A Method for Fault Detection in Wireless Sensor Network Based on Pearson's Correlation Coefficient and Support Vector Machine Classification

Priyajit Biswas (priyajit.biswas@yahoo.com)
Indian Institute of Engineering Science and Technology https://orcid.org/0000-0002-6933-1858
Tuhina Samanta
Indian Institute of Engineering Science and Technology

Research Article

Keywords: Wireless Sensor Network, Machine Learning Algorithm, Correlation Coefficient, SVM, Measurement Fault

Posted Date: July 1st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-380070/v1

Version of Record: A version of this preprint was published at Wireless Personal Communications on November 1st, 2021. See the published version at https://doi.org/10.1007/s11277-021-09257-7.

Noname manuscript No.
(will be inserted by the editor)

A Method for Fault Detection in Wireless Sensor Network Based on Pearson's Correlation Coefficient and Support Vector Machine Classification

Priyajit Biswas · Tuhina Samanta

Received: date / Accepted: date

Abstract Sensor nodes are tiny low-cost devices, prone to various faults. So, it is imperative to detect those faults. This paper presents a sensor measurement fault detection algorithm based on Pearson's correlation coefficient and the Support Vector Machine(SVM) algorithm. As environmental phenomena are spatially and temporally correlated but faults are somewhat uncorrelated, Pearson's correlation coefficient is used to measure correlation. Then we used SVM to classify faulty readings from normal reading. After classification, faulty readings are discarded. We used network simulator NS-2.35 and Matlab for evaluation of our proposed method. We evaluated our fault detection algorithm using performance metrics, namely, Accuracy, Precision, Sensitivity, Specificity, Recall, F_1 Score, Geometric Mean(G_mean), Receiver Operating Characteristics (ROC), and Area Under Curve(AUC).

Keywords Wireless Sensor Network · Machine Learning Algorithm · Correlation Coefficient · SVM · Measurement Fault

1 Introduction

Wireless Sensor Network(WSN) consists of a large number of sensor nodes scattered over a large spacial region with one or more base stations(BS). The basic functionality of the WSN node is to collect information and send it to BS for analysis[9, 17]. These sensor nodes are tiny, low-cost devices equipped with

Department of Information Technology,
Indian Institute of Engineering Science and Technology, Shibpur,
Howrah, West Bengal, India 711103
E-mail: priyajit.biswas@yahoo.com

Department of Information Technology,
Indian Institute of Engineering Science and Technology, Shibpur,
Howrah, West Bengal, India 711103
E-mail: t_samanta@it.iiests.ac.in

one or more sensors. Sensor nodes are prone to various faults[7]. There are many factors involved that can cause a fault, such as hardware failure, software fault, communication error, etc.[19] These faults may persist for a long time or maybe instantaneous. Since faults are unavoidable, discovering faulty and fault-free nodes is crucial in the field of WSN[4]. The fault detection algorithm needs to be designed in such a way that it can detect such inconsistent behavior that can disrupt the normal functionality of the WSN. Sensors are deployed to monitor the region, and if any suspicious event occurs, report the same to the base station. Moreover, these cheap nodes are prone to various faults, including measurement fault. This implies that sensor measurements are not reliable. So the event can't be concluded based on a single measurement instance. Measurements need to be correlated spatially and temporally to decide whether a measurement is faulty, normal, or event. We know that environmental phenomena are spatially and temporally correlated [1]. In our work, we exploited this property to detect faulty reading. We used the Person's correlation coefficient[18] between two-time slot to detect the presence of a fault. Then we used SVM[2] to classify between faulty reading and normal reading.

Our contribution in this work is as follows,

- We used Pearson's correlation coefficient (ρ) to measure the correlation between two-time slots. ρ is invariant of scaling and self normalizing, which makes it suitable for spatially correlated environmental features.
- We used SVM to classify faulty and normal reading. SVM performs well in higher dimensional feature space. Moreover, using the kernel trick of SVM, nonlinear data can also be solved. As our dataset is nonlinear and has two features, namely humidity, and temperature, SVM is suitable for our method.
- NS2 and Matlab is used to simulate and analyze our proposed method, respectively. We also tested our method over a real-world dataset.

The rest of the paper is organized as follows: related works are discussed in section-2. Problem definition is given in Section-3. Preliminaries are described in section-4. Our proposed method is presented in section-5. Simulation results and performance evaluation are shown in section-6. The conclusion is drawn in section-7.

2 Related Works

In [1], authors proposed General anomaly detection(GAD), a fully distributed scheme for practical large scale networked industrial sensing systems(NISSs). Real-time detection, Distributed solution, and General solution are three properties of GAD. Their method, distributed matching-based grouping algorithm (DMGA), divides all sensing components into small, strongly correlated groups in a fully distributed way. Spatial correlation is a natural property in various physical phenomena. "Since physical phenomena are continuous, these spatial

correlations should be temporally correlated to previous mappings." They assumed measurement errors follow Gaussian distributions. They evaluated their method using a successful detection rate (SDR) and false-positive detection rate (FPDR).

In [19], authors used SVM with Gaussian kernel to detect faults in WSN. They classified Data faults as Offset fault, Gain fault, Stuck-at fault, Out of bounds. In their proposed method, received sensor data is classified using SVM with Gaussian kernel function. Their learning phase is performed in BS. Then, the decision function was transmitted to each cluster head to classify new measured data. In their proposed method, every time a new data is measured, an observation vector composed of last three data measurement of two sensors (V_t, V_{t-1}, V_{t-2}) is constructed by the data preparation block although SVM is capable of multidimensional classification. After that, this observation vector is classified by SVM using the decision function. If SVM output is positive new data is normal else data is faulty. Their labeled dataset is based on an existing dataset published by the researchers of the University of North Carolina at Greensboro[11].

A fault diagnosis protocol based on gradient descent and evolutionary approach was proposed by Swain et al. in [12]. Their method detects the faulty nodes and isolates them from the network. Their proposed protocol comprises four phases, namely clustering phase, communication phase, fault detection and classification phase, and isolation phase. They used a genetic algorithm and neural network for fault detection. Their method classifies faults into four types according to the rate of faults. Then faults are isolated. Their comparison showed that evolutionary approach outperforms gradient descent for their sensor data.

In [13], the authors proposed a fault diagnosis protocol that detects heterogeneous faults in WSN. Their protocol is capable of detecting soft permanent, hard permanent, transient, and intermittent faults, as claimed by authors. In their method, hard permanent faults are identified by a time out status register mechanism. Soft permanent, intermittent, and transient faults are detected by a statistical test, namely analysis of variance (ANOVA) test. In this phase, m number of measurements of n nodes in a cluster was used for the ANOVA test. This test was repeated for r times to detect faults. They also used the feed-forward probabilistic neural network (PNN) to classify these heterogeneous faults. However, as the protocol relies on the coordinator node, any inconsistent behavior such as node failure or erroneous results of the coordinate node will lead to the degradation of the performance of the protocol.

In [10], authors have proposed a density-based spatial clustering of applications with noise (DBSCAN) algorithm for detecting anomalies by evaluating three features from eight features of the Intel Berkeley Research lab(IRLB) dataset. The main idea behind the density-based approach is that a larger part of data is normal, and these data are stored in the cluster head with high density. DBSCAN detects low-density regions in the network and removes the anomaly. They choose two parameters of DBSCAN by performing the algorithm in different parameters and comparing their obtained results. After that,

they labeled data based on obtained results as normal and anomaly. Then they used these labeled data to train SVM classifier.

In [14], authors proposed a complete fault diagnosis methodology to detect faulty sensors in WSN. Their proposed method contains four phases, namely, initialization phase, fault detection phase, fault classification phase, and fault tolerance phase. They used checksum and Fletcher's checksum method to detect hard fault and link fault. They also used Mann-Whitney U statistical test for soft fault detection. Then they utilized Gaussian transformation function to classify the soft faults. They also used a stepwise regressional method to tolerate fault in their method. The Detection phase is performed on each cluster head with its own and member sensor measurements. In Mann–Whitney U statistical test, P-value decides the status of the sensor measurement value. They used a threshold value θ based on application and situation of the sensor network. If P-value is less than the threshold, their method declares it as soft fault.

In [16], the authors proposed DODS (Distributed Outlier Detection Scheme), where outliers are detected locally by each node. They used four data types, i.e., temperature, voltage, humidity, and light. The main idea is to clean sensed data (measurements) from outlier (incorrect data). The scheme operates in nodes that made the sensing operation and does not require any neighbor's communication. The solution exploits the temporal correlations existing in the sensed data (current and history sensed data) of the same node and its remaining energy level. Outlier detection is performed using Bayes' classifier for each type of data. Only nodes belong to an interesting region (IR) participate in the outlier detection process. To learn the prior probability and to compute all conditional probability, they used a supervised off-line method. They considered a different set of classes (small, medium, large) for different data types and used the maximum a posteriori (MAP) concept in order to determine optimal class. But their method does not differentiate between faults and events. No spatial correlation is considered in their method.

3 Problem Description

Let, N number of nodes are deployed in a region. Each node is equipped with k sensors to sense $k(k > 1)$ number of environmental features. These nodes periodically sense environmental features and send them to their associated Cluster Head (CH). Upon receiving sensed data, CH starts analyzing them to detect if there is any fault.

Suppose, sensor nodes periodically sense environmental data with time interval T. We divide total time into time slots t ($t = 0, 1, 2 \dots$) depending on T as illustrated in Fig. 1.

Let at time slot t sensed data is stored in CH in vector $Z_t = [Z_{1,t}, Z_{2,t}, \dots, Z_{m,t}]$, where m is member nodes of the cluster head. As sensor nodes are faulty, this Z_t is composed of both faulty and normal reading. CH detects these faults and eliminates them. In our work, CH computes Pearson's correla-

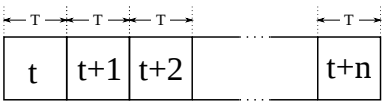

Fig. 1 Illustration of time slot.

tion coefficient(ρ) between Z_t and Z_{t-1} i.e., $\rho(Z_{t-1}, Z_t)$. Then, CH classifies $\rho(Z_{t-1}, Z_t)$ as faulty or normal using Support Vector Machine Classifier.

4 Preliminaries

Spatiotemporal correlation is the nature of various physical phenomena such as temperature, humidity, illumination, and many more. Spatial correlation implies a correlation mapping of measures between two neighboring nodes at time t. As physical phenomena are continuous, there should also be a mapping between current measure (t) and previous measure ($t-1$). This is called temporal correlation. Several possible correlation measures are there, among which Pearson's correlation coefficient is most popular.

Definition 1 The (Pearson) **correlation coefficient** between two random variable X and Y is defined as

$$\text{corr}[X, Y] = \frac{\text{Cov}[X, Y]}{\sqrt{\text{var}[X], \text{var}[Y]}} \tag{1}$$

where $\text{cov}[X, Y]$ is the covariance of X and Y. Whereas $\text{var}[X]$ is the variance of random variable X. The correlation coefficient can be viewed as a degree of linearity between X and Y. [18]

4.1 Support Vector Machine

Let a training sample set of length of length k is given with two separable classes P and N:

$$\{(X_k, y_k), k = 1, \ldots, K\}$$

where $y_k \in \{1, -1\}$ labels X_k to belong to either of the two classes. we want to find a hyper-plane in terms of weight vector(w) and bias term(b), that separates the two classes.

In case of linear classification, SVM classifier computes a decision hyperplane i.e. $\mathbf{x}^T\mathbf{w} + b = 0$ to separate the two classes, $P = \{(\mathbf{x}_i, 1)\}$ and $N = \{(\mathbf{x}_i, -1)\}$. This implies for both $\mathbf{x}_i \in P$ and $\mathbf{x}_i \in N$ has to satisfy (2).

$$y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 0 \tag{2}$$

From all possible hyperplanes, our objective is to find out the optimal hyperplane that satisfies the above condition. We need to place the optimal

hyperplane in a position such that distance from the hyperplane to the closest point of either side is the same. Now the problem of finding the optimal decision plane in terms of $\mathbf{w}$ and b can be formulated as:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\mathbf{w}^T\mathbf{w} = \frac{1}{2}||\mathbf{w}||^2 \\ &\text{subject to } y_i(\mathbf{x}_i^T\mathbf{w} + b) \geq 1 \end{aligned}$$

Its solution gives us the optimal margin classifier. The solution can be obtained using the Lagrange multipliers method, as shown in (3).

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = \sum_{i \in sv} \alpha_i y_i \mathbf{x}_i \tag{3}$$

where α_i is Lagrange multiplier and $\mathbf{x}_i$ is support vector.

Definition 2 The vectors $\mathbf{x}_i$ residing on either side of separation hyperplane for which $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$ holds are called support vector(sv). SVM only depends on the support vectors. Other sample vectors are not important. [2]

Substituting w from (2) we get,

$$y_i(\sum_{j \in sv} \alpha_j y_j \mathbf{x}_i^T \mathbf{x}_j + b) = 1$$

For the optimal weight vector $\mathbf{w}$ and optimal bias b, we have:

$$\begin{aligned} ||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w} &= \sum_{i \in sv} \alpha_i y_i \mathbf{x}_i^T \sum_{j \in sv} \alpha_j y_j \mathbf{x}_j \\ &= \sum_{i \in sv} \alpha_i y_i \sum_{j \in sv} \alpha_j y_j \mathbf{x}_i^T \mathbf{x}_j \\ &= \sum_{i \in sv} \alpha_i (1 - y_i b) = \sum_{i \in sv} \alpha_i - b \sum_{i \in sv} \alpha_i y_i \\ &= \sum_{i \in sv} \alpha_i \end{aligned} \tag{4}$$

4.1.1 SVM for nonlinear classification/Kernel Mapping

The method mentioned above is called linear SVM, which converges in case of linearly separable data. However, using kernel trick, SVM also works with nonlinear data set. With the help of Kernel trick, a sample x is mapped into a higher dimensional feature space where sample x is linearly separable as:

$$\mathbf{x} \longrightarrow \phi(\mathbf{x})$$

Moreover, the decision function can be rewritten for the new space as:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w} + b = \sum_{j=1}^{m} \alpha_j y_j (\phi(\mathbf{x})^T \phi(\mathbf{x}_j)) + b \tag{5}$$

where

$$\mathbf{w} = \sum_{j=1}^{m} \alpha_j y_j \phi(\mathbf{x}_j)$$

and b are the parameters of the decision plane in the new space.

Furthermore, the classification function in new space becomes:

$$y_i(\phi(\mathbf{x})^T \mathbf{w} + b) \geq 0 \tag{6}$$

From (5) and (6), we can see that vector x_j appears only in inner products of both decision function and learning law. This implies, we don't need to explicitly specify mapping function$\phi(X)$. We just need the inner product of the vectors in new space ϕ.

Definition 3 In Machine Learning, a kernel refers to kernel trick, which is used to solve a non-linear problem using a linear classifier. A kernel function takes $\mathbf{x}_i$ and $\mathbf{x}_j$ vectors as arguments and returns the inner product of their images $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$: [8]

$$K(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1)^T \phi(\mathbf{x}_2)$$

A kernel function only returns the inner product of two vectors. So the dimension of kernel space is not so important. Kernel $K(\mathbf{x}_1, \mathbf{x}_2)$ needs to be positive semidefinite to fulfill the criteria of Reproducing Kernel Hilbert Space(RKHS) where optimization problem has a finite-dimensional solution that converges to an optimal one.

Now by replacing $\mathbf{x}^T$ with $\phi(\mathbf{x})^T$ of (2), we obtain the new separation hyperplane in kernel space as shown in (7). We used this equation for classification of non linear problem.

$$y_i(\phi(\mathbf{x})^T \mathbf{w} + b) = y_i(\sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}, \mathbf{x}_j) + b) \geq 0 \tag{7}$$

The bias term (b) can be computed from any of the support vectors x_i as shown in (8).

$$b = y_i - \phi(\mathbf{x}_i)^T \mathbf{w} = y_i - \sum_{j=1}^{m} \alpha_j y_j (\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)) \tag{8}$$

$$= y_i - \sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

5 Proposed Scheme

In this work, we applied Pearson's correlation coefficient(ρ) and Support Vector Machine(SVM) to determine faults. Pearson's ρ is suitable for fault detection because of some of its intrinsic properties such as ρ is invariant to scaling i.e. $\rho(x, y) = \rho(x, ax + b)$, where a and b are constants. This implies normal reading, as well as event reading, will show a high correlation, whereas the presence of faulty reading will show a lesser correlation[6]. SVM is capable of classifying higher dimensional data, and using kernel trick SVM also can classify nonlinear data. As our data set in nonlinear and has two features, we used SVM classifier in this work. Fig. 2 shows our learning process(training and Classification).

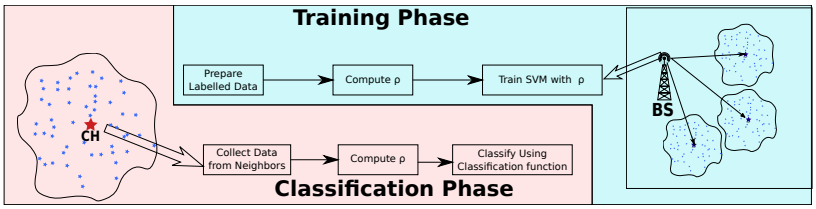

Fig. 2 Learning Process

5.1 Training Phase

We used Support Vector Machine(SVM) to classify between faulty reading and normal reading. SVM is trained at Base Station with labeled data $(Z_{i,t}, y)$, where $y \in \{-1, 1\}$. Here label $y = 1$ represents not faulty and $y = -1$ represents faulty class.

Our labeled data $Z_{i,t}$ is composed of humidity and temperature. At first, $Z_{i,t}$ is separated in two vectors ${}_H Z_{i,t}$ and ${}_T Z_{i,t}$ for humidity and temperature respectively. Then two new vector is constructed as ${}_H Z_t = ({}_H Z_{1,t}, {}_H Z_{2,t}, \ldots, {}_H Z_{m,t})$ and ${}_T Z_t = ({}_T Z_{1,t}, {}_T Z_{2,t}, \ldots, {}_T Z_{m,t})$. These are done in data preparation phase.

Then Pearson's correlation coefficient(ρ) is computed between two time slot $t-1$ and t for all ${}_H Z_t$ and ${}_T Z_t$ as ρ_H and ρ_T respectively. After that vector ρ is set as ρ= [ρ_H,ρ_T]. Then SVM is trained with (ρ, y) using procedure given in Section (4.1). Then w and b of classification hyperplane is send to all cluster head nodes.

In our method, we used Gaussian kernel function(K) as (10).

$$K(x, x_j) = \exp \frac{-||x - x_j||}{2\sigma^2} \tag{10}$$

where x and x_j are two vectors and σ is a free parameter.

5.2 Classification Phase

At each time slot(t), every sensor node collects environmental phenomena(e.g., humidity and temperature) and sends them to its associated cluster-head. Cluster head(CH) collects the reading of the neighboring node. CH computes the correlation between time-slot t and $t-1$ of each feature of all the neighboring nodes.

At each time slot, Cluster Head (CH) stores measurements of its member nodes in a vector Z_t. Then CH computes correlation coefficient ρ between Z_t and Z_{t-1} using (1). After that, decision function (5) is applied to this ρ. If ρ belongs to positive class then Z_t is not faulty otherwise Z_t is faulty.

We presented our fault detection algorithm in Algorithm 1. As described in the algorithm, like any other supervised learning, our method is also divided into two phases. First, the training phase, which is performed at BS using training data. The training phase produces a decision function in terms of w and b. Second, Classification phase, which is performed at each CH using the decision function.

Algorithm 1: Training and classification.

```
1  Function TrainiingPhase() is
2  |  /* On Base-station.                                          */
3  |  Compute ρ between z_t and z_{t-1} using (1);
4  |  Compute w and b using (5) and (8);
5  |  Send w and b to all CH node;
6  end
7  Function ClassificationPhase() is
8  |  /* On each CH.                                               */
9  |  foreach Timeslot T(T=1,2,3, ...) do
10 |  |  foreach Member Node m(m=1,2,3, ...) do
11 |  |  |  Collect data from each member node;
12 |  |  |  Z_t ← [Z_{1,t}, Z_{2,t}, ..., Z_{m,t}];
13 |  |  end
14 |  |  Compute ρ between z_t and z_{t-1} using (1);
15 |  |  Classify ρ using (7);
16 |  end
17 end
```

6 Simulation Scenario and Performance Evaluation

To evaluate our proposed method, we used NS2.35[3] and Matlab. NS2 is used to simulate our network scenario and generate measurement data. Then, this generated data is analyzed using Matlab. Performance evaluation is also done using Matlab.

We generated our simulation data using NS2.35. Here we used Gaussian distribution(N) with mean(μ) and variance(σ) of normal and event reading of

the dataset[11]. This dataset contains only measurements of 4 sensor nodes. However, actual WSN consists of hundreds of sensor nodes. In our simulation for simplicity, we simulated for only one cluster head with 100 member nodes. We also showed results for actual dataset by computing temporal correlation, as shown in Section 6.3.

In our network scenario, 100 sensors are randomly deployed in a $300m \times 300m$ with CH at the center. The transmission range of each node is $60m$.

Normal sensor readings for temperature are drawn from $N(\mu_{1t}, \sigma_{1t}^2)$ and event readings from $N(\mu_{2t}, \sigma_{2t}^2)$. Where $\mu_{1t} = 28.1273, \mu_{2t} = 29.3112$ and $\sigma_{1t} = 1.0952, \sigma_{2t} = 4.5588$.

Normal sensor readings for humidity are drawn from $N(\mu_{1h}, \sigma_{1h}^2)$ and event readings from $N(\mu_{2h}, \sigma_{2h}^2)$. Where $\mu_{1h} = 59.6504, \mu_{2h} = 78.4943$ and $\sigma_{1h} = 9.7391, \sigma_{2h} = 11.3831$.

All the faulty readings for temperature is also drawn from $N(\mu_{2t}, \sigma_{2t}^2)$. Where $\mu_{2t} = 29.3112$ and $\sigma_{2t} = 4.5588$. Moreover all the faulty readings for humidity is also drawn from $N(\mu_{2h}, \sigma_{2h}^2)$. Where $\mu_{2h} = 78.4943$ and $\sigma_{2h} = 11.3831$.

In our simulation, for analyzing our proposed method, the first 20% data are drawn from normal reading, and the last 20% data are drawn from event reading. Rest is a mix of normal and fault reading. Fault readings are mixed according to fault percentage. For example, in the case of 10% fault, 10% data is faulty, and the rest 90% data is normal. Simulation parameters are given in Table 1.

Table 1 Simulation Parameters

Parameters	Values
Network size	300 m × 300 m
Number of nodes	100
Mac protocol	802.15.4
Routing Protocol	AODV
Transport Protocol	UDP
Node's transmission range	60 m
Cluster Head location	(150,150)m
Simulation time	06h:30m:50s
Time slot width	5 sec
Total time slot	4690
Normal sensor reading (Temperature)	$N(28.1273, 1.0952)$
Normal sensor reading (Humidity)	$N(59.6504, 9.7391)$
Event sensor reading (Temperature)	$N(29.3112, 4.5588)$
Event sensor reading (Humidity)	$N(78.4943, 11.3831)$
Faulty sensor reading (Temperature)	$N(29.3112, 4.5588)$
Faulty sensor reading (Humidity)	$N(78.4943, 11.3831)$
Percentage of measurement faulty nodes	50%, 40%, 30%, 20%, 10%, and 5%

6.1 Performance Evaluation

6.1.1 Confusion Matrix

In the case of binary classification, only four possible outcomes may occur. They are, Positive samples tested as positive (TP), Negative sample tested as negative (TN), Positive sample tested as negative (FN) and, Negative sample tested as positive (FP). We can describe these four outcomes in the form of a confusion matrix[15], as shown in Table 2.

Table 2 Confusion Matrix

		Predicted	
		True	False
Actual	True	True Positive (TP)	False Negative (FN) (Type II error)
	False	False Positive (FP) (Type I error)	True Negative (TN)

To evaluate our proposed method, we used the following performance matrices[5]. These matrices are derived from the confusion matrix.

Accuracy Measure: "Accuracy is the ratio of correctly predicted observation to the total observation."

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{11}$$

Precision: "Precision is the ratio of the number of true positives to the number of true positives plus the number of false positives."

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

Recall: "Recall is the ratio of the number of true positives to the number of true positives plus the number of false negatives."

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

F_1 **Score**: "F_1 Score is a weighted harmonic mean of *Precision* and *Recall*."

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

Gmean: "Geometric mean (Gmean) is the square root of true positive rate and true negative rate."

$$Gmean = \sqrt{TPR \times TNR} = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{FP + TN}} \tag{15}$$

Sensitivity: "Sensitivity is the ratio between the total number of positive samples to the number of samples tested as positive in the test."

$$Sensitivity = \frac{TP}{TP + FN} \tag{16}$$

Specificity: "Sensitivity is the ratio between the total number of negative samples to the number of samples tested as negative in the test."

$$Specificity = \frac{TN}{FP + TN} \tag{17}$$

Accuracy, Precision, Sensitivity, Specificity, Recall, F_1, Score, G_mean for faults 50%, 40%, 30%, 20%, 10%, and 5% is shown in Table 3. From the results presented in Table 3, it can be seen clearly that with increasing fault percentages, performances increases. This is because, with increasing fault percentages, the correlation between faulty and normal date decreases. However, the normal reading correlation remains high, making classification hyperplane more and more precise.

Table 3 Performance Evaluation

	Fault Percentage					
Metrics	50%	40%	30%	20%	10%	5%
Accuracy	0.9981	0.9968	0.9927	0.9833	0.9377	0.8096
Precision	0.9991	0.9982	0.9951	0.9881	0.9509	0.8069
Sensitivity	0.9978	0.9965	0.9928	0.9844	0.9460	0.8071
Specificity	0.9986	0.9971	0.9925	0.9813	0.9241	0.7953
Recall	0.9978	0.9965	0.9928	0.9844	0.9460	0.8071
F_1 Score	0.9984	0.9973	0.9940	0.9863	0.9484	0.8053
G mean	0.9982	0.9968	0.9927	0.9829	0.9349	0.7975
AUC	0.9996	0.9996	0.9993	0.9979	0.9823	0.8575

ROC Analysis: Receiver Operating Characteristics (ROC) analysis studies the sensitivity and the Specificity of the classifier. A ROC curve is a plot in which x-axis is the Specificity, and y-axis is the sensitivity of the classifier.

Area Under curve: The total area under the ROC curve is abbreviated as AUC.

ROC curves are generally used to evaluate the performance of various machine learning algorithms as it gives a comprehensive and visual method of summering the accuracy of an algorithm[5]. Therefore, in this paper, we used ROC and AUC analysis as one of the performance metrics to evaluate our proposed method. From the ROC curve, AUC is calculated by calculating the size of the area under curve. The higher the area, the Better the performance of the method. Fig. 3 shows ROC for various fault percentages. AUC is given in Table 3.

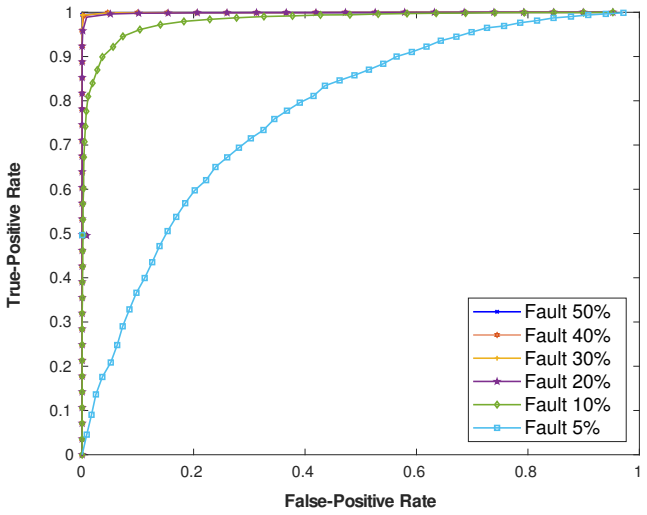

Fig. 3 ROC Curve

6.2 Comparison with existing work

We also compared our results with the existing work of Zidi et al. [19]. Fig. 4(a) shows comparative results of Detection Accuracy for our method and the method of Zidi et al. for fault percentages 1 to 5 (5% to 50% already shown in Table 3). Fig. 4(a) shows that our method works better with higher fault percentages. Our approach is based on the correlation coefficient. As we know, if the fault percentages increase, the correlation coefficient among data measurements decreases. For this reason, our method works better in high fault percentages.

Fig. 4(b) shows comparative results of Average False Positive Rate (FPR) for our method and the method of Zidi et al. Fig. 4(b) shows that our method shows 14% improvement in terms of Average False Positive Rate compared to Zidi et al. Correlation among measurements of non-faulty node is high. So, a non-faulty node measurement detected as faulty is less likely to happen. That is why our method has better FPR.

6.3 Performance Evaluation For dataset

We also evaluated our proposed method using a real-world environmental dataset of [11]. From this dataset, we took data of temperature and humidity for the indoor sensor node of the multi-hop scenario with anomalies. In this dataset, the anomaly is achieved on a sensor node by a hot water kettle, which increases the temperature and the humidity simultaneously.

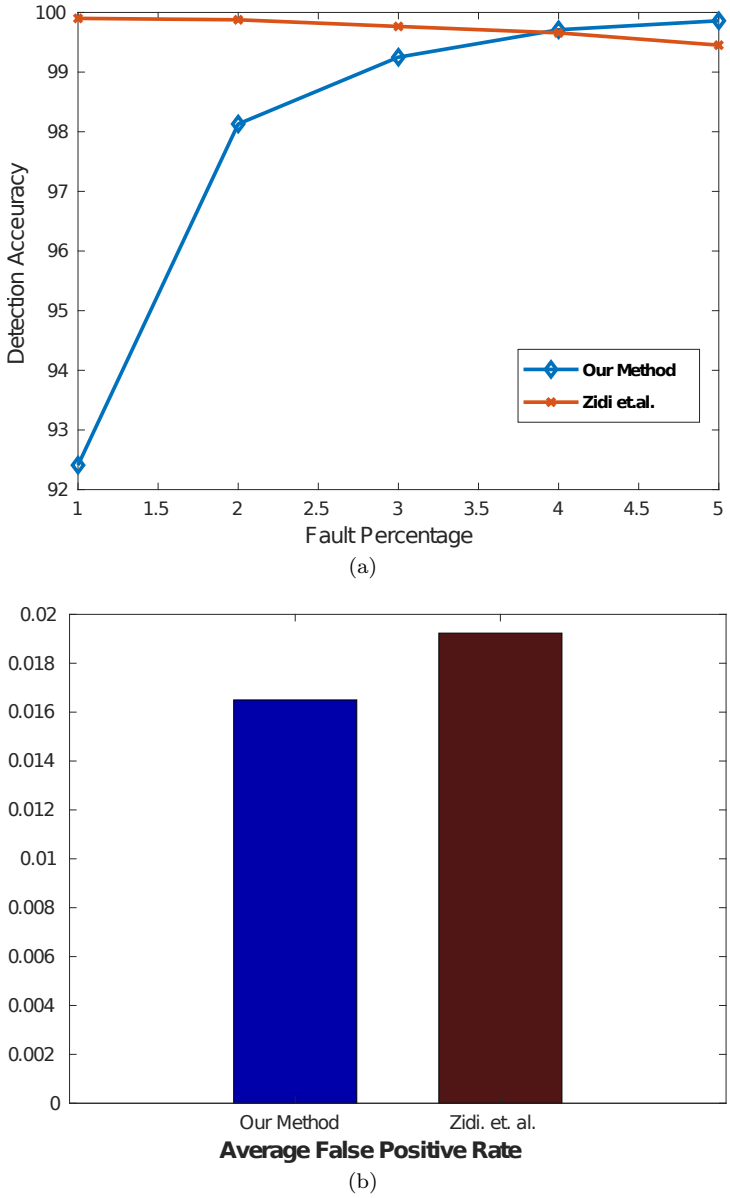

Fig. 4 Detection Accuracy and False Positive Rate

Accuracy, Precision, Sensitivity, Specificity, Recall, F_1, Score, G_mean for faults 40%, 30%, and 20% is shown in Table 4.

AUC and ROC are given in Fig. 5.

Table 4 Performance Evaluation

	Fault Percentage		
Metrics	40%	30%	20%
Accuracy	0.9412	0.9412	0.8824
Precision	1	1	0.7500
Sensitivity	0.8750	0.8333	0.7500
Specificity	1	1	0.9231
Recall	0.8750	0.8333	0.7500
F_1 Score	0.9333	0.9091	0.7500
G mean	0.9354	0.9129	0.8321

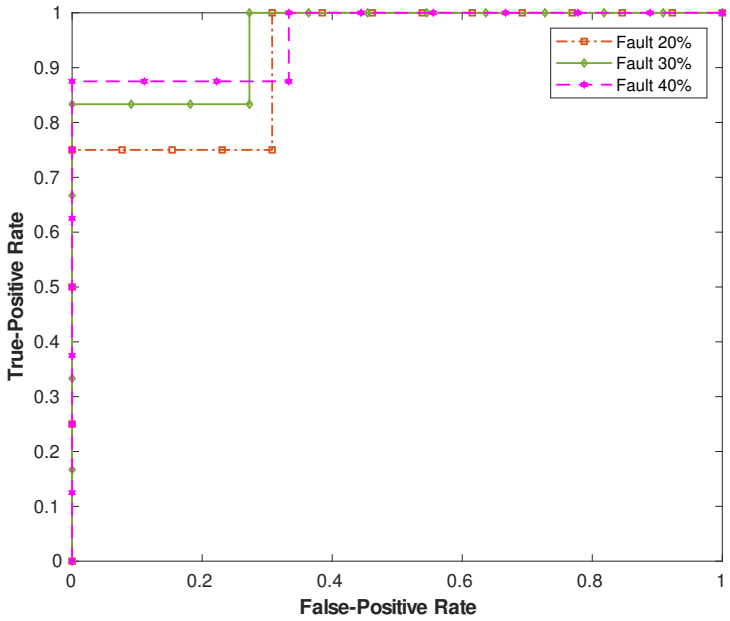

Fig. 5 ROC Curve

7 Conclusion

In this work, we have presented a fault detection algorithm based on Pearson's correlation coefficient and Support Vector Machine classification. Simulation results show that our method works better in high fault percentages. This is because, with increasing fault percentage, correlation decreases. We also evaluated our proposed method using a dataset[11]. Simulation results, as well as results from real-world data, show that our proposed method successfully classifies fault and normal sensor reading. A possible future directive could be working with more than one machine learning algorithm to detect faults. Applicability of the machine learning methods over some complex networks and data with more event parameters may also be studied in the future.

8 Declarations

Not applicable.

References

1. Chen PY, Yang S, McCann JA (2015) Distributed real-time anomaly detection in networked industrial sensing systems. IEEE Transactions on Industrial Electronics 62(6):3832–3842
2. Cortes C, Vapnik V (1995) Support-vector networks. Machine learning 20(3):273–297
3. Fall K, Varadhan K, et al. (2005) The ns manual (formerly ns notes and documentation). The VINT project 47:19–231
4. Guesmi H, Ben Salem S, Bacha K (2015) Smart wireless sensor networks for online faults diagnosis in induction machine. Computers & Electrical Engineering 41:226 – 239, DOI https://doi.org/10.1016/j.compeleceng.2014.10.015, URL http://www.sciencedirect.com/science/article/pii/S0045790614002614
5. Japkowicz N, Shah M (2011) Evaluating learning algorithms: a classification perspective. Cambridge University Press
6. Krishnamachari B, Iyengar S (2004) Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. IEEE Transactions on Computers 53(3):241–250
7. Muhammed T, Shaikh RA (2017) An analysis of fault detection strategies in wireless sensor networks. Journal of Network and Computer Applications 78:267 – 287, DOI https://doi.org/10.1016/j.jnca.2016.10.019, URL http://www.sciencedirect.com/science/article/pii/S1084804516302545
8. Ng A (2000) Cs229 lecture notes. CS229 Lecture notes 1(1):1–3
9. Priya KCK, Terence S (2013) Retp: Reliable event transmission protocol in a wireless sensor network. In: 2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), pp 181–188, DOI 10.1109/ICE-CCN.2013.6528489
10. Saeedi Emadi H, Mazinani SM (2018) A novel anomaly detection algorithm using dbscan and svm in wireless sensor networks. Wireless Personal Communications 98(2):2025–2035, DOI 10.1007/s11277-017-4961-1, URL https://doi.org/10.1007/s11277-017-4961-1
11. Suthaharan S, Alzahrani M, Rajasegarar S, Leckie C, Palaniswami M (2010) Labelled data collection for anomaly detection in wireless sensor networks. In: 2010 Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, pp 269–274, DOI 10.1109/ISSNIP.2010.5706782
12. Swain RR, Khilar PM (2017) Composite fault diagnosis in wireless sensor networks using neural networks. Wireless Personal Communications

95(3):2507–2548, DOI 10.1007/s11277-016-3931-3, URL https://doi.org/10.1007/s11277-016-3931-3

13. Swain RR, Khilar PM, Bhoi SK (2018) Heterogeneous fault diagnosis for wireless sensor networks. Ad Hoc Networks 69:15 – 37, DOI https://doi.org/10.1016/j.adhoc.2017.10.012, URL http://www.sciencedirect.com/science/article/pii/S1570870517301841
14. Swain RR, Dash T, Khilar PM (2019) A complete diagnosis of faulty sensor modules in a wireless sensor network. Ad Hoc Networks 93:101924, DOI https://doi.org/10.1016/j.adhoc.2019.101924, URL http://www.sciencedirect.com/science/article/pii/S1570870518309211
15. Ting KM (2010) Confusion Matrix, Springer US, Boston, MA, pp 209–209. DOI 10.1007/978-0-387-30164-8_157, URL https://doi.org/10.1007/978-0-387-30164-8_157
16. Titouna C, Naït-Abdesselam F, Khokhar A (2019) Dods: A distributed outlier detection scheme for wireless sensor networks. Computer Networks 161:93 – 101, DOI https://doi.org/10.1016/j.comnet.2019.06.014, URL http://www.sciencedirect.com/science/article/pii/S138912861930012X
17. Ul Islam R, Hossain MS, Andersson K (2018) A novel anomaly detection algorithm for sensor data under uncertainty. Soft Computing 22(5):1623–1639, DOI 10.1007/s00500-016-2425-2, URL https://doi.org/10.1007/s00500-016-2425-2
18. Wright S (1921) Correlation and causation. J agric Res 20:557–580
19. Zidi S, Moulahi T, Alaya B (2018) Fault detection in wireless sensor networks through svm classifier. IEEE Sensors Journal 18(1):340–347, DOI 10.1109/JSEN.2017.2771226