ is close to −1 or +1, this indicates an important correlation between the considered variables. However, when the value is close to zero, we conclude that there is a weak correlation between the variables.

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}. \quad (17)$$

3.2.3 Distance correlation feature

To measure the distance correlation between sets of random variables, the Fourier transform is applied.

Assume p is a positive number and $X = (X_1, \ldots, X_p) \in R^p$ is a random vector. In vector $s = (s_1, \ldots, s_p) \in R^p$, the norm $||s|| = (s_1^2 + \cdots + s_p^2)^{1/2}$ depicts the standard Euclidean norm on R^p.

Further, let us consider $\langle s,X\rangle = s_1X_1 + \ldots + s_pX_p$ the standard inner product of s and X.

Let us also consider the positive numbers q and α, a vector $t \in R^q$, and finally a random vector $Y \in R^q$. The inner product $\langle t,Y\rangle$ and the Euclidean norm $||t||$ on R^q are depicted as follows.

The common characteristic function of random vectors (X,Y) is given by Equation 18:

$$\phi_{X,Y}(s,t) = \mathbb{E}\exp[\sqrt{-1}\langle s,X\rangle + \sqrt{-1}\langle t,Y\rangle], \quad (18)$$

where $\phi_X(s) = \phi_{X,Y}(s,0) = {}^{E}\exp[\sqrt{-1}\langle s,X\rangle]$ and $\phi_Y(t) = \phi_{X,Y}(0,t) = {}^{E}\exp[\sqrt{-1}\langle t,Y\rangle]$ are the marginal characteristic functions of Y and X. If $\phi_{X,Y}(s,t) = \phi_X(s)\phi Y(t)$, then X and Y are independent for any $s \in R^p$ and $t \in R^q$.

For random vectors X and Y, the covariance distance is a non-negative number (X,Y), here defined by Equation 19:

$$\mathcal{V}^2(X,Y) = \frac{1}{c_p c_q}\int_{\mathbb{R}^q}\int_{\mathbb{R}^p}\frac{|\phi_{X,Y}(s,t) - \phi_X(s)\phi_Y(t)|^2}{\|s\|^{p+1}\|t\|^{q+1}}\mathrm{d}s\mathrm{d}t, \quad (19)$$

where $c_p = \frac{\pi^{(p+1)/2}}{\Gamma((p+1)/2)}$.

The correlation distance between X and Y is expressed by Equation 20:

$$R(X,Y) = \frac{\mathcal{V}(X,Y)}{\sqrt{\mathcal{V}(X,X)} \cdot \sqrt{\mathcal{V}(Y,Y)}}. \tag{20}$$

The distance correlation is denoted by T and is given by Equation 21. Values of T are in [0,1] and T is equal to zero if $\varphi_{X,Y} = \varphi_X \varphi_Y \mu$ − a.e.

$$T(X,Y;\mu) = \int_{\mathbb{R}^{p+q}} |\varphi_{X,Y}(s,t) - \varphi_X(s)\varphi_Y(t)|^2 \mu(ds,dt), \tag{21}$$

where $\varphi_X(t) = {}^{E}[e^{i\langle t,Z\rangle}], t \in R^d$ denotes a characteristic function and $X \in R^d$ a random vector.

When μ has a Lebesgue density with positive number on R^{p+q} and if $T(X,Y;\mu) = 0$, this may result that $X \perp Y$.

An empirical version $T_n(X,Y;\mu)$ of $T(X,Y;\mu)$ is obtained if attributes in Equation 21 are changed by their corresponding empirical versions. Then, based on the distribution of T_n under the *null* hypothesis, X and Y are considered as independent.

3.2.4 Correlation relationship feature

The correlation coefficient, named r, allows measuring the linearity relationship between two variables. The correlation coefficient can take any value between −1 and +1.

The interpretation of the values of the correlation coefficient is as follow:

- 0 demonstrates a nonlinear relationship;
- +1 demonstrates a good “positive linear relationship”. When the values of a single variable increase, then the values of another variable will also increase;
- −1 demonstrates a good “negative linear relationship”. When the values of a single variable decrease, then the values of another variable will decrease also;
- Values that fall between 0 and 0.3 (or −0.3 and 0) demonstrate a weak positive (negative) relationship using a shaky linear relationship rule;
- Values that fall between 0.3 and 0.7 (or −0.7 and −0.3) demonstrate a moderate positive (negative) linear relationship using a fuzzy firm linear rule;
- Values that fall between 0.7 and 1.0 (or −1.0 and −0.7) demonstrate a strong positive (negative) linear relationship using a firm linear rule;
- The value of r^2, also termed the coefficient of determination, shows that r^2 tends to be understood as the per cent of the variation of one variable produced by another variable, or the per cent of variation that is shared between two variables.

To calculate the correlation coefficient of two variables *X* and *Y*, let us consider *zX* and *zY* the standardized versions of *X* and *Y*, respectively. Both *zX* and *zY* are restated to represent means equaling 0 as well as standard deviations of 1. The expressions we used in order to obtain these standardized scores are represented in Equations 22 and 23:

$$\mathrm{zX}_i = [\mathrm{X}_i - mean(X)]/\mathrm{s.d.}(X), \tag{22}$$

$$\mathrm{zY}_i = [\mathrm{Y}_i - mean(Y)]/\mathrm{s.d.}(Y). \tag{23}$$

The correlation coefficient can be defined as the mean product of these standardized scores (zX_i,zY_i), as expressed in Equation 24:

$$r_{X,Y} = sum \ \text{ of } \ [\mathrm{zX}_i \times \mathrm{zY}_i]/(n-1), \tag{24}$$

where *n* represents the sample size.

Features that are calculated based on three forms of collected data (actual data from *S*, historical data stored in *S*, and actual data collected from the best neighbor of *S*) are expressed as follow:

1. Pearson correlation feature → f_1
2. Spearman rank correlation feature→ f_2
3. Distance correlation feature → f_3
4. Correlation coefficient feature → f_4
5. Variable from the actual data collected from the best neighbor→ f_n

Hence, we are able now to construct a feature matrix, denoted by FeatMatrix, that will be used by the outlier detection algorithm as it is shown by Equation 25.

$$\text{Feat Matrix} = \begin{pmatrix} f1_{s_1} & f2_{s_1} & f3_{s_1} & f4_{s_1} & fn_{s_1} \\ f1_{s_2} & f2_{s_2} & f3_{s_2} & f4_{s_2} & fn_{s_2} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ f1_{s_n} & f2_{s_n} & f3_{s_n} & f4_{s_n} & fn_{s_n} \end{pmatrix}. \tag{25}$$

3.3 Outlier detection

In this article, we compare five different classification algorithms, which are: random forest (RF),[46] Naive Bayes (NB),[47] k-nearest neighbors (kNN),[30] support vector machine (SVM),[48] and neural network (NN).[49] In this article, these five classification algorithms are tested using the features previously detailed in order to determine the best algorithm, which provides the highest accuracy. The proposed experiments are decentralized, with algorithms are running on each sensor node. In this case, it is important to consider the size of the memory that is used by the data history at each node in addition to the accuracy.

4 EXPERIMENTATIONS

4.1 Dataset description

This section details the simulation steps followed in order to evaluate the performance of the proposed outlier detection algorithm. MATLAB and R programming tools are used to simulate the results depicted in this article. Experiments were conducted using a dataset from the Intel Berkeley Research Laboratory,[50] which is one of the most frequently-used datasets in several recent works, such as.[31] The data collected from 54 individual sensor nodes deployed in the Intel Berkeley Research Laboratory between February 28 and April 5, 2004 has been gathered in a dataset that includes reading data of approximately 2.3 M records. In the Intel Berkeley Research Laboratory, Mica2Dot sensors with weatherboards have been used. Mica2Dot sensors are third generation mote modules that are employed to enable the deployment of low power WSNs. These sensors allow the collection of time-stamped topological information, as well as humidity, temperature, light, and voltage values every 31 s. These data were collected using the TinyDB network query processing system, built on the TinyOS platform.[50] Figure 2 presents a schematic of the sensor nodes' positioning in the test environment thus considered.

Thisdatasetcollectsseveraltypesofsensorydataproperties,rangingfromtemperature,humidity,environmentallight, and sensor node battery voltage. Different types of information collected by sensors are displayed in the following formats: "Date (yyyy-mm-dd)," "Time (hh:mm:ss.xxx)," "Epoch (Integer)", "moteid (Integer)," "Temperature (Real), Humidity (Real)," "Light (Real)," and "Voltage (Real)". All data were initially collected on intervals of a 31 s timestamp.

4.2 Results and discussion

To ensure an effective detection of outliers, the best neighbors for each sensor node are selected. Table 1 presents sensor neighbors resulting from the execution of the Monte Carlo simulation on the considered scenario, as illustrated by Figure 2.

In this article, five classification algorithms, namely RF, NB, kNN, SVM, and NN, are used to identify outliers in the dataset previously mentioned.

The parameters' values being considered for the five classification algorithms are detailed in our previous work.[17]

The number of decision trees is one of the most important parameters in RF algorithms. To obtain an accurate result using the RF method, hundreds or thousands of decision trees are created. When the number of trees increases, the accuracy of results also increases. However, sometimes a larger number of decision trees can affect the system's performance, especially since sensor nodes have limited resources.

Figure 3 shows a sample of a decision tree used by the RF algorithm to ensure outliers' detection.

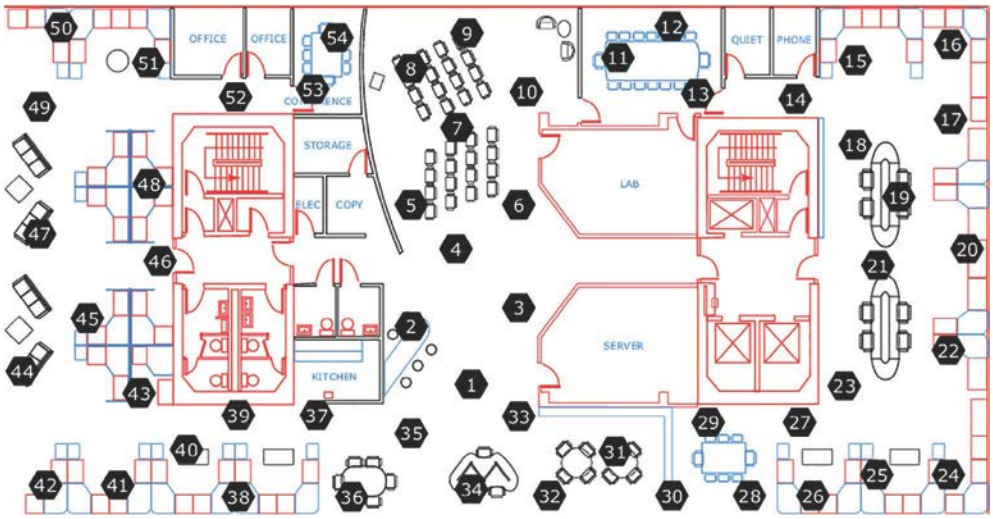

FIGURE 2 Schematic of the sensors' positioning

TABLE 1 Sensor nodes with their selected neighbors

Sensor node	Neighbors
S1	S31, S2, S3, S33
S2	S1, S3, S4, S35
S3	S1, S4, S2, S5
S4	S5, S3, S2, S6
S5	S4, S6, S3, S9
S6	S9, S7, S5, S8
S7	S52, S8, S51, S6
S8	S7, S9, S52, S10
S9	S8, S10, S6, S7
S10	S9, S11, S12, S8
S11	S10, S12, S13, S9
S12	S11, S13, S10, S9
S13	S12, S16, S11, S15
S14	S15, S13, S16, S17

S15	S16, S17, S14, S13
S16	S17, S15, S13, S19
S17	S16, S18, S19, S15
S18	S19, S17, S20, S16
S19	S18, S17, S20, S21
S20	S21, S19, S18, S22
S21	S25, S20, S19, S23
S22	S23, S24, S20, S21
S23	S22, S24, S25, S26
S24	S26, S23, S25, S28
S25	S21, S27, S24, S26
S26	S24, S28, S25, S27
S27	S29, S25, S28, S26
S28	S26, S29, S30, S27
S29	S27, S28, S30, S31
S30	S29, S28, S32, S31
S31	S1, S29, S32, S33
S32	S30, S33, S31, S34
S33	S35, S32, S34, S1
S34	S36, S33, S32, S35

S35	S37, S33, S34, S36
S36	S34, S38, S37, S35
S37	S35, S38, S36, S41
S38	S37, S39, S36, S41
S39	S40, S38, S36, S41
S40	S39, S38, S41, S42
S41	S38, S37, S42, S43
S42	S43, S41, S40, S45
S43	S42, S44, S41, S45
S44	S43, S45, S46, S41
S45	S43, S44, S46, S47
S46	S45, S47, S50, S49
S47	S49, S48, S46, S45
S48	S49, S47, S50, S46
S49	S48, S47, S50, S46
S50	S51, S49, S46, S52
S51	S50, S52, S7, S6
S52	S7, S51, S8, S50

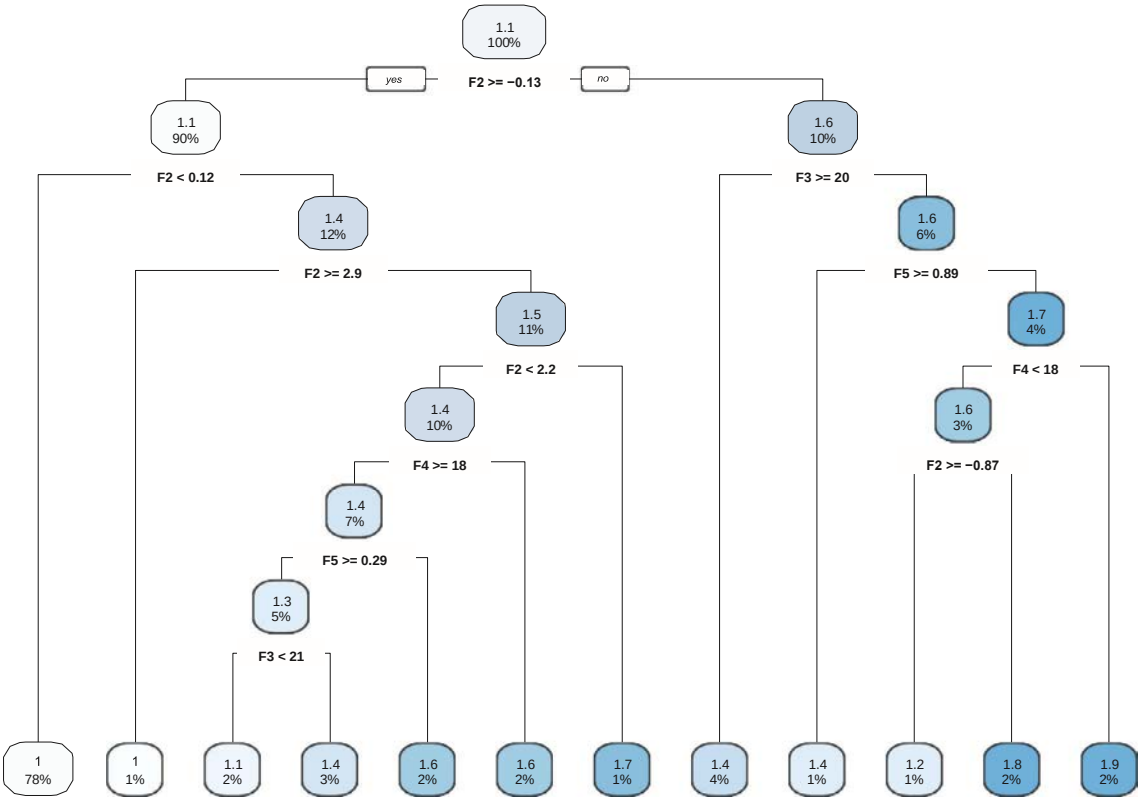

FIGURE 3 Sample of one decision tree used by the RF algorithm

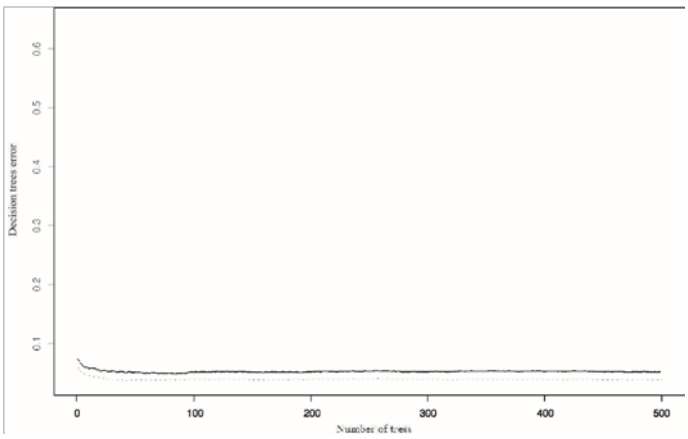

FIGURE 4 Evaluation of the error rate of the RF algorithm according to the trees' number

Figure 4 shows that with 36 decision tress, RF algorithm achieves the optimum error reduction.

Table 2 illustrates a comparison between the five considered classification algorithms with a noise level of 10%, 15%, and 20% of the total data. "Actual" data represent real data and "prediction" data represent classified data or the output of the classification method. 0 depicts normal data, whereas the value 1 depicts outlier data. The Σ represents the sum of values.

Figure 5 depicts a comparison of the accuracy of the outlier detection between the five considered classification algorithms. This figure shows that an RF algorithm can detect the outlier data with 99.1%

accuracy in 10% noisy sensory data followed by kNN, NN, NB, and SVM. The outlier detection accuracy of RF will decrease very slowly with the increase of the noisy data but it still has the best accuracy compared to the other algorithms. With a huge amount of noisy sensory

TABLE 2 Confusion matrices for the five classification algorithms

Classification algorithm	Data with noise level %		
	10%	15%	20%
RF	Prediction: 0 / 1 / Σ Actual 0: 335724 / 762 / 336486 Actual 1: 2236 / 35158 / 37394 Σ: 337960 / 35920 / 373880	Prediction: 0 / 1 / Σ Actual 0: 316613 / 1194 / 317807 Actual 1: 2721 / 53352 / 56073 Σ: 319334 / 54546 / 373880	Prediction: 0 / 1 / Σ Actual 0: 297227 / 1892 / 299119 Actual 1: 3147 / 71614 / 74761 Σ: 300374 / 73506 / 373880
kNN	Prediction: 0 / 1 / Σ Actual 0: 332159 / 4327 / 336486 Actual 1: 10165 / 27229 / 37394 Σ: 342324 / 31556 / 373880	Prediction: 0 / 1 / Σ Actual 0: 309955 / 7852 / 317807 Actual 1: 15325 / 40748 / 56073 Σ: 325280 / 48600 / 373880	Prediction: 0 / 1 / Σ Actual 0: 287298 / 11821 / 299119 Actual 1: 20666 / 54095 / 74761 Σ: 307964 / 65916 / 373880
NB	Prediction: 0 / 1 / Σ Actual 0: 332089 / 4397 / 336486 Actual 1: 10000 / 27394 / 37394 Σ: 342089 / 31791 / 373880	Prediction: 0 / 1 / Σ Actual 0: 299812 / 17995 / 317807 Actual 1: 45810 / 10263 / 56073 Σ: 345622 / 28258 / 373880	Prediction: 0 / 1 / Σ Actual 0: 260830 / 38289 / 299119 Actual 1: 48251 / 26510 / 74761 Σ: 309081 / 64799 / 373880
SVM	Prediction: 0 / 1 / Σ Actual 0: 171216 / 165270 / 336486 Actual 1: 22640 / 14754 / 37394 Σ: 193856 / 180024 / 373880	Prediction: 0 / 1 / Σ Actual 0: 159517 / 158290 / 317807 Actual 1: 33270 / 22803 / 56073 Σ: 192787 / 181093 / 373880	Prediction: 0 / 1 / Σ Actual 0: 144626 / 154493 / 299119 Actual 1: 44063 / 30698 / 74761 Σ: 188689 / 185191 / 373880
NN	Prediction: 0 / 1 / Σ Actual 0: 171216 / 165270 / 336486 Actual 1: 22640 / 1475 / 37394 Σ: 193856 / 180024 / 373880	Prediction: 0 / 1 / Σ Actual 0: 312303 / 5504 / 317807 Actual 1: 25386 / 30687 / 56073 Σ: 337689 / 36191 / 373880	Prediction: 0 / 1 / Σ Actual 0: 277344 / 21775 / 299119 Actual 1: 20156 / 54605 / 74761 Σ: 297500 / 76380 / 373880

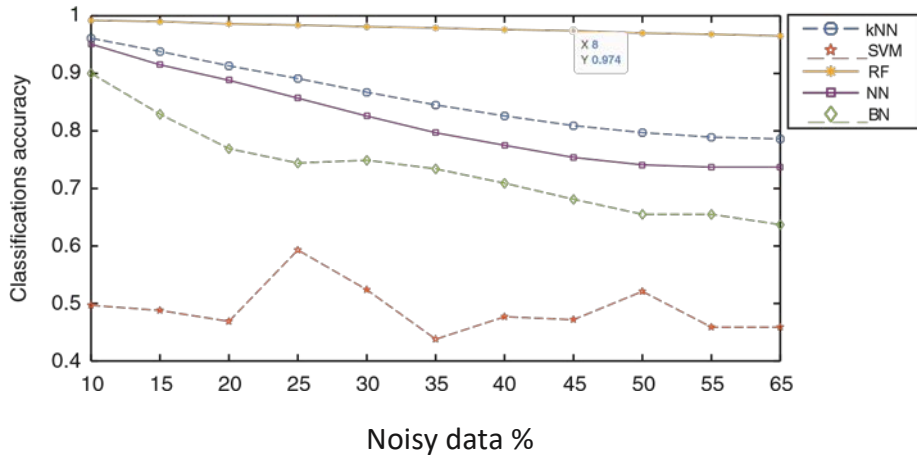

FIGURE 5 Comparison of the accuracy between the five classification algorithms according to the percentage of noisy data

data, RF can detect 97.8% of outlier data but the accuracy of the kNN algorithm drops dramatically to less than 80%. In this article, SVM provides the most inaccurate results for the outlier detection problem compared to the other four classification algorithms. An important extension of this article will be to combine results of the five classifiers instead of using only one of them.[51]

When the level of noise increases, the gap between noisy data and healthy data will also increase; therefore, the classification algorithms can detect outliers more accurately. Figure 6 shows a scenario with a noise-level σ equals to 0, 5, and 10. For this scenario, the accuracy of the RF algorithm has increased from 98% to 99%.

But σ, or noise level, is not the only factor influencing the outlier detection accuracy. Another important factor is the total of noisy data in the considered dataset. This article shows that when the amount of noisy data increases, the accuracy of the outlier detection algorithm decreases. Figure 6 demonstrates that when the σ value increases, the algorithm can identify 100% of outliers data up to 20% of noisy data.

In this article, three simulation rounds were executed with the same configuration and while changing the value of σ in each round to test the accuracy and the behavior of the five classification algorithms (RF, NB, kNN, SVM, and NN). The value of σ was changed during the simulation rounds as follow $\sigma = \{5, 7.5, 10\}$. The increase of the value of σ has a direct effect on the increase of the noise level since σ is one of the main values in the Gaussian noise. Figures 7, 8 and 9 show that the accuracy of the outlier detection algorithms changes when the σ value increases. Due to the increase of the noise level in the dataset, most of the algorithms can classify the outlier data from normal data more accurately. However, for the SVM algorithm and with increasing the σ value, the behavior of outlier detection has been changed from a random prediction to a flow of prediction and classification (the SVM graph shape becomes smoother). This shows that the SVM cannot classify the outlier data with small amounts of noise, but regardless, this does not mean that the classification accuracy provided by SVM has gradually changed.

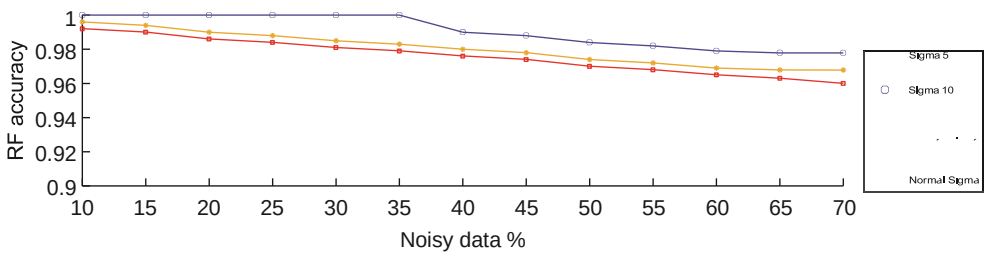

FIGURE 6 Accuracy of the RF outlier detection algorithm in a noisy environment when $\sigma = 0$, 5, and 10

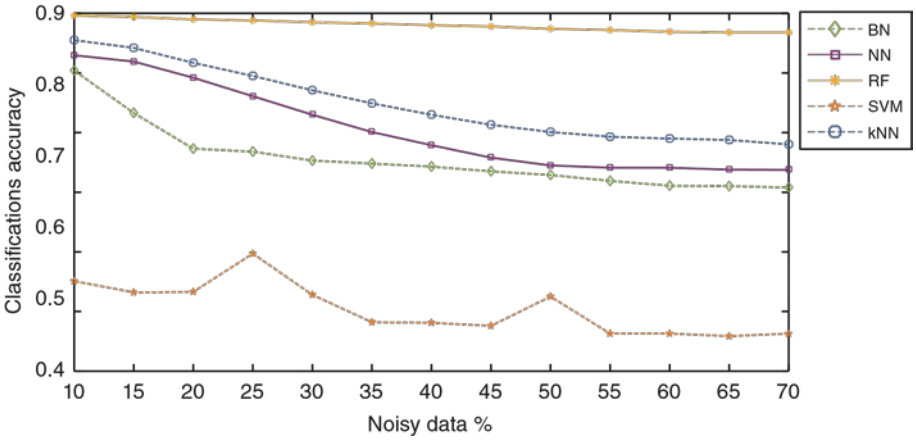

FIGURE 7 Accuracy of the five classification algorithms in a noisy environment when $\sigma = 5$

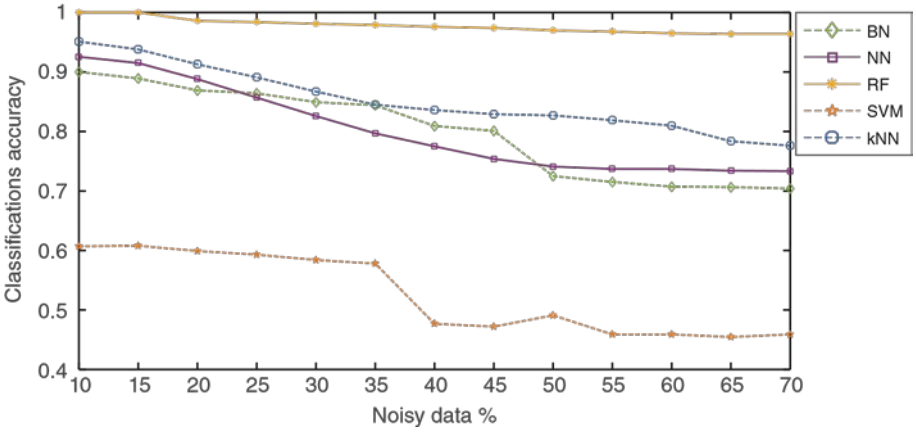

FIGURE 8 Accuracy of the five classification algorithms in a noisy environment when $\sigma = 7.5$

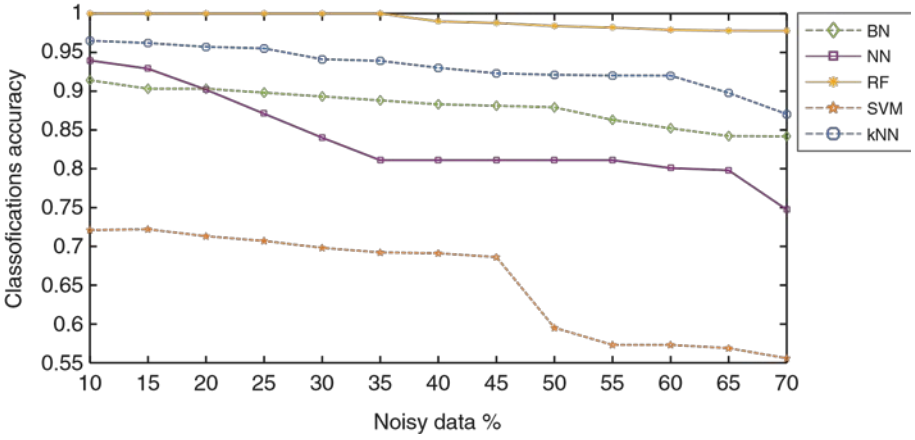

FIGURE 9 Accuracy of the five classification algorithms in a noisy environment when $\sigma = 10$

The RF accuracy increased and reached the maximum classification accuracy, which is 100% at certain points of the simulation rounds. The accuracy percentage increased rapidly when the value of σ increased

from 7.5 to 10. Indeed, the overall accuracy, when σ is equal to 10, is more than 99.7% for all percentages of noisy data.

For kNN and NB algorithms, the accuracy of outlier detection has been increased with the increase of the value of σ. The accuracy of NB, in particular, has increased very quickly compared to kNN, but overall, as shown in Figure 9, the accuracy of kNN is higher than that of NB. Concerning NN, in some parts of the simulation, it shows the same value and this is due to the problems of fitting or stack at epoch. But overall, the output analysis shows that the RF algorithm provides the highest outlier detection accuracy compared to the other classification algorithms.

Finally, identifying the importance of features that have been considered for the RF algorithm requires discussion in this section. Figure 10 shows the importance of the features based on two measurements: mean decrease accuracy (MDA) and mean decrease in Gini (MDG). The first one shows how much the accuracy will be reduced if we exclude each feature from the proposed algorithm. For instance, Fn has the highest impact on the algorithm accuracy, which means that without this feature the proposed algorithm can detect outliers but probably with a very low accuracy reaching less than 30%. The considered features are plotted in descending importance; a feature with a high accuracy means that considering this feature will lead to better outlier detection. The second measurement, MDG, depicts the impurity of features. It is used as a metric to divide data into smaller groups in the decision tree; therefore, the MDG shows how pure the nodes are at the end of the tree.

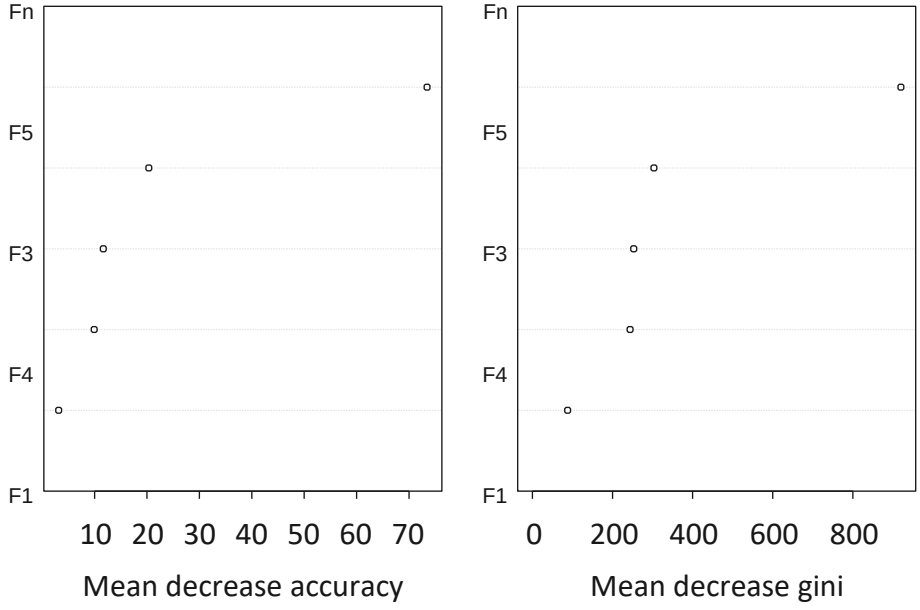

FIGURE 10 Comparison of features' importance for the global outlier detection algorithm

5 CONCLUSION

This article proposes a novel global outlier detection approach for WSNs. Our approach is based on time-series analysis, entropy technique, and random forest-based classification algorithm. This approach allows for the utilization of actual sensory data, as well as historical data and data collected from the best neighbor, in order to detect outliers. Experimental results obtained from a real and synthetic dataset have

proven the capabilities of our proposed detection approach to adapt its behavior to suit different dynamics and noise level scenarios, thus achieving a significant classification accuracy compared to existing nontime-series approaches. In future work, this approach can be enhanced by proposing effective solutions for the sensor nodes' detention problem in WSNs, which can prevent further negative effects on the decision-making process. In addition, we plan to consider different datasets to conduct more comprehensive experiments allowing to confirm the effectiveness of the proposed approach. Finally, an interesting perspective of the present article would be to investigate the impact of varying the number of features on the proposed approach performance.

ACKNOWLEDGMENT

The authors would like to thank Prince Sultan University for their support.

DATA AVAILABILITY STATEMENT

Data will be available upon request to the corresponding author.

ORCID

*Maha Driss*https://orcid.org/0000-0001-8236-8746

*Wadii Boulila*https://orcid.org/0000-0003-2133-0757

REFERENCES

1. Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. *Comput Netw*. 2008;52(12):2292-2330. arXiv:1011.1529. https://doi.org/ 10.1016/j.comnet.2008.04.002
2. Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. *Comput Netw*. 2002;38(4):393-422. https://doi. org/10.1016/S1389-1286(01)00302-4
3. D. M. Hawkins, *Identification of Outliers*, Springer, 1980. https://doi.org/10.1007/978-94-015-3994-4
4. Ord K. Outliers in statistical data. *Int J Forecast*. 1996;12(1):175-176. https://doi.org/10.1016/0169-2070(95)00625-7
5. Kandhari R, Chandola V, Banerjee A, Kumar V, Kandhari R. Anomaly detection. *ACM Comput Surv*. 2009;41(3):1-6. https://doi.org/10. 1145/1541880.1541882
6. Ding M, Chen D, Xing K, Cheng X. Localized fault-tolerant event boundary detection in sensor networks. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2005); 2 (C), 2005:902-913. https://doi.org/10.1109/ INFCOM.2005.1498320
7. Chen Q, Lam KY, Fan P. Comments on εdistributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networksε. *IEEE Trans Comput*. 2005;54(9):1182-1183. https://doi.org/10.1109/TC.2005.140
8. Martincic F, Schwiebert L. Distributed event detection in sensor networks. Proceedings of the International Conference on Systems and Networks Communications; 2006:43. https://doi.org/10.1109/ICSNC.2006.32
9. Zhang Y, Hamm NA, Meratnia N, Stein A, Van De Voort M, Havinga PJ. Statistics-based outlier detection for wireless sensor networks. *Int J Geograph Inf Sci*. 2012;26(8):1373-1392.

10. Chen J, Kher S, Somani A. Distributed fault detection of wireless sensor networks. Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks; 2006:65-72. https://doi.org/10.1145/1160972.1160985
11. Luo X, Dong M, Huang Y. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Trans Comput*. 2006;55(1):58-70. https://doi.org/10.1109/TC.2006.13
12. da Silva APR, Martins MHT, Rocha BPS, Loureiro AAF, Ruiz LB, Wong HC. Decentralized intrusion detection in wireless sensor networks. Proceedingsofthe1stACMInternationalWorkshoponQualityofService&SecurityinWirelessandMobileNetworks;2005:16-23.https:// doi.org/10.1145/1089761.1089765
13. Bhuse V, Gupta A. Anomaly intrusion detection in wireless sensor networks. *J High Speed Netw*. 2006;15(1/2006):33-51. https://doi.org/ 10.1007/978-3-540-77871-4_14
14. Ayadi A, Ghorbel O, Obeid AM, Abid M. Outlier detection approaches for wireless sensor networks: a survey. *Comput Netw*.
2017;129:319-333.
15. Subramaniam S, Palpanas T, Papadopoulos D, Kalogeraki V, Gunopulos D. Online outlier detection in sensor data using non-parametric models. Proceedings of the 32nd International Conference on Very Large Data Bases VLDB '06; 2006:187-198; http://www.vldb.org/conf/ 2006/p187-subramaniam.pdf
16. Gupta M, Gao J, Aggarwal CC, Han J. Outlier detection for temporal data: a survey. *IEEE Trans Knowl Data Eng*. 2013;26(9):2250-2267.
17. Safaei M, Ismail AS, Chizari H, et al. Standalone noise and anomaly detection in wireless sensor networks: a novel time-series and adaptive Bayesian-network-based approach. *Softw Pract Exp*. 2020;50(4):428-446.
18. MeratniaN,HavingaP.Outlierdetectiontechniquesforwirelesssensornetworks:asurvey.*IEEECommunSurvTutor*.2010;12(2):159-170. https://doi.org/10.1109/SURV.2010.021510.00088
19. Pang-Ning T, Steinbach M, Kumar V. Introduction to data mining. *Lib Congr*. 2006;796:51-56. https://doi.org/10.1016/00224405(81)90007-8
20. Han J, Kamber M. Data mining: concepts and techniques; Vol. 54, 2006; arXiv:arXiv:1011.1669v3. https://doi.org/10.1007/978-3-64219721-5
21. Safaei M, Asadi S, Driss M, et al. A systematic literature review on outlier detection in wireless sensor networks. *Symmetry*. 2020;12(3):328. 22. Vuran MC, Akan ÖB, Akyildiz IF. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Comput Netw*. 2004;45(3):245-259.
23. Breuniq M, Kriegel HP, Ng R, Sander J. LOF: identifying density-based local outliers, SIGMOD record (ACM Special Interest Group on Management of Data) arXiv:342009.335388. https://doi.org/10.1145/335191.335388
24. Saneja B, Rani R. An efficient approach for outlier detection in big sensor data of health care. *Int J Commun Syst*. 2017;30(17):e3352. https://doi.org/10.1002/dac.3352
25. Shahid N, Naqvi IH, Qaisar SB. One-class support vector machines: analysis of outlier detection for wireless sensor networks in harsh environments. *Artif Intell Rev*. 2013;43:515-563. https://doi.org/10.1007/s10462-013-9395-x

26. Rajasegarar S, Leckie C, Palaniswami M, Bezdek JC. Distributed anomaly detection in wireless sensor network. Proceedings of the 2006 IEEE Singapore international conference on communication systems, ICCS 2006; 2006. https://doi.org/10.1109/ICCS.2006.301508
27. Zhuang Y, Chen L. In-network outlier cleaning for data collection in sensor networks. Proceedings of the Workshop in VLDB.
28. Wu W, Cheng X, Ding M, Xing K, Liu F, Deng P. Localized outlying and boundary data detection in sensor networks. *IEEE Trans Knowl Data Eng*. 2007;19:1252-1261. https://doi.org/10.1109/TKDE.2007.1062
29. Guenterberg E, Ghasemzadeh H, Loseu V, Jafari R. Separating the wheat from the chaff: practical anomaly detection schemes in ecological applications of distributed sensor networks. Aspnes J, Scheideler C, Arora A, Madden S, eds. *Distributed Computing in Sensor Systems*; Berlin, Heidelberg: Springer; 2007. https://doi.org/10.1007/978-3-540-73090-3_15
30. Sheng B, Li Q, Mao W, Jin W. Outlier detection in sensor networks. Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing - MobiHoc '07; 2007:219-228. https://doi.org/10.1145/1288107.1288137
31. Abid A, Masmoudi A, Kachouri A, Mahfoudhi A. Outlier detection in wireless sensor networks based on OPTICS method for events and errors identification. *Wirel Personal Commun*. 2017;97(1):1503-1515. https://doi.org/10.1007/s11277-017-4583-7
32. Barakkath Nisha U, Uma Maheswari N, Venkatesh R, Yasir Abdullah R. Fuzzy-based flat anomaly diagnosis and relief measures in distributed wireless sensor network. *Int J Fuzzy Syst*. 2017;19:1528-1545. https://doi.org/10.1007/s40815-016-0253-2
33. Chebbi I, Boulila W, Farah IR. Improvement of satellite image classification: approach based on hadoop/mapreduce. Proceedings of the 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP); 2016:31-34; IEEE.
34. Chebbi I, Boulila W, Mellouli N, Lamolle M, Farah IR. A comparison of big remote sensing data processing with hadoop mapreduce and spark. Proceedings of the 2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP); 2018:1-4; IEEE.
35. Ghalem SK, Kechar B, Bounceur A, Euler R. A probabilistic multivariate copula-based technique for faulty node diagnosis in wireless sensor networks. *J Netw Comput Appl*. 2019;127(February 2018):9-25. https://doi.org/10.1016/j.jnca.2018.11.009
36. Qiao Y, Cui X, Jin P, Zhang W. Fast outlier detection for high-dimensional data of wireless sensor networks. *Int J Distrib Sens Netw*. 2020;16(10):1550147720963835.
37. Gupta U, Bhattacharjee V, Bishnu PS. Outlier detection in wireless sensor networks based on neighbourhood. *Wirel Personal Commun*. 2021;116(1):443-454.
38. Wang F, Li R, Wang H, Zhu H, Xiong N. Ts-Padm: anomaly detection model of wireless sensors based on spatial-temporal feature points. *Wirel Commun Mob Comput*. 2021;2021:6656498.
39. Carvalho C, Gomes DG, Agoulmine N, De Souza JN. Improving prediction accuracy for wsn data reduction by applying multivariate spatio-temporal correlation. *Sensors*. 2011;11(11):10010-10037.
40. Xie M, Hu J, Guo S. Segment-based anomaly detection with approximated sample covariance matrix in wireless sensor networks. *IEEE Trans Parallel Distrib Syst*. 2014;26(2):574-583.

41. Jiang L, Liu A, Hu Y, Chen Z. Lifetime maximization through dynamic ring-based routing scheme for correlated data collecting in WSNS. *Comput Electr Eng*. 2015;41:191-215.
42. Almeida FR, Brayner A, Rodrigues JJ, Maia JEB. Improving multidimensional wireless sensor network lifetime using Pearson correlation and fractal clustering. *Sensors*. 2017;17(6):1317.
43. Li Y. Anomaly detection in wireless sensor networks based on time factor. *J Intell Fuzzy Syst*. 2019;34(4):4639-4645.
44. Rajesh G, Chaturvedi A. Correlation analysis and statistical characterization of heterogeneous sensor data in environmental sensor networks. *Comput Netw*. 2019;164:106902.
45. Angiulli F, Basta S, Lodi S, Sartori C. Reducing distance computations for distance-based outliers. *Expert Syst Appl*. 2020;147:113215.
46. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5-32.
47. Janakiram D, Reddy V, Kumar AP. Outlier detection in wireless sensor networks using Bayesian belief networks. Proceedings of the 2006 1st International Conference on Communication Systems Software & Middleware; 2006:1-6.
48. Zhang Y, Meratnia N, Havinga PJ. Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine. *Ad Hoc Netw*. 2013;11(3):1062-1074.
49. Yang P, Zhu Q, Zhong X. Subtractive clustering based RBF neural network model for outlier detection. *JCP*. 2009;4(8):755-762.
50. Madden S Intel lab data. 2014. http://db.csail.mit.edu/labdata/labdata.html
51. Boulila W, Farah IR, Ettabaa KS, Solaiman B, Ghézala HB. Improving spatiotemporal change detection: a high level fusion approach for discovering uncertain knowledge from satellite image databases. *Icdm*. Vol 9. Citeseer; 2009:222-227.
52. Farouq MW, Boulila W, Abdel-Aal M, Hussain A, Salem A-B. A Novel Multi-Stage Fusion based Approach for Gene Expression Profiling in Non-Small Cell Lung Cancer. *IEEE Access*. 2019;7:37141–37150.