Received: 22 May 2019 | Revised: 24 June 2020 | Accepted: 10 July 2020

DOI: 10.4218/etrij.2019-0207

ORIGINAL ARTICLE

ETRI Journal WILEY

A precise sensor fault detection technique using statistical techniques for wireless body area networks

Smrithy Girijakumari Sreekantan Nair | **Ramadoss Balakrishnan**

Department of Computer Applications, National Institute of Technology, Tiruchirappalli, India

Correspondence
Smrithy Girijakumari Sreekantan Nair, Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India.
Email: smrithygs1990@gmail.com

Funding information
This work was supported by the Department of Electronics & Information Technology (DeitY), which is a division of Ministry of Communications and IT of the Government of India, under the Visvesvaraya PhD scheme for Electronics & IT.

One of the major challenges in wireless body area networks (WBANs) is sensor fault detection. This paper reports a method for the precise identification of faulty sensors, which should help users identify true medical conditions and reduce the rate of false alarms, thereby improving the quality of services offered by WBANs. The proposed sensor fault detection (SFD) algorithm is based on Pearson correlation coefficients and simple statistical methods. The proposed method identifies strongly correlated parameters using Pearson correlation coefficients, and the proposed SFD algorithm detects faulty sensors. We validated the proposed SFD algorithm using two datasets from the Multiparameter Intelligent Monitoring in Intensive Care database and compared the results to those of existing methods. The time complexity of the proposed algorithm was also compared to that of existing methods. The proposed algorithm achieved high detection rates and low false alarm rates with accuracies of 97.23% and 93.99% for Dataset 1 and Dataset 2, respectively.

KEYWORDS
correlation methods, fault diagnosis, sensors, statistical analysis, wireless sensor networks

1 | INTRODUCTION

Wireless body area networks (WBANs) enable continuous monitoring through sensors in both medical (remote patient monitoring, rehabilitation, assisted living, telemedicine, and biofeedback) and non-medical (sports, military, lifestyle, and entertainment) applications [1]. In healthcare scenarios, WBANs can effectively improve the quality of treatment offered through the continuous monitoring of physiological parameters instead of requiring routine checkups. Some examples of physiological parameters are temperature, pulse, photoplethysmogram, galvanic skin response, blood pressure, electrocardiogram, electromyogram, heart rate, electroencephalogram, and blood glucose level. SHIMMER [2], LOTUS [3], IRIS [4], MicaZ [5], TinyNode [6], Sun SPOT [7], Cricket [8], and TelosB [9] are commonly used nodes in the medical field. The robustness of the WBAN healthcare system heavily relies on the accuracy of sensor data. Unexpected failures in hardware, defective sensors, and maliciously inserted data can result in inaccurate sensor readings. Therefore, sensor fault detection is one of the key challenges in healthcare systems. Various techniques have been proposed to detect sensor anomalies. Hill and Minsker [10] proposed a data-driven model for detecting anomalies in environmental sensor data streams. Their approach only focuses on the detection of point anomalies and does not consider sensor correlation, resulting in a high false alarm rate. Liu and others [11] proposed sensor anomaly detection based on the Mahalanobis distance technique. Their method calculates the Mahalanobis distance between actual and forecasted sensor observations. If a calculated Mahalanobis distance is above a certain threshold, then anomalous sensor values are identified. The major drawback of this method is that it assumes neighboring nodes gather the same type of

data. Salem and others [12] proposed an anomaly detection technique using support vector machines (SVMs) and linear regression models in WBANs. However, this method is inefficient in terms of data updating and prediction. Haque and others [13] proposed a sensor anomaly detection technique to reduce false alarms in healthcare systems.

Their technique uses the sequential minimal optimization regression (SMOR) algorithm to predict sensor values based on historical data. The predicted values are then compared to the observed values for a specific duration, and the estimated difference is compared to a threshold to determine whether the sensor values are anomalous. The major drawback of this method is that equal weights are assigned to all sensors (ie, it does not consider correlation among sensors). Saneja and Rani [14] proposed an outlier detection technique for big sensor data in the healthcare field. However, their method has high complexity and limited accuracy. Al Rasyid [15] proposed an anomalous data detection method for WBANs using Gaussian regression and majority voting. However, this method incurs high computational complexity, a large number of required training data samples for modeling, and a high false alarm rate. Nagdeo and Mahapatro [16] proposed an anomaly detection method for WBANs using artificial neural networks (ANNs) and linear regression. This method has high computational complexity based on the use of a back-propagation ANN, as well as a low detection rate and high false alarm. Nezhad and Eshghi [17] used a combination of decision trees and linear regression to detect anomalies in healthcare scenarios. However, their method fails to use a sliding window approach, resulting in poor effectiveness for detecting anomalies in dynamic systems, such as healthcare systems. Boudargham and others [18] presented an anomaly detection technique using the modified cumulative sum (MCUSUM) method for WBANs. The major drawback of this method is a high false alarm rate caused by a lack of consideration for weighted correlations among various sensors.

Table 1 provides a brief summary of various methods described above. To overcome the challenges discussed above, we propose a sensor fault detection algorithm using simple statistical techniques. Our method does not make use of any prediction techniques, which significantly reduces its computational complexity.

The main contributions of this paper can be summarized as follows:

1. We propose a precise sensor fault detection technique using simple statistical techniques in WBANs.
2. The proposed scheme uses Pearson correlation to identify strongly correlated parameters.
3. We validate the proposed work using two datasets, namely MIMIC DB dataset 221 (Dataset 1) and MIMIC DB dataset 276 (Dataset 2), from the MIMIC database [19]. The results are compared to those of previous methods.

The remainder of this paper is organized as follows. Section 2 presents the proposed sensor fault detection method. Section 3 presents performance analysis results. Section 4 concludes this paper.

TABLE 1 Brief summary of various methods

Method	Technique	Type	Remarks
[11]	Mahalanobis distance	Distance based	• Infeasible in real-time healthcare scenarios
[12]	SVM and linear regression	Classification/ regression based	• Low accuracy • Infeasible in practical scenarios
[13]	SMOR	Regression based	• High false alarm rate caused by the absence of weighted correlations
[14]	Dynamic SMOR	Regression based	• High false alarm rate • High computational complexity
[15]	Gaussian regression	Regression based	• High computation complexity • Large number of training data samples required for modeling • High false alarm
[16]	ANN and linear regression	Classification/ regression based	• High computational complexity based on the use of a back-propagation ANN • Low detection rate • High false alarm rate
[17]	Linear regression and J48	Regression based/ decision tree	• Fails to use a sliding window mechanism
[18]	MCUSUM	Statistical based	• High false alarm rate caused by the absence of weighted correlations
Proposed method	Pearson correlation and statistical techniques	Statistical based	• High detection rate • Low false alarm rate • Low computation complexity • Feasible in real-time healthcare scenarios • Considers weighted correlations

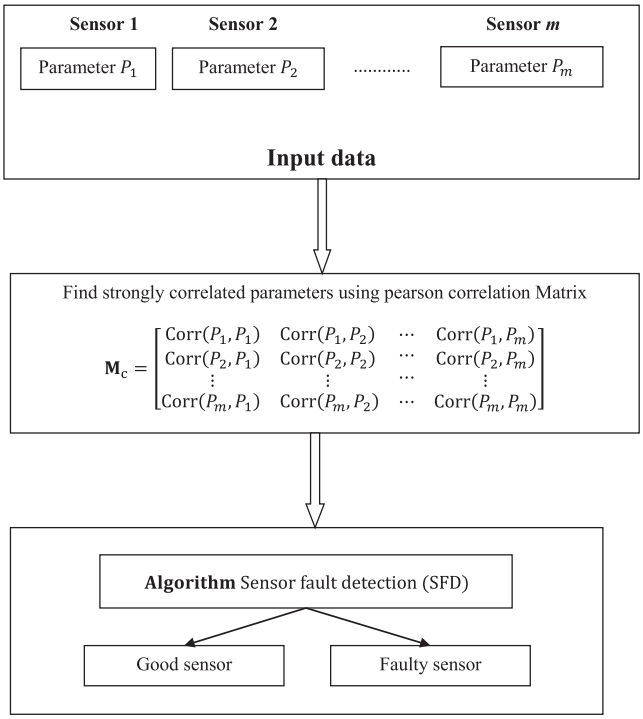

FIGURE 1 Proposed sensor fault detection method

2 | PROPOSED SENSOR FAULT DETECTION METHOD

Figure 1 presents the proposed sensor fault detection method. This framework consists of two stages:

1. Correlation Estimation.
2. Sensor Fault Detection (SFD).

2.1 | Correlation estimation

The main goal of the correlation estimation stage is to find strongly correlated parameters in the given set of parameters using Pearson correlation. A Pearson correlation coefficient is a value between -1 and 1 that represents the degree to which two variables are linearly related.

Correlation values are calculated among different parameters and defined in the form of a matrix. For m sensors, an $m \times m$ matrix is created. The correlation matrix $\mathbf{M}_c$ can be represented as follows:

$$\mathbf{M}_c = \begin{bmatrix} \mathrm{Corr}\left(P_1,P_1\right) & \mathrm{Corr}\left(P_1,P_2\right) & \cdots & \mathrm{Corr}\left(P_1,P_m\right) \\ \mathrm{Corr}\left(P_2,P_1\right) & \mathrm{Corr}\left(P_2,P_2\right) & \cdots & \mathrm{Corr}\left(P_2,P_m\right) \\ \vdots & \vdots & \cdots & \vdots \\ \mathrm{Corr}\left(P_m,P_1\right) & \mathrm{Corr}\left(P_m,P_2\right) & \cdots & \mathrm{Corr}\left(P_m,P_m\right) \end{bmatrix}, \quad (1)$$

where $\mathrm{Corr}\left(P_i,P_j\right), \{i=1,2,\ldots,m \text{ and } j=1,2,\ldots,m\}$ is the Pearson correlation coefficient between two parameters P_i and P_j. This Pearson correlation coefficient can be calculated as follows:

$$\mathrm{Corr}(P_i,P_j) = \frac{m\left(\sum P_iP_j\right) - \left(\sum P_i\right)\left(\sum P_j\right)}{\sqrt{\left(m\sum P_i^2 - \left(\sum P_i\right)^2\right)\left(m\sum P_j^2 - \left(\sum P_j\right)^2\right)}} \quad (2)$$
$$i=1,2,\ldots,m \text{ and } j=1,2,\ldots,m.$$

Algorithm Sensor Fault Detection (SFD)

Input:

N: Total Number of correlated sensors for particular sensor, S_{as}

W_{as}: Window of sensor S_{as} at time t

μ_{as}, σ_{as}: Mean and Standard deviation for the normal range of Parameter of S_{as} respectively

W_i: Window of correlated parameter i for sensor S_i at time t, $(i = 1, 2, \ldots, N)$

μ_i, σ_i: Mean and Standard deviation for the normal range of Correlated Parameters respectively $(i = 1, 2, \ldots, N)$

TH: Threshold for classifying S_{as}

Pos: Positive Count, Initially $\mathrm{Pos} = 0$

Output:

Whether sensor S_{as} is faulty or not

```
Begin
    mean_as = findmean(W_as)
    diff_as = |μ_as − mean_as|
    n_as = ⌈diff_as / σ_as⌉
    For(i = 1 to N)
        mean_i = findmean(W_i)
        diff_i = |μ_i − mean_i|
        n_i = ⌈diff_i / σ_i⌉
        If(n_as == n_i)
            Pos = Pos + 1
        End If
    End For
        // Check for sensor S_as fault
    If(Pos ≥ TH)
        Sensor S_as is good condition (ie, W_as is true medical condition)
    Else
        Sensor S_as is faulty (ie, W_as is not a true medical condition)
    End If
End
```

2.2 | SFD

The proposed SFD algorithm classifies a sensor S_{as} as either "good condition" or "faulty condition." It uses statistical measures such as means and standard deviations to obtain useful information via simple arithmetic. Means and standard deviations can be calculated as follows:

$$\text{Mean}\,(\mu) = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{3}$$

$$\text{Standard deviation}\,(\sigma) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i - \mu^2}, \tag{4}$$

where n is the total number of observations and x_i is the ith observation.

As inputs, the SFD algorithm takes the numbers of correlated sensors for a particular sensor from the correlation estimation phase, mean and standard deviation of the normal range of each physiological parameter, a threshold, and counter. Consider the following scenario. Suppose there are two sensors S_1 and S_2 measuring blood pressure and pulse, respectively. The normal ranges of blood pressure and pulse are 90/60 mm Hg to 120/80 mm Hg and 60 bpm–100 bpm, respectively. Suppose the values of S_1 and S_2 deviate from the normal range. If we consider S_1 and S_2 independently, then both values appear to be anomalous. However, when we consider correlations, this deviation may represent a real medical condition because when blood pressure increases, pulse also increases because these two physiological parameters are highly correlated. Otherwise, this deviation represents a sensor flaw.

The proposed algorithm operates as follows. We consider a consistent sliding window size for all physiological parameters. The proposed algorithm first determines how much the current window of S_{as} deviates from the mean of the normal range of the physiological parameter associated with S_{as} (ie, $\mu_{as} \pm n_{as}.\sigma_{as}$), as shown in steps 2 to 4 in the pseudo-code below. The algorithm performs the same steps for all sensors correlated to S_{as}, as shown in steps 6 to 8. Next, the algorithm checks for deviations between sensor S_{as} and sensor S_i, as shown in step 9. If the condition in step 9 is satisfied, then the counter Pos is incremented. In step 14, the counter Pos is compared to a threshold TH. The threshold TH can be set based on experimental results (eg, average total number of correlated sensors for S_{as}). If the condition is satisfied, then the sensor S_{as} is classified as normal (true medical condition). Otherwise, the sensor S_{as} is classified as faulty.

A good anomaly detection system requires an optimal threshold to maximize the true-positive rate (TPR) and minimize the false-positive rate (FPR). There should be a tradeoff between TPR and FPR for fixed value of TH. If TH increases, then FPR and TPR decrease. In contrast, if TH decreases, then FPR and TPR increase. This relationship can be mathematically expressed as

$$\text{TH} \propto \frac{1}{\text{FPR}}.$$

When TH is very low, an anomaly detection system may misclassify normal instances, which will result in unnecessary interventions from healthcare professionals. In contrast, if TH is very high, then a system may misclassify anomalous instances, thereby reducing the detection rate (ie, real medical conditions may go undetected). In a healthcare scenario, detecting real medical conditions is extremely important because a patient's life may depend on detection.

TABLE 2 Confusion matrix for the proposed method for Dataset 1

		Actual condition	
		Anomaly	**Benign**
Predicted condition	Anomaly	1571 (True positive)	348 (False positive)
	Benign	0 (False negative)	10 652 (True negative)

3 | EXPERIMENTAL RESULTS AND DISCUSSION

The hardware specifications used for our experiments are a Core i7-2600 3.40 GHz CPU, Windows 7 Professional (64 bit) operating system, and 8 GB of RAM. To validate the proposed method, we considered two datasets: MIMIC DB dataset 221 (Dataset 1) and MIMIC DB dataset 276 (Dataset 2) from the MIMIC database. Dataset 1 covers the physiological parameters of respiration, pulse, arterial blood pressure, heart rate, and oxygen saturation. Dataset 2 covers the physiological parameters of respiration, pulse, arterial blood pressure, heart rate, oxygen saturation, central venous pressure, pulmonary artery wedge pressure, C.O, Tblood, ST1, ST2, ST3, and pulmonary arterial pressure. To evaluate the efficacy of the proposed algorithm, we added anomalous instances to these datasets. For our experimentation, we considered a sliding window size of 50. We considered two parameters to be highly correlated when $0.75 \leq \text{Corr}(P_i, P_j) \leq 1$.

The proposed algorithm was used to determine whether a sensor was. For Dataset 1, we considered a total of 12 571 windows. Among these 12 571 windows, we added anomalous instances to 1571 windows. We injected these anomalies

TABLE 3 Confusion matrix for the proposed method for Dataset 2

		Actual condition	
		Anomaly	**Benign**
Predicted condition	Anomaly	3456 (True positive)	1384 (False positive)
	Benign	27 (False negative)	18 616 (True negative)

in a random fashion. We had 11 000 benign windows of which 10 652 windows were correctly classified and all 1571 anomalous windows were correctly classified. For Dataset 2, we considered a total of 23 483 windows. Among these 23 483 windows, we added anomalous instances to 3483 windows. We injected these anomalies in a random fashion. We had 20 000 benign windows of which 18 616 windows were classified correctly and 3483 anomalous windows of which 3456 windows were classified correctly.

The performance of the proposed SFD algorithm was evaluated using the following statistical measures. TPR, detection rate, and recall refer to the ratio of the number of anomalous windows correctly categorized as anomalous over the total number of anomalous windows. FPR is the ratio between the number of benign windows incorrectly categorized as anomalous and the total number of benign windows.

The true-negative rate (TNR) is the ratio between the number of benign windows correctly categorized as benign and the total number of benign windows. The false-negative rate (FNR) is the ratio between the number of anomalous windows incorrectly categorized as benign and the total number of anomalous windows. Accuracy is the ratio between the number of benign and anomalous windows correctly categorized and the total number of windows (ie, (TP + TN + FP + FN)). Precision is the ratio between the number of anomalous windows correctly categorized and the predicted condition anomaly (ie, (TP + FP)). F_1 score is a statistical measure based on precision and recall.

$$\text{True positive rate (TPR)/Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}},$$

$$\text{False positive rate (FPR)} = \frac{\text{FP}}{\text{TN}+\text{FP}},$$

TABLE 4 Performance analysis for Dataset 1

	Metrics						
Method	**TPR**	**FPR**	**TNR**	**FNR**	**Accuracy**	**Precision**	**F_1 score**
[11]	67	36	64	33	64.37	21.00	31.98
[12]	100	20	80	0	82.50	41.66	58.82
[13]	100	5.08	94.92	0	95.55	73.76	84.90
[14]	100	4.28	95.72	0	96.25	76.94	86.97
[15]	100	4.39	95.61	0	96.16	76.49	86.68
[16]	100	4.67	95.33	0	95.91	75.36	85.95
[17]	100	4.5	95.5	0	96.06	76.04	86.39
[18]	99	5.83	94.17	1	94.77	70.80	82.56
Proposed method	100	3.16	96.84	0	**97.23**	81.87	90.03

TABLE 5 Performance analysis for Dataset 2

	Metrics						
Method	**TPR**	**FPR**	**TNR**	**FNR**	**Accuracy**	**Precision**	**F_1 score**
[11]	64	49	51	36	52.93	18.53	28.74
[12]	97	23	77	3	79.97	42.35	58.95
[13]	97	9.13	90.87	3	91.78	64.92	77.78
[14]	99	8.37	91.63	1	92.72	67.32	80.14
[15]	98.19	8.52	91.48	1.81	92.48	66.74	79.47
[16]	98.03	8.76	91.24	1.97	92.25	66.09	78.95
[17]	99	8.67	91.33	1	92.47	66.54	79.59
[18]	95.8	10.17	89.83	4.2	90.72	62.13	75.37
Proposed method	99.22	6.92	93.08	0.78	**93.99**	71.40	83.05

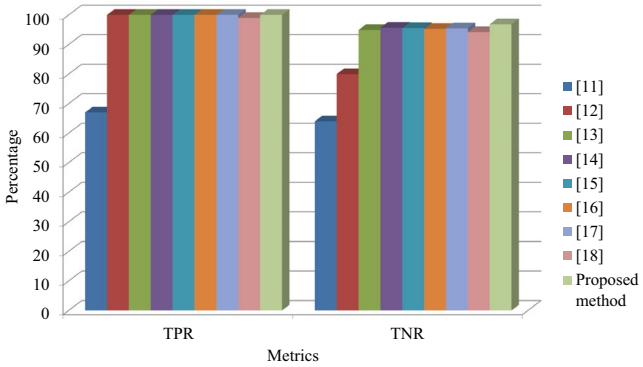

FIGURE 2 Performance analysis of various schemes based on TPR and TNR for Dataset 1

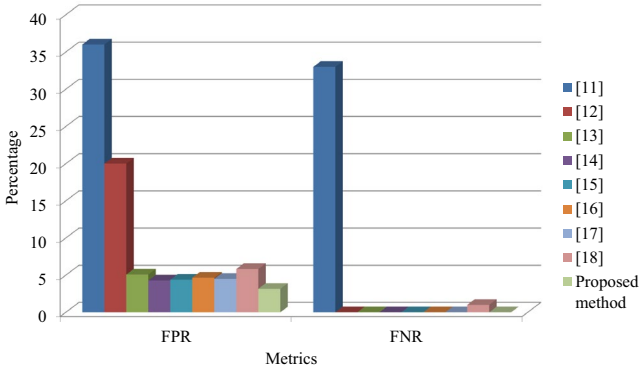

FIGURE 3 Performance analysis of various schemes based on FPR and FNR for Dataset 1

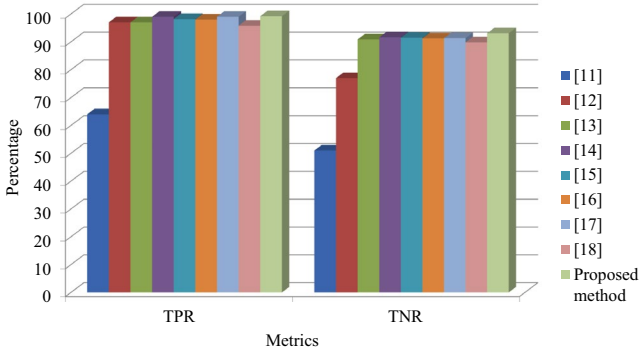

FIGURE 4 Performance analysis of various schemes based on TPR and TNR for Dataset 2

$$\text{True negative rate (TNR)} = \frac{\text{TN}}{\text{TN}+\text{FP}},$$

$$\text{False negative rate (FNR)} = \frac{\text{FN}}{\text{TP}+\text{FN}},$$

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}},$$

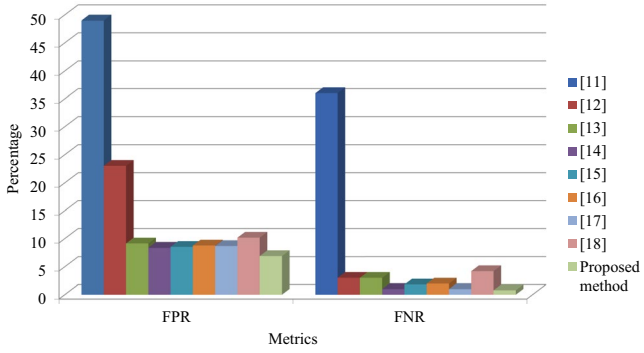

FIGURE 5 Performance analysis of various schemes based on FPR and FNR for Dataset 2

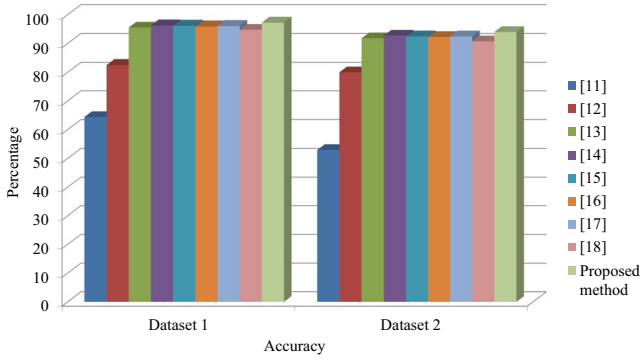

FIGURE 6 Accuracy results for various schemes

$$\text{F1 score} = 2\times\left(\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\right).$$

Tables 2 and 3 present the confusion matrices for the proposed scheme for Dataset 1 and Dataset 2, respectively. Tables 4 and 5 present complete evaluations of the proposed scheme based on the statistical metrics discussed above for Dataset 1 and Dataset 2, respectively. Figure 2 presents the performance of the SFD algorithm for Dataset 1 in terms of TPR and TNR. One can clearly see that the anomaly detection rate of the proposed scheme is 100%, which matches the results of existing schemes [12–17]. Figure 3 presents the performance of the SFD algorithm for Dataset 1 in terms of FPR and FNR. The reduction in FPR can be calculated as

$$\text{FPR}_{\text{Reduction}} = 1-\left(\frac{\text{FPR}_{\text{ProposedScheme}}}{\text{FPR}_{\text{ExistingScheme}}}\right),$$

where $\text{FPR}_{\text{ProposedScheme}}$ and $\text{FPR}_{\text{ExistingScheme}}$ are the FPR of the proposed scheme and an existing scheme, respectively. The proposed SFD algorithm significantly reduces FPR by 91.22%, 84.20%, 37.80%, 26.17%, 28.02%, 32.33%, 29.78%, and 45.80% compared to the existing schemes from Ref. [11–18], respectively. The proposed scheme for Dataset 1 achieves high accuracy, precision, and F_1 score based on its high detection

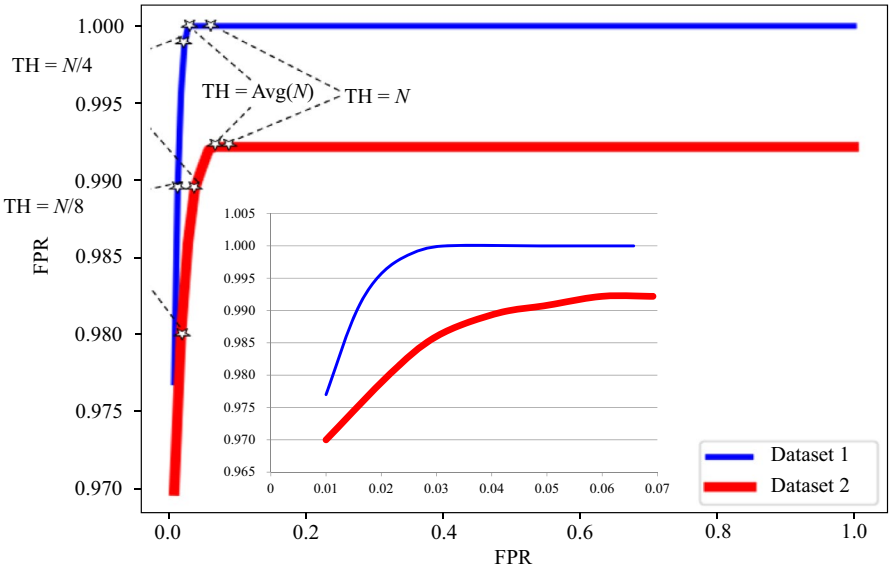

FIGURE 7 Performance analysis of the proposed method based on different thresholds for Dataset 1 and Dataset 2

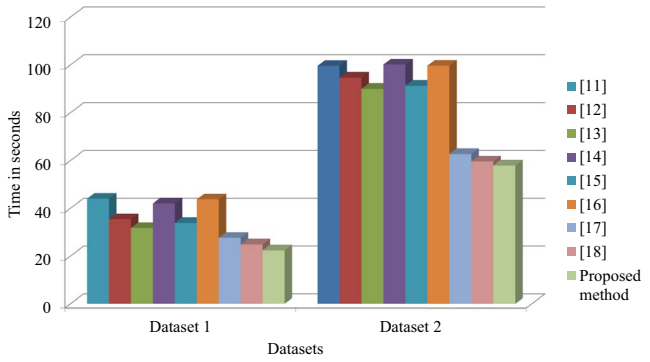

FIGURE 8 Run times of various schemes for Dataset 1 and Dataset 2

rate and low FPR. In summary, it can be concluded that the proposed scheme outperforms existing schemes on Dataset 1.

Figure 4 presents the performance of the SFD algorithm for Dataset 2 in terms of TPR and TNR. One can clearly see that the anomaly detection rate of the proposed algorithm is 99.22%. Figure 5 presents the performance of the SFD algorithm for Dataset 2 in terms of FPR and FNR. The anomaly detection rates of the existing schemes from Ref. [11–18] are 64%, 97%, 97%, 99%, 98.19%, 98.03%, 99%, and 95.8%, respectively. The proposed SFD algorithm significantly reduces the FPR by 85.88%, 69.91%, 24.21%, 17.32%, 18.78%, 21%, 20.18%, and 31.96%, respectively, when compared to these schemes. The proposed scheme for Dataset 2 achieves high accuracy, precision, and F_1 score based on a high detection rate and low FPR. Therefore, it can be concluded that the proposed scheme exhibits better performance compared to the existing schemes for Dataset 2. In Figure 6, one can clearly see that the proposed scheme achieves higher accuracy than existing schemes.

It should also be noted that the TH value plays a crucial role in experiments. It has been observed that when TH is equal to Avg(N), TPR reaches its highest value. However, when TH is less than Avg(N), TPR remains almost constant, but FPR increases. In contrast, when TH is greater than Avg(N), the detection rate decreases. Therefore, based on experimental results, we set TH equal to the average of the total number of correlated sensors because this TH value yields optimal results (low FPR and high TPR). The performance of the proposed algorithm with different thresholds is presented in Figure 7. In Table 6, one can see that when TH = Avg(N), the TPR of the proposed method reaches 100% with an FPR of 3.16% for Dataset 1. Similarly, for Dataset 2, TPR reaches 99.22% with an FPR of 6.92%.

The hardware specifications for computing the run times of various schemes are the same as those described above. We utilized the Python programming environment and

TABLE 6 TPR vs FPR on varying threshold for proposed method

		Threshold (TH)				
Datasets	**Metrics**	***N*/8**	***N*/4**	**Avg (*N*)**	**3*N*/4**	***N***
Dataset 1	TPR	98.95	99.89	100.00	100.00	100.00
	FPR	1.53	2.57	3.16	4.95	6.57
Dataset 2	TPR	98.02	98.95	99.22	99.22	99.22
	FPR	2.16	4.07	6.92	7.18	8.21

TABLE 7 Time complexities of various schemes

Method	Time complexity
[11]	$O(m^3)$
[12]	$O(m+n^3)$
[13]	$O(m+n^3)$
[14]	$O(Nn^3)$
[15]	$O(m+n^3)$
[16]	$O(m+n^5)$
[17]	$O(nm\log n)$
[18]	$O(m)$
Proposed method	$O(N)$

Note: m: number of physiological parameters; N: number of correlated physiological parameters; n: number of records in the training phase.

datasets 1 and 2 with numbers of windows of 12 571 and 23 483, respectively. The sliding window size was set to 50, and the threshold TH was set to Avg(N). Figure 8 presents the run times of various schemes for Dataset 1 and Dataset 2. One can see that the proposed method outperforms existing methods on both datasets 1 and 2 when TH is set to Avg(N).

3.1 | Theoretical analysis

This section presents theoretical analysis of the proposed SFD algorithm. In general, the time complexity of any given algorithm primarily depends on the time required to execute each instruction and the number of times each instruction is executed. In the proposed algorithm, suppose the ith instruction consumes C_i time per execution and executes T_i times. Then, the total cost of the ith instruction is C_iT_i, where C_i is a constant. One can clearly see that steps 2 to 4 and 14 to 17 are executed only once. Therefore, the total cost for steps 2 to 4 and 14 to 17 is

$$C_2+C_3+C_4+C_{14}+C_{15}+C_{16}+C_{17}.$$

The total cost for instructions in the loop can be computed as follows:

Step 5 will be executed $N+1$ times, meaning the total cost for step 5 is $C_5(N+1)$. The inner loop in steps 6 to 9 will be executed N times. Therefore, the total cost for steps 6 to 9 is

$$\left(C_6+C_7+C_8+C_9\right)N.$$

Calculating the cost of step 10 is somewhat more difficult. For step 10, T_{10} varies between 0 and N. In general, the total cost for step 10 is $C_{10}T_{10}$. Therefore, we consider the following three cases:

Case 1: When $T_{10}=0$

Total cost of step 10 is $C_{10}\times 0=0$.

Case 2: When $0<T_{10}\le N/2$

If T_{10} is at the higher bound (ie, $T_{10}=N/2$), then the total cost of step 10 is $C_{10}\times N/2$.

Case 3: When $N/2<T_{10}\le N$

If T_{10} is at the higher bound (ie, $T_{10}=N$), then the total cost of step 10 is $C_{10}\times N$.

Therefore, the total cost of the SFD algorithm is

$$T(N)=C_2+C_3+C_4+C_{14}+C_{15}+C_{16}+C_{17}+C_5(N+1)+\left(C_6+C_7+C_8+C_9\right)N+C_{10}T_{10}. \quad (5)$$

Equation (5) can be rewritten as

$$T(N)=\left(C_2+C_3+C_4+C_5+C_{14}+C_{15}+C_{16}+C_{17}\right)+\left(C_5+C_6+C_7+C_8+C_9\right)N+C_{10}T_{10}. \quad (6)$$

For **Case 1**, (6) will become

$$T(N)=\left(C_2+C_3+C_4+C_5+C_{14}+C_{15}+C_{16}+C_{17}\right)+\left(C_5+C_6+C_7+C_8+C_9\right)N. \quad (7)$$

For **Case 2**, (6) will become

$$T(N)=\left(C_2+C_3+C_4+C_5+C_{14}+C_{15}+C_{16}+C_{17}\right)+\left(C_5+C_6+C_7+C_8+C_9+\frac{C_{10}}{2}\right)N. \quad (8)$$

For **Case 3**, (1) will become

$$T(N)=\left(C_2+C_3+C_4+C_5+C_{14}+C_{15}+C_{16}+C_{17}\right)+\left(C_5+C_6+C_7+C_8+C_9+C_{10}\right)N. \quad (9)$$

Based on (7), (8), and (9), one can clearly see that the growth of the function is linear (ie, $a+b$), where a and b are constants. Therefore, based on this analysis, it can be concluded that for all cases, the computational complexity of the SFD algorithm is $O(N)$. Table 7 presents comparisons between the SFD and existing schemes based on computational complexity.

4 | CONCLUSION

This paper proposed an SFD algorithm for identifying faulty sensors precisely in healthcare applications. The proposed system makes use of Pearson correlation coefficients to identify strongly correlated sensors and applies simple statistical techniques to classify a sensor as faulty or normal. Our system was validated using two datasets from the MIMIC database. Experimental results revealed high detection rates (TPRs) of 100% and 99.22% for Dataset 1 and Dataset 2,

respectively. Our method also reduced the FPR by 26.17% and 17.32% compared to the method in [14] for Dataset 1 and Dataset 2, respectively. The proposed method achieved accuracy rates of 97.23% and 93.99% for Dataset 1 and Dataset 2, respectively. Another advantage is that our method does not make use of any prediction techniques, which reduces the complexity of the proposed system. Experimental results clearly highlighted the improved efficiency and accuracy of the proposed method. Additionally, the time complexity of the SFD algorithm is $O(N)$.

ORCID

Smrithy Girijakumari Sreekantan Nair https://orcid.org/0000-0002-9456-9817

REFERENCES

1. B. Chandrasekaran, R. Balakrishnan, and Y. Nogami, *Secure data communication using file hierarchy attribute based encryption in wireless body area networks*, J. Commun. Softw. Syst. **14** (2018), no. 1, 75–81.
2. SHIMMER, available at http://www.shimmer-research.com/tag/sensor
3. Lotus, available at https://www.memsic.com/userfiles/files/Datasheets/WSN/6020-0705-01_A_LOTUS.pdf
4. Q. Sun, F. Hu, and Q. Hao, *Mobile target scenario recognition via low-cost pyroelectric sensing system: Toward a context-enhanced accurate identification*, IEEE Trans. Syst., Man. Cybern. Syst. **44** (2014), no. 3, 375–384.
5. MICAZ/ZigBee Series (MPR2400), available at http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf
6. H. Dubois et al., *TinyNode: A comprehensive platform for wireless sensor network applications*, in Proc. Inf. Process. Sensor Netw. (Nashville, TN, USA), Apr. 19–21, 2006, pp. 358–365.
7. Sun SPOT, available at http://www.sunspotworld.com/
8. Cricket Mote, available at http://www.willow.co.uk/html/cricket_mote_platform.php
9. Telosb, available at http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf
10. D. J. Hill and B. S. Minsker, *Anomaly detection in streaming environmental sensor data: A data-driven modeling approach*, Environ. Model. Softw. **25** (2010), no. 9, 1014–1022.
11. F. Liu, X. Cheng, and D. Chen, *Insider attacker detection in wireless sensor networks*, in Proc. IEEE Int. Conf. Comput. Commun. (Barcelona, Spain), May 2007, pp. 1937–1945.
12. O. Salem et al., *Anomaly detection in medical wireless sensor networks using SVM and linear regression models*, Int. J. E-Health Med. Commun. **5** (2014), no. 1, 20–45.
13. S. A. Haque, M. Rahman, and S. M. Aziz, *Sensor anomaly detection in wireless sensor networks for healthcare*, Sensors **15** (2015), no. 4, 8764–8786.
14. B. Saneja and R. Rani, *An efficient approach for outlier detection in big sensor data of health care*, Int. J. Commun. Syst. **30** (2017), no. 17, article no. e3352.
15. M. U. H. Al Rasyid, *Anomalous data detection in WBAN measurements*, in Proc. Int. Electron. Symp. Knowl. Creation Intell. Comput. (Bali, Indonesia), Oct. 2018, pp. 303–309.
16. S. K. Nagdeo and J. Mahapatro, *Wireless body area network sensor faults and anomalous data detection and classification using machine learning*, in Proc. IEEE Bombay Section Signature Conf. (Mumbai, India), July 2019, pp. 1–6.
17. M. M. Nezhad and M. Eshghi, *Sensor single and multiple anomaly detection in wireless sensor networks for healthcare*, in Proc. Iranian Conf. Electr. Eng. (Yazd, Iran), May 2019, pp. 1751–1755.
18. N. Boudargham et al., *Toward fast and accurate emergency cases detection in BSNs*, IET Wireless Sensor Syst. **10** (2020), no. 1, 47–60.
19. A. L. Goldberger et al., *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals*, Circulation **101** (2000), no. 23, e215–e220.

AUTHOR BIOGRAPHIES

Smrithy Girijakumari Sreekantan Nair received her B. Tech degree in Information Technology from the University of Kerala, India, in 2011 and her M. Tech degree in Engineering Statistics from the Cochin University of Science and Technology, India, in 2014. Currently, she is a PhD student at the Department of Computer Applications of the National Institute of Technology, Tiruchirappalli, India. Her research interests include statistical techniques, anomaly detection, cloud computing, and the internet of things.

Ramadoss Balakrishnan received his M. Tech degree in Computer Science and Engineering from the Indian Institute of Technology, Delhi, India, in 1995 and his PhD in Applied Mathematics from the Indian Institute of Technology, Bombay, India, in 1983. Currently, he is working as a Professor of Computer Applications at the National Institute of Technology, Tiruchirappalli, Tamil Nadu, India. His main research interests are software testing methodologies, security and privacy in big data and the cloud, software metrics, data warehousing, data mining, WBL, and XML.